HEURÍSTICAS BASEADAS EM BUSCA LOCAL PARA A PROGRAMAÇÃO E ROTEIRIZAÇÃO DE EMBARCAÇÕES DE APOIO À EXPLORAÇÃO DE PETRÓLEO *OFF-SHORE*, ANALISANDO MÚLTIPLAS ESTRUTURAS DE VIZINHANÇA

Dalessandro Soares Vianna[1,3]
Edwin Benito Mitacc Meza[1,4]
Fernando Correa Hentzy[2,5]
Carlos Bazilio Martins[1,6]
Adriana Pereira de Medeiros[1,7]

[1]Instituto de Ciência e Tecnologia, Universidade Federal Fluminense.
Rua Recife, s/n – Jardim Bela Vista – CEP: 28890-000 – Rio das Ostras, RJ – Brasil.

[2]PETROBRAS – Petróleo Brasileiro S.A.
Rua Governador Roberto Silveira, 148, Centro, Macaé, RJ, 27910-000, Brasil.

[3]dalessandrosoares@yahoo.com.br, [4]emitacc@gmail.com, [5]fernando_hentzy@yahoo.com.br,
[6]bazilio@ic.uff.br, [7]adrianapm26@gmail.com

RESUMO

O crescente aumento de investimentos nas atividades *offshore* de petróleo tem promovido uma verdadeira corrida contra o tempo para se desenvolver toda a cadeia produtiva do setor. Neste cenário, as atividades de apoio logístico têm desafiado aos gestores a buscarem alternativas que possam reduzir os custos operacionais e ao mesmo tempo, atender a expectativa do cliente em relação ao nível de serviço ofertado. Dentre estas atividades, a programação e construção de roteiros de uma frota de embarcações, para o transporte de suprimentos para as unidades marítimas tem sido uma das mais importantes e ao mesmo tempo complexa devido ao aumento da frota e da demanda. Neste artigo são propostas três heurísticas baseadas nas metaheurísticas GRASP, ILS e VNS para o problema de programar e elaborar os roteiros de cada embarcação da frota. Os resultados computacionais apresentados demonstram a adequação das heurísticas propostas ao problema abordado, tendo-se destacado a heurística ILS em relação às outras.

PALAVRAS CHAVE. Programação e Roteirização, Otimização, Metaheurísticas.

ABSTRACT

The increasing investment in offshore oil activities has promoted a race against time to develop the entire production chain. In this scenario, the logistics activities has challenged managers to find alternatives that can reduce operating costs without meeting customer expectation regarding the level of service offered. Among these activities, the planning and construction of routes for a fleet of ships to transport supplies to marine units have been one of the most important and yet complex due to huge fleet's size and demand. This paper proposes three heuristics based on the metaheuristics GRASP, VNS and ILS for the problem of creating and scheduling routes of each vessel in the fleet. The computational results demonstrate that the proposed heuristics suit the problem addressed, having stood out the ILS heuristic among others.

KEYWORDS. Scheduling and Routing, Optimization, Metaheuristics.

1. Introdução

Diante das recentes descobertas de acumulações gigantescas de petróleo e gás natural na extensa camada Pré-Sal presente em boa parte da plataforma continental brasileira, a Petrobras, após mais de meio século de pesquisas e experiências acumuladas, novamente depara-se com desafios tecnológicos, gerenciais e logísticos, proporcionais à grandeza dos reservatórios localizados nessa nova fronteira tecnológica. Nesse contexto, as atividades *off-shore* têm sido fortemente intensificadas, recebendo um percentual significativo dos investimentos para que os objetivos estratégicos da companhia e do país possam ser alavancados.

Dentre estas atividades, a logística tem mostrado uma especial importância, dado o impacto financeiro e estratégico que exerce sobre as outras atividades. Do ponto de vista estratégico, a gestão eficiente dos processos logísticos pode se apresentar como vantagem competitiva, sendo a fonte desta vantagem a diferenciação da organização aos olhos dos clientes e a redução dos custos de operação.

Com a busca constante por melhorias no atendimento e aumento da eficiência (incluindo reduções de custos e de tempo de entrega), as organizações investem cada vez mais em sistemas informatizados. Nos últimos anos, os operadores logísticos atuantes no Brasil procuraram ampliar a utilização de sistemas ERP (*Enterprise Resource Planning* ou Sistemas Integrados de Gestão Empresarial, no Brasil) e roteirizadores (SILVA, 2010; SIKILERO, 2009; SLACK et al., 2009)

Dentro da Petrobras, vários suprimentos são transportados até as Unidades Marítimas (UM's) por meio de embarcações para que os projetos de exploração e produção de petróleo possam ser desenvolvidos. Assim, uma importante atividade é a determinação da roteirização e da programação de atendimento destas embarcações, de forma a atender todas as solicitações de suprimentos nos prazos pré-determinados para que as atividades a bordo das UM's possam ser realizadas de forma contínua, sem que haja interrupções que gerem atrasos nos projetos e a consequente perda econômica para a empresa.

