LII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2020

Modelo Matemático e *Variable Neighborhood Search* aplicados ao Problema do Roteamento de Esteiras no Embarque de Minério de Ferro

Cleber Silva Ferreira
Instituto Federal do Espírito Santo - Serra
ES-010 Km-6,5 - Manguinhos, Serra - ES, 29173-087
cleber.silva.ferreir@gmail.com

Leandro Colombi Resendo
Instituto Federal do Espírito Santo - Serra
ES-010 Km-6,5 - Manguinhos, Serra - ES, 29173-087
leandro@ifes.edu.br

RESUMO

Neste artigo o Problema do Roteamento de Esteiras no Embarque de Minério de Ferro é modelado pelo Problema de Alocação de Demandas encontrado no planejamento de produção. Com o objetivo de minimizar o tempo de execução da lista demandas, foi proposto um modelo de programação linear inteira e uma heurística, baseada no método *Variable Neighborhood Search*. Para a resolução das instâncias geradas pelo modelo matemático foi usado o CPLEX. Um cenário de pequeno porte foi usado para avaliar o desempenho dos dois métodos e um cenário de um porto real foi usado para verificar a aplicabilidade da heurística.

PALAVRAS CHAVE. Programação Linear Inteira, *Variable Neighborhood Search*, Roteamento, Porto, Carregamento de Minério

Tópicos (Problema de alocação de demandas em portos de graneis sólidos, Roteamento de esteiras)

ABSTRACT

In this article, the iron ore shipping routing problem is modeled by Demand Allocation Problem found in production planning. In order to minimize the execution time of a demand list, an Integer Linear Programming Model and a heuristic model, based on the Variable Neighborhood Search method have been proposed. To solve the instances generated by mathematics model the CPLEX was used. A small scenario was used to assess the performance of the two methods and a real port scenario was used to verify the heuristic applicability.

KEYWORDS. Integer Linear Programming, Variable Neighborhood Search, Routing, Port, Ore Loading

Paper topics (Demand allocation problem in solid bulk ports, conveyor routing)

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

1. Introdução

O transporte marítimo é o instrumento logístico mais importantes no comércio mundial. Conforme consta no relatório da conferência das nações unidas sobre comercio e desenvolvimento (UNCTAD) de 2018, 80% do volume do comercio global é movimentado através de navios e a taxa de crescimento deste setor é de 4% ao ano. O produto mais transportado nos oceanos é o granel sólido que representa 42,3% de todo volume transportado pelo mar. O minério de ferro representou 46,07 % de toda a produção de granel sólido em 2018 [on Trade e Development, 2018].

O minério de ferro extraído nas minas é transportado em composições de vagões em uma estrada de ferro, é recebido pelo porto nos viradores de vagões e estocado em pátios pelas empilhadeiras. Posteriormente o minério é retomado pelas recuperadoras e embarcado em navios através dos carregadores de navios. Esses equipamentos são interligados pelas esteiras e juntos formam o caminho que define uma rota [Menezes et al., 2017]. Na literatura encontramos a utilização de Modelo de Programação Linear Inteira Mista (PLIM) provendo solução para o atendimento de demandas em portos por rotas de equipamentos como encontrado nos trabalhos [Menezes et al., 2017] e [Menezes et al., 2016]. No artigo [Unsal e Oguz, 2019] é abordado o agendamento de tarefas de um terminal integrado de graneis sólidos sem considerar a existência de rotas de equipamentos. No artigo [Nishi et al., 2020] um Problema de Alocação de Berços (PAB) foi resolvido com um modelo PLIM baseado em um Problema de Roteamento de Veículos com Vários Depósitos com Janelas Temporais (PRVVDJT). O artigo [Belassiria et al., 2019] mostra um porto que atende a uma usina elétrica movida a carvão, que é transportado por um sistema de esteiras, onde foi proposto um modelo PLIM e uma heurística baseada em busca tabu. O problema de recebimento de graneis em uma usina siderúrgica é abordado no artigo [Tang et al., 2016a] onde é proposto um PLIM resolvido por decomposição de Benders.