A partir deste cenário bastante complexo no qual são consideradas inúmeras variáveis de decisão na elaboração das roteirizações e das programações das embarcações, é natural o surgimento de questionamentos sobre a eficácia do atual *modus operandi* aplicado à programação da frota existente. Principalmente, se for considerado as recentes descobertas de reservas de petróleo que apontam para um forte crescimento da demanda, o que leva a pensar em uma solução computacional para resolver o problema de programação e roteirização das embarcações.

Cabe ressaltar, que devido à existência de várias combinações possíveis dos parâmetros em um problema de programação e roteirização, nem sempre é possível encontrar na literatura um problema que se encaixe exatamente nas condições reais estudadas. Diante deste cenário, em (HENTZY et al., 2012) é proposto um modelo matemático que permite auxiliar ao operador na determinação das rotas e da programação de atendimento de cada uma das embarcações da frota, as quais podem estar localizadas nos portos ou cumprindo escala de atendimentos, de forma a minimizar as distâncias percorridas, otimizando a utilização da frota e reduzindo os riscos de paradas operacionais nas unidades marítimas por falta de suprimentos. Segundo Hentzy et al. (2012), o problema abordado é um problema de otimização combinatória de alta complexidade computacional, sendo do tipo NP-Hard. A medida que o número de UM's e embarcações aumentam, o tempo computacional requerido pelos métodos exatos se eleva de forma exponencial.

O objetivo deste trabalho é desenvolver heurísticas para a solução do modelo matemático descrito por Hentzy et al. (2012), referente à roteirização e programação de embarcações. Estas heurísticas são baseadas nas metaheurísticas GRASP, ILS, VNS e no método VND.

2. Descrição e Modelagem Matemática do Problema

Atualmente, na exploração e produção de petróleo em campos marítimos, são desenvolvidas atividades que necessitam de equipamentos de última geração e de uma série de suprimentos, tais como: água, diesel, alimentos, tubos de perfuração e produção, cimento, granéis

sólidos (baritina, bentonita e calcário) e granéis líquidos (fluidos de perfuração e completação), indispensáveis à execução dos projetos de poços marítimos (HENTZY et al., 2012). Estes insumos são transportados até as Unidades Marítimas (UM's) através de embarcações de transporte de suprimentos com características operacionais diversas (tipo, tamanho, velocidade e capacidade de carga).

A etapa de roteirização e programação das embarcações consiste em definir para cada embarcação o seu roteiro (escala de atendimento) com as tarefas de entrega (pedidos enviados pelas UM's) que cada uma deverá cumprir. Outras restrições devem ser consideradas no momento da programação, como a capacidade de transporte de cada embarcação devido à frota heterogênea.

Ao distribuir as programações para cada embarcação (roteirização e programação), o programador busca reduzir as distâncias percorridas por esta, utilizando para isto a ferramenta computacional GIS-SUB que é um software de localização geográfica que permite identificar a posição de cada UM e também calcular as distâncias entre elas.

Segundo Hentzy et al. (2012), a programação das demandas de uma UM deve, prioritariamente, utilizar o menor número possível de embarcações para que seja reduzida a quantidade de visitações a esta UM, otimizando a utilização da frota. Quando não há uma embarcação que possua estoque suficiente para atender à toda demanda da UM, a mesma poderá ser fracionada entre outras embarcações até que seja completamente atendida. Desta maneira, o objetivo principal da programação elaborada pelo programador é atender toda a demanda apresentada, empregando para isto o menor número possível de embarcações e minimizando as distâncias percorridas até o final do roteiro de cada uma delas.

Assim, para este problema, o modelo matemático proposto por Hentzy et al. (2012), faz uso das seguintes variáveis e notações.

- V, P: Conjunto de todos os vértices e de todos os produtos, respectivamente;
- N, C: Subconjuntos de V que representam as embarcações e os clientes, respectivamente;
- DEM_{ik}, CAP_{rk}: Representam a demanda do produto k pelo cliente i e a capacidade de fornecimento do produto k pela embarcação r;
- d_{ij}: distância entre os nós (cliente/embarcação);
- x_{rij}: Variável Binária que indica se a aresta (i, j) esta sendo utilizada pela embarcação r (x_{rij}=1) ou não (x_{rij}=0);
- u_{ri}: Variável Binária que indica se o cliente i esta sendo atendido pela embarcação r (u_{ri}=1) ou não (u_{ri} =0);
- y_{rij}: Fluxo em cada aresta (i, j) conduzido pela embarcação r;
- z_{rik}: Quantidade do produto k fornecido pela embarcação r ao cliente i.

Utilizando esta notação, o problema foi formulado como um modelo de programação inteira mista, onde o objetivo (1) é minimizar a distância percorrida pelas embarcações. A equação (2) garante que a rota é conexa, ou seja, a rota inicia na embarcação que irá atendê-la e finaliza no último cliente a ser atendido, após visitar todos os outros clientes associados à rota. A desigualdade (3) assegura que o fluxo na aresta (i,j) só poderá ocorrer se a aresta (i,j) esta sendo usada por alguma embarcação. A restrição (4) garante que, deve haver exatamente uma aresta chegando no nó j vindo da embarcação r, caso j seja atendida por r. Analogamente, a desigualdade (5) assegura que, deve haver no máximo uma aresta saindo no nó j vindo da embarcação r. A equação (6) garante que a quantidade atendida do produto k tem que ser igual à demanda do cliente pelo produto k. Já que mais de um produto pode ser transportado por uma embarcação, a restrição (7) é introduzida para que a quantidade fornecida do produto k, por uma determinada embarcação r, seja menor ou igual a sua respectiva capacidade do produto k. Devido à desigualdade (8), só poderá haver fornecimento se o cliente é atendido por uma determinada embarcação, onde M é um número muito grande. O domínio adequado das variáveis é determinado pelas restrições (9), (10), (11) e (12).

$$MIN \sum_{r\in N}\sum_{i\in V}\sum_{j\in C} d_{ij}x_{rij} \quad (1)$$

s.a.

$$\sum_{i\in V} y_{rij} - \sum_{i\in C} y_{rji} = u_{rj} \qquad \forall r \in N, \forall j \in C \quad (2)$$

$$y_{rij} \le M x_{rij} \qquad \forall r \in N, \forall j \in C \quad (3)$$

$$\sum_{i\in V} x_{rij} = u_{rj} \qquad \forall r \in N, \forall j \in C \quad (4)$$

$$\sum_{i\in C} x_{rji} \le u_{rj} \qquad \forall r \in N, \forall j \in V \quad (5)$$

$$\sum_{r\in N} z_{rik} = DEM_{ik} \qquad \forall i \in C, \forall k \in P \quad (6)$$

$$\sum_{i\in C} z_{rik} \le CAP_{rk} \qquad \forall r \in N, \forall k \in P \quad (7)$$

$$z_{rik} \le M u_{ri} \qquad \forall r \in N, \forall i \in V, \forall k \in P \quad (8)$$

$$x_{rij} \in [0,1] \qquad \forall r \in N, \forall i \in V, \forall j \in C \quad (9)$$

$$u_{ri} \in [0,1] \qquad \forall r \in N, \forall i \in V \quad (10)$$

$$y_{rij} \ge 0 \qquad \forall r \in N, \forall i \in V, \forall j \in C \quad (11)$$

$$z_{rik} \ge 0 \qquad \forall r \in N, \forall i \in V, \forall k \in P \quad (12)$$

Testes realizados em (HENTZY et al., 2012) mostraram que, para instâncias maiores, as soluções ótimas do modelo são praticamente impossíveis de serem obtidas em um tempo computacional hábil, se comparado com o tempo que tem o programador para construir a programação e roteirização das embarcações da frota. Para estes tipos de problemas com alta complexidade computacional, existem na literatura diversos trabalhos, que fazem uso de heurísticas em diversos problemas de roteirização (HAUGHTON, 2007; RODRIGUES, 2008; MAURI; LORENA, 2009; BELFIORE; YOSHIZAKI, 2009; PARK; KIM, 2010; BRANCHINI; ARMENTANO; LØKKETANGEN, 2010; ARANHA; MONTANÉ; VIANNA, 2012; MENDES, 2007), os quais incorporam diversas restrições operacionais como janelas de tempo, capacidade homogênea/heretogênea da frota, carga fracionada, dentre outras.

3. Modelagem Heurística

As heurísticas propostas neste trabalho são baseadas nas metaheurísticas GRASP (*Greedy Randomized Adaptive Search Procedure*) (FEO;RESENDE, 1995; RESENDE; RIBEIRO, 2003), ILS (*Iterated Local Search*) (LOURENÇO; MARTIN; STÜTZLE, 2002) e VNS (*Variable Neighborhood Search*). Dentre as heurísticas desenvolvidas, o método VND (*Variable Neighborhood Descent*) (MLADENOVIC; HANSEN, 1997) foi utilizado explorando três estruturas de vizinhança.

Segundo Feo e Resende (1995), o GRASP é uma estratégia composta por múltiplos inícios, usada primeiramente na otimização de funções contínuas e em otimização combinatória. Cada iteração do GRASP consiste em duas fases: construção de uma solução inicial e busca local. Na primeira fase, a construtiva, é criada uma lista restrita de candidatos (LRC) que é

formada pelos elementos cuja incorporação à solução parcial leva aos menores custos incrementais. Estes elementos podem ser, por exemplo, lotes de produção, lotes de estoque, ou seja, valores para as variáveis de decisão. A introdução de elementos finaliza quando a solução tornar-se factível, ou seja, quando a solução atende todas as restrições do problema. O GRASP é executado por um número máximo de iterações, o qual corresponde à quantidade de soluções factíveis encontradas pela fase construtiva em que foram aplicadas a busca local. Dentre estas soluções geradas é escolhida aquela de melhor valor.