Este artigo propõe a investigação do processo de otimização do roteamento de demandas no parque de equipamentos, em um porto de minério localizado em Vitória-ES, Brasil. Para isso é proposto um modelo PLIM, baseado em alocação de demandas, capaz de sequenciar as demandas atendidas por rotas de equipamentos, visando minimizar seu tempo de atendimento. Devido a complexidade computacional do problema, ainda é proposto nesse trabalho um método heurístico baseado no *Variable Neighborhood Search* (VNS). O restante deste artigo está dividido da seguinte forma: a Seção 2 apresenta uma revisão bibliográfica. A Seção 3 apresenta a descrição do problema investigado e o conceito de Problema de Alocação de Demandas (PAD). A Seção 4 descreve o modelo matemático proposto. A Seção 5 apresenta a proposta da heurística. A Seção 6 apresenta os resultados numéricos dos experimentos realizados e finalmente, a Seção 7 apresenta as conclusões.

2. Revisão Bibliográfica

Na literatura, existem poucos trabalhos diretamente relacionados com rotas de equipamentos em portos de embarque de granéis sólidos. Na pesquisa [Menezes et al., 2017] foi observado que as rotas são previamente definidas, considerando o tipo de produto a ser embarcado no porto e os custos causados por alterações na qualidade do material. Foi proposto um modelo de PLIM, resolvido através de procedimento de geração de colunas e um algoritmo *branch and price*. No trabalho [Menezes et al., 2016] uma abordagem hierárquica foi apresentada para resolver os problemas de planejamento da lista de demandas e sequenciamento da operação dos atendimentos das demandas, além do modelo PLIM foi gerado uma heurística baseada em *Greedy Randomized Adaptive Search Procedure* (GRASP). No trabalho [Pratap et al., 2016] é mostrado um cenário de um porto de carvão que utiliza descarregadores de navios e estoque em pátios onde é proposto um PLIM e duas heurísticas, uma baseada em algoritmo genético associado a busca gulosa e a outra utilizando algoritmo evolutivo. Porém, em tal trabalho não são consideradas as rotas de equipamentos. No trabalho [Pratap et al., 2018] foi abordado os desafios do cenário do artigo [Pratap

et al., 2016] sendo mais abrangente ao considerar o final da cadeia produtiva. Para a otimização, um modelo PLIM foi proposto. A otimização de um porto de exportação de carvão é o tema abordado no artigo [de Paula et al., 2019], que mostrou o transporte do carvão através de esteiras até o embarque do produto nos navios. Além da integração com um PAD, tambem foi proposto uma heurística baseada em algoritmo genético. O artigo [Tang et al., 2016b] propôs um modelo polinomial não determinístico observando fatores como número de navios, número de berços, as restrições de atracação dos navios, entre outros fatores de portos de granéis sólidos. Neste caso foi proposto uma heurística baseada no algoritmo de enxame de partículas multifásicas.

Vários artigos sobre PAB são encontrados na literatura. O artigo [Correcher et al., 2019b] exibe um PAB onde contempla restrições espaciais que limitam o trafego de navios, foi proposto um PLIM e uma heurística de busca local iterada. Um PAD integrado com um problema de atribuição de guindastes de cais é apresentado no artigo [Correcher et al., 2019a] onde foi apresentado um PLIM e um procedimento iterativo associado a um algoritmo de *branch and cut*. No artigo [Zhang et al., 2020] é apresentado um PLIM para resolver PAB em canais de tráfego bidirecional com profundidade afetada pelas marés. Adicionalmente, também foi desenvolvido uma heurística baseada em algoritmo genético. PAB em portos de exportação de containers é o assunto abordado no artigo [Nishi et al., 2020], onde um modelo de PLIM e um algoritmo de decomposição lagrangiana foram propostos. Destacamos também os trabalho sobre o PAD: [Ernst et al., 2017], que tratou restrições associadas a maré; [Venturini et al., 2017], que sequenciou posições de atracação em terminais de contêineres; [Iris et al., 2017], que utilizou uma heurística baseada em *Adaptive Large Neighborhood Search* (ALNS) para sequenciar guindastes de cais em terminais de contêineres; [Mauri et al., 2016], que propôs modelos discreto e contínuo do PAB e uma heurística beseada em ALNS; e [Agra e Oliveira, 2018], que apresentou uma heurística baseada em *Rolling Horizon Heuristic* (RHH) para o PAD.

Tratando do problema de roteamanto, destacamos os seguintes trabalhos: [Lahyani et al., 2019], que aborda o tema de PRV com vários depósitos, onde foi proposto uma heurística denominada como híbrida adaptativa para grandes vizinhanças; Problema similar foi tratado no artigo [Ruiz et al., 2019], onde foi apresentado a proposta de uma heurística baseada em algoritmo genético. Tratando de situações semelhantes às investigadas neste trabalho, os seguintes trabalhos que resolveram o PRV com VNS: [Vidović et al., 2017], que resolveu o problema