O algoritmo ILS (LOURENÇO; MARTIN; STÜTZLE, 2002) envolve a aplicação repetida de um algoritmo de busca local aplicada às soluções encontradas pelos candidatos de um processo mais amplo de pesquisa que envolve uma caminhada aleatória tendenciosa através do espaço de busca. O algoritmo funciona através da construção de uma solução inicial, que é refinada usando uma estratégia de busca local. O laço de repetição do algoritmo envolve três etapas: uma perturbação da solução corrente; aplicação da busca local na solução perturbada; e um critério de aceitação que decide se a solução (ótimo local) obtida pelo processo de busca local deve ou não substituir a solução corrente.

A metaheurística de Busca em Vizinhança Variável, conhecida como VNS, foi proposta por Mladenovic e Hansen (1997). Esta metaheurística perturba sistematicamente a solução corrente utilizando para isso diferentes estruturas de vizinhança, o que pode resultar na degradação da solução. Este processo funciona como um mecanismo de diversificação da busca. Após o processo de perturbação da solução é executado um processo de busca local que é tipicamente um mecanismo de intensificação da busca. Sempre que a busca local não consegue melhorar a melhor solução altera-se o mecanismo de perturbação utilizando outra estrutura de vizinhança.

Já o VND, segundo Mladenovic e Hansen (1997), é um método que explora o espaço de soluções através do uso de várias estruturas de vizinhança, diferentemente do método de busca local tradicional que utiliza uma única estrutura de vizinhança. Estas vizinhanças estendidas procuram por soluções aprimorantes que estão "mais distantes" da solução atual, permitindo ao método escapar de ótimos locais com respeito a uma vizinhança menor.

4. Heurísticas Propostas

Foram propostas neste trabalho três heurísticas: a primeira baseada na metaheurística GRASP; a segunda na metaheurística ILS; e a última na metaheurística VNS. Todas as três heurísticas utilizam, em sua etapa de busca local, o método VND. Desta forma, as heurísticas desenvolvidas foram nomeadas de **GRASP+VND**, **ILS+VND** e **VNS+VND.**

As subseções a seguir descrevem cada etapa desenvolvida para as heurísticas propostas**.**

4.1. Representação da solução

A estrutura principal utilizada para codificar uma solução para o problema abordado foi um vetor *v* de *N* posições; em cada posição *r* deste vetor é armazenado outro vetor representando a rota percorrida pelo navio *r*. A Figura 1 apresenta um exemplo desta codificação para *N*=4 navios e *C*=6 clientes (unidades marítimas).

1	2	3	
2	1	6	5
3	5	3	4
4	2	1	

Figura 1. Exemplo de codificação.

Na Figura 1, por exemplo, o navio 2 visitará os clientes 1, 6 e 5, nesta ordem. É importante lembrar que, no problema abordado, um cliente pode ter a sua demanda atendida por mais de um navio.

Também é utilizada uma estrutura de dados auxiliar $Atendimento_{N\times C\times P,}$ a qual informa a quantidade do produto $k\in P$ demandado pelo cliente $i\in C$ que é fornecido pelo navio $r\in N$.

4.2. Função objetivo

Uma solução é avaliada pelo somatório das distâncias percorridas por cada navio, ou seja, pelo somatório das distâncias de todas as rotas.

4.3. Método construtivo guloso aleatorizado

O método construtivo insere gradativamente um cliente i na rota de um navio r. A escolha do próximo cliente i a ser atendido, qual o navio r que irá atendê-lo e a posição s da rota do navio r em que o cliente i será inserido utiliza o seguinte critério guloso: é escolhido a tupla (i, r, s) que acrescenta a menor distância ao somatório das rotas, ou seja, a que produz o menor incremento à função objetivo.

É montada então uma lista LC de tuplas candidatas (i, r, s), as quais são tuplas onde o cliente i possui demanda de algum produto k ainda não atendida e o navio r ainda possui oferta do mesmo produto k. De LC é montada uma lista restrita de candidatas, LRC, a qual possui os elementos de LC mais promissores. Esses elementos são aqueles que apresentam valores não superiores a $t_{mim} + \alpha \times (t_{max} - t_{min})$, onde t_{min} e t_{max} representam, respectivamente, o incremento da tupla mais promissora e da menos promissora de LC; e $\alpha \in [0,1]$ é um parâmetro de entrada do algoritmo, no qual α=0 representa um algoritmo puramente guloso e α=1 um método completamente aleatório. Este algoritmo construtivo é descrito na Figura 2.

Algoritmo Construtivo (α)
Início
  Enquanto existir um cliente ainda não atendido por completo **faça**
  Início
    Monte a lista de candidatos LC.
    De LC extraia a lista restrita de candidatos LRC.
    Escolha uma tupla $t(i,r,s)$ aleatoriamente de LRC.
    Insira o cliente i na posição s da rota do navio r.
    Atualize a estrutura auxiliar *Atendimento* com a quantidade máxima de cada produto $k \in P$ que o navio r pode atender da demanda do cliente i pelo produto k.
  Fim
Fim

Figura 2. Algoritmo construtivo guloso aleatorizado.

4.4. Estruturas de vizinhança

Foram utilizados quatro movimentos: TROCA, REALOCAÇÃO, INTERCÂMBIO e INTERCÂMBIO2. O primeiro é um movimento intra-rota e os outros inter-rota. Detalhes sobre as vizinhanças obtidas através destes dois movimentos são dados nas Subseções 4.4.1, 4.4.2, 4.4.3 e 4.4.4.

4.4.1. Vizinhança V_TROCA

A vizinhança V_TROCA é obtida através dos movimentos TROCA. Um movimento TROCA é caracterizado pela troca de posição de dois clientes que estão na mesma rota. Ela é caracterizada pela tupla (r, i_1, s_1, i_2, s_2), onde, na rota do navio r, o cliente i_1 que se encontra na posição s_1 passará a ocupar a posição s_2; e o cliente i_2 que se encontra na posição s_2 passará a ocupar a posição s_1. A Figura 3 apresenta um exemplo de movimento TROCA, onde os clientes 1 e 5 são trocados na rota do navio 2.

1	2	3	
2	1	6	5
3	5	3	4
4	2	1	

→

1	2	3	
2	5	6	1
3	5	3	4
4	2	1	

Figura 3. Exemplo de movimento TROCA.

4.4.2. Vizinhança V_REALOCAÇÃO

A vizinhança V_REALOCAÇÃO é obtida através dos movimentos REALOCAÇÃO. Um movimento REALOCAÇÃO é caracterizado pela migração de um cliente de uma rota para outra. Ela é caracterizada pela tupla (r_1, i, s_1, r_2, s_2), onde o cliente i que se encontra na posição s_1 da rota do navio r_1 passará a ocupar a posição s_2 da rota do navio r_2. É importante destacar que a tupla (r_1, i, s_1, r_2, s_2) só é possível se o navio r_2 puder atender a demanda de cada produto $k \in P$ do cliente i atendida pelo navio r_1. A Figura 4 apresenta um exemplo deste movimento, onde o cliente 1 da rota do navio 2 (posição 1) é migrado para a rota do navio 1 (posição 2).

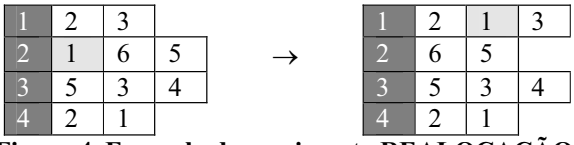

Figura 4. Exemplo de movimento REALOCAÇÃO.

Vale enfatizar uma peculiaridade deste movimento: se o cliente i já existir na rota r_2 então a rota do navio r_2 não é modificada, só alterando a estrutura *Atendimento* para refletir este incremento no atendimento do cliente i pelo navio r_2. A Figura 5 apresenta um exemplo deste caso, onde o cliente 1 é realocado da rota do navio 2 para a rota do navio 4.

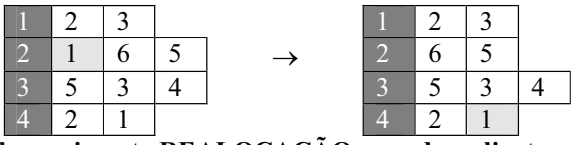

Figura 5. Exemplo de movimento REALOCAÇÃO quando o cliente existe na rota destino.

4.4.3. Vizinhança V_INTERCÂMBIO

A vizinhança V_INTERCÂMBIO é obtida através dos movimentos INTERCÂMBIO. Um movimento INTERCÂMBIO é caracterizado pela troca de posição entre clientes que estão localizados em rotas diferentes. Ela é caracterizada pela tupla (r_1, i_1, s_1, r_2, i_2, s_2), onde o cliente i_1 que se encontra na posição s_1 da rota do navio r_1 trocará de posição com o cliente i_2 que se encontra na posição s_2 da rota do navio r_2. É importante destacar que a tupla (r_1, i_1, s_1, r_2, i_2, s_2) só é possível se atender quatro condições: o navio r_2 pode atender a demanda de cada produto $k_1 \in P$ do cliente i_1 atendida pelo navio r_1; o navio r_1 pode atender a demanda de cada produto $k_2 \in P$ do cliente i_2 atendida pelo navio r_2; o cliente i_1 não existe na rota do navio r_2; e o cliente i_2 não existe na rota do navio r_1. A Figura 6 apresenta um exemplo de movimento INTERCÂMBIO, onde o cliente 1 da rota do navio 2 (posição 1) será intercambiado com o cliente 3 do navio 1 (posição 2).

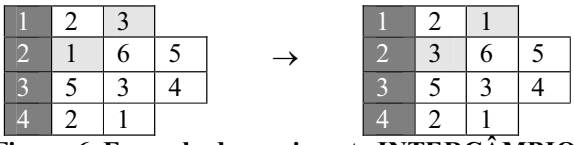

Figura 6. Exemplo de movimento INTERCÂMBIO.

4.4.4. Vizinhança V_INTERCÂMBIO2

A vizinhança V_INTERCÂMBIO2 é obtida através dos movimentos INTERCÂMBIO2. Este movimento é uma extensão do movimento INTERCÂMBIO, onde dois movimentos INTERCÂMBIO são realizados consecutivamente.

4.5. VND desenvolvido

Como já mencionado anteriormente, as três heurísticas propostas neste trabalho utilizam o método VND em sua etapa de busca local. A Figura 7 apresenta o pseudocódigo do método VND utilizado neste trabalho, no qual são avaliadas as vizinhanças V_TROCA (N_1), V_REALOCAÇÃO (N_2) e V_INTERCÂMBIO (N_3).

```
Algoritmo VND (s)
Início
    r ← 3; // número de vizinhanças
    k ← 1; // tipo da vizinhança corrente
    Enquanto k ≤ r faça
    Início
        Encontre o melhor vizinho s' ∈ N_k (s);
        Se f(s') < f(s) então
            s ← s';
            k ← 1; // primeira vizinhança
        Senão
            k ← k + 1; // próxima vizinhança
    Fim
    Retorne s;
Fim
```

Figura 7. Método VND utilizado.

4.6. Heurística GRASP+VND proposta

A Figura 8 apresenta o pseudocódigo da heurística **GRASP+VND** desenvolvida, no qual uma solução inicial é criada e em seguida refinada. Este processo é repetido até que o tempo limite (*tempo_limite*) de execução seja alcançado. Na etapa de construção foi usado o algoritmo **Construtivo** descrito na Subseção 4.3, com α=0.1. O método **VND** descrito na Subseção 4.5 é utilizado na etapa de refinamento. Por fim, a melhor solução encontrada é retornada.

```
Algoritmo GRASP+VND (α, tempo_limite)
Início
    s* ← ∅;
    Enquanto tempo_limite não atingido faça
    Início
        Construa uma solução s_0 com o algoritmo Construtivo;
        Aplique, em s_0, o método VND, gerando a solução s;
        Se f(s) < f(s*) então
            s* ← s;
    Fim
    Retorne s*;
Fim
```

Figura 8. Heurística GRASP+VND.

4.7. Heurística ILS+VND proposta

A Figura 9 apresenta o algoritmo da heurística **ILS+VND** desenvolvida. O algoritmo recebe como entrada o tempo de execução em segundos *tempo_limite*. Na etapa de construção foi usado o algoritmo **Construtivo** descrito na Subseção 4.3, com α=0 (puramente guloso). O método **VND** descrito na Subseção 4.5 é utilizado na etapa de refinamento. A vizinhança V_INTERCÂMBIO2 foi utilizada na etapa de perturbação. Por fim, a melhor solução encontrada é retornada.

```
Algoritmo ILS+VND (tempo_limite)
Início
    Construa uma solução s0 com o algoritmo Construtivo;
    Aplique, em s0, o método VND, gerando a solução s*;
    Enquanto tempo_limite não atingido faça
    Início
        Escolha um vizinho s de s* na vizinhança V_INTERCÂMBIO2;
        Aplique, em s, o método VND, gerando a solução s';
        Se f(s') < f(s*) então
            s* ← s';
    Fim
    Retorne s*;
Fim
```

Figura 9. Heurística ILS+VND.

4.8. Heurística VNS+VND proposta

A Figura 10 apresenta o algoritmo da heurística **VNS+VND** desenvolvida. O algoritmo recebe como entrada o tempo de execução em segundos *tempo_limite*. Na etapa de construção foi usado o algoritmo **Construtivo** descrito na Subseção 4.3, com α=0 (puramente guloso). O método **VND** descrito na Subseção 4.5 é utilizado na etapa de refinamento. As vizinhanças V_INTERCÂMBIO (N_1) e V_INTERCÂMBIO2 (N_2) foram utilizadas na etapa de perturbação. Por fim, a melhor solução encontrada é retornada.

```
Algoritmo VNS+VND (tempo_limite)
Início
    Construa uma solução s* com o algoritmo Construtivo;
    r ← 2;
    Enquanto tempo_limite não atingido faça
    Início
        k ← 1;  //primeira estrutura de vizinhança
        Enquanto k ≤ r faça
        Início
            Escolha um vizinho s de s* na vizinhança Nk;
            Aplique, em s, o método VND, gerando a solução s';
            Se f(s') < f(s*) então
                s* ← s';
                k ← 1;  //primeira estrutura de vizinhança
            Senão
                k ← k + 1;  //próxima estrutura de vizinhança
        Fim
    Fim
    Retorne s*;
Fim
```

Figura 10. Heurística VNS+VND

5. Testes Computacionais

5.1. Características dos cenários usados

Com o objetivo de testar a metodologia proposta foram construídos três grupos de cenários distintos para o problema programação e roteirização de embarcações. Estes grupos definem cenários de pequeno, médio e grande porte.

Para a construção dos dois primeiros grupos de cenários, pequeno e médio porte, foram utilizados dados obtidos de um banco de dados com informações reais de demanda, estoque, entre outras, permitindo assim uma melhor compreensão do problema abordado e suas restrições.

As localizações das UM's bem como das embarcações foram obtidas através do programa desenvolvido pela equipe de Geodésia da Petrobras chamado GIS-SUB, sendo utilizadas as coordenadas geográficas UTM (N, E).

Para a construção do grupo de cenários de grande porte, foi construído um gerador de instâncias/cenários. Esse gerador recebe como entrada o número de navios, de clientes e de produtos e cria cenários onde cada atributo é definido de maneira aleatória. Foram geradas 5 instâncias com um número maior de embarcações e de UM's.

A Tabela 1 descreve as características destes grupos de cenários que servirão de entrada para a realização dos testes em ambiente computacional.

Tabela 1 – Características dos cenários utilizados para testes.

Grupo	Nome do cenário	Número de Embarcações	Número de UM´s (Clientes)	Número de Produtos
Pequeno Porte	PP–1	4	11	5
	PP–2	4	12	5
	PP–3	5	13	5
Médio Porte	MP–1	7	23	5
	MP–2	10	28	5
Grande Porte	GP–1	10	50	5
	GP–2	15	60	5
	GP–3	15	70	8
	GP–4	15	85	8
	GP–5	15	100	10

5.2. Análise dos resultados

Os testes foram executados utilizando a seguinte configuração de hardware: processador Intel Core 2 Duo, memória RAM de 4.0 GB e sistema operacional Windows Vista.

Como critério de parada, para as três heurísticas propostas, foi utilizado tempo de execução: 60 segundos para cada cenário de cada grupo.

A Tabela 2 apresenta os resultados encontrados para cenários de pequeno porte, comparando os resultados obtidos com as soluções ótimas do modelo e as melhores soluções obtidas pelas heurísticas propostas após cinco execuções. Cabe ressaltar, que devido às características dos cenários serem de pequeno porte foi possível a obtenção do valor ótimo do problema.

Tabela 2 – Resultados dos cenários de pequeno porte.

Nome	Solução Ótima (km)	GRASP+VND		ILS+VND		VNS+VND	
		(km)	Gap(%)	(km)	Gap(%)	(km)	Gap(%)
PP–1	739	739	0,0	739	0,0	739	0,0
PP–2	719	719	0,0	719	0,0	719	0,0
PP–3	371	371	0,0	371	0,0	371	0,0

A Tabela 3 apresenta os resultados encontrados para cenários de médio porte, comparando os resultados obtidos com os roteiros construídos intuitivamente por decisores da empresa. No primeiro cenário, as três heurísticas atingiram uma boa melhora ($\cong 45\%$) em relação ao roteiro do decisor; no segundo cenário, a heurística **GRASP+VND** se sobressaiu em relação às outras, alcançando uma melhora de 43,0% contra uma melhora de 37,8% das heurísticas **ILS+VND** e **VNS+VND**.

Tabela 3 – Resultados dos cenários de médio porte.

Nome	Decisor (km)	GRASP+VND		ILS+VND		VNS+VND	
		(km)	Gap(%)	(km)	Gap(%)	(km)	Gap(%)
MP–1	3711	2016	-45,7	2028	-45,4	2028	-45,4
MP–2	4881	2784	-43,0	3035	-37,8	3035	-37,8

A Tabela 4 apresenta os resultados encontrados para cenários de grande porte. Estes cenários foram montados no intuito de comparar as três heurísticas propostas quando as dimensões do problema aumentam. A segunda coluna (Mínimo) representa o valor da melhor solução encontrada dentre as três heurísticas propostas; o objetivo é comparar as soluções encontradas por cada heurística com a melhor solução conhecida (coluna Mínimo). Neste teste pode-se perceber que a heurística **ILS+VND** começa a se destacar conforme as dimensões do problema aumentam, enquanto há uma tendência de perda de eficiência da heurística **VNS+VND**. Na média, a heurística **ILS+VND** se sobressaiu em relação às outras. A heurística **GRASP+VND** também apresentou bons resultados.

Tabela 4 – Resultados dos cenários de grande porte.

Nome	Mínimo (km)	GRASP+VND		ILS+VND		VNS+VND	
		(km)	Gap(%)	(km)	Gap(%)	(km)	Gap(%)
GP–1	284	289	1,8	287	1,1	284	0,0
GP–2	364	378	3,8	376	3,3	364	0,0
GP–3	533	533	0,0	554	3,9	562	5,4
GP–4	598	647	8,2	598	0,0	842	40,8
GP–5	664	692	4,2	664	0,0	973	46,5
Média:			3,6		1,7		18,6

6. Considerações Finais

O objetivo deste trabalho foi o de desenvolver heurísticas para auxiliar na programação e elaboração dos roteiros das embarcações de uma determinada frota no ambiente *off-shore*. Com esse intuito, três heurísticas foram elaboradas, **GRASP+VND**, **ILS+VND** e **VNS+VND**, as quais combinam as metaheurísticas GRASP, ILS e VNS e o método VND.

Para avaliação das heurísticas propostas, foram elaborados três grupos de testes, os quais somam oito diferentes cenários. Os dois primeiros grupos são baseados em dados reais da empresa e o último grupo foi gerado neste trabalho.

Para os cenários de pequeno porte, as heurísticas propostas alcançaram os ótimos globais, divulgados em (HENTZY et al., 2012).

Para os cenários de médio porte, as heurísticas foram comparadas com os roteiros construídos intuitivamente por decisores da empresa. As três heurísticas foram bem superiores aos resultados descritos pelo decisor, apresentando um desempenho semelhante.

Os cenários de grande porte foram construídos com o objetivo de avaliar as três heurísticas propostas quando as dimensões do problema aumentam. Nestes cenários, percebeu-se que a heurística **ILS+VND** começa a se destacar conforme as dimensões do problema aumentam, enquanto há uma tendência de perda de eficiência da heurística **VNS+VND**. Quando a média dos *gaps* foi avaliada, a heurística **ILS+VND** mostrou-se superior às outras. A heurística **GRASP+VND** também apresentou bons resultados.

Como trabalho futuro, pretende-se adicionar outras restrições ao modelo como, por exemplo, janela de tempo, coleta-entrega, limite de combustível, dentre outros.

7. Referências Bibliográficas

[1] ARANHA, M. T. S.; MONTANÉ, F. A. T. & VIANNA, D. S. Heurísticas para o problema de transporte terrestre de pessoas da petrobras na Bacia de Campos. Submetido ao Brazilian Journal of Operations & Production Management, 2012.

[2] BELFIORE, P. & YOSHIZAKI, H.T. Y. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. European Journal of Operational Research, Vol. 199, pp. 750-758, 2009.
[3] BRANCHINI, R. M.; ARMENTANO, V. A. & LØKKETANGEN, A. Adaptive granular local search heuristic for a dynamic vehicle routing problem. Computers & Operations Research, Vol. 36, pp. 2955-2968, 2009.
[4] CHRISTIANSEN, M.; FAGERHOLT, K & RONEN, D. Ship routing and scheduling: status and perpectives. Transportation Science, 38:1, p. 1-18, 2004.
[5] FEO, T. A. & RESENDE, M. G. C. Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109-133, 1995.
[6] HAUGHTON, M. A. Assigning delivery routes to drivers under variable customer demands. Transportation Research, Part E, Vol. 43, pp. 157-172, 2007.
[7] HENTZY, F. C.; MEZA, E. B. M.; VIANNA, D. S. & DIANIN, M. F. V. Um modelo matemático para a programação e roteirização de embarcações de apoio à exploração de petróleo offshore. Submetido ao Encontro Nacional de Engenharia de Produção, 2012.
[8] LOURENÇO, H. R.; MARTIN, O. C. & STÜTZLE, T. Iterated local search. In: Glover F, Kochenberger G (eds.). Handbook of Metaheuristics. Kluwer, 2002. p.321–353.
[9] MAURI, G. R. & LORENA, L. A. N. Uma nova abordagem para o problema dial-a-ride, Produção, Vol. 19, No. 1, pp. 041-054, 2009.
[10] MENDES, A.B. Programação de Frota de Apoio a Operações "Offshore" sujeita à requisição de Múltiplas Embarcações para uma mesma Tarefa. Tese de doutorado em Engenharia apresentada à Escola Politécnica da USP, 2007.
[11] MLADENOVIC, N. & HANSEN, P. Variable Neighborhood Search. Computer and Operations Research 24, 1097-1100, 1997.
[12] PARK, J. & KIM, B. The school bus routing problem: A review. European Journal of Operational Research, Vol. 202, pp. 311-319, 2010.
[13] RESENDE, M. G. C. & RIBEIRO, C. C. Greedy randomized adaptive search procedures. In F. Glover e G. Kochenberger (eds.), Handbook of metaheuristics. Kluwer, 219-249, 2003.
[14] RODRIGUES, S. B. Metaheurística Colônia de Formigas aplicada a um Problema de Roteamento de Veículos: caso da Itaipu Binacional. In: Simpósio Brasileiro de Pesquisa Operacional, 40, 2008.
[15] SIKILERO, C. B. Contribuição das características logísticas para as prioridades competitivas: um estudo de caso em uma empresa da indústria de refrigerantes. Dissertação de Mestrado. Universidade do Vale do Rio dos Sinos - Unisinos. 2009.
[16] SILVA, G. L. Uma nova abordagem para o problema de roteirização de veículos com restrições operacionais. Tese (Doutorado em Transportes)-Universidade de Brasília, Brasília, 2010.
[17] SLACK, N.; CHAMBERS, S; HARRISON, A. & JOHNSTON, R. Administração da Produção. São Paulo: Atlas, 2009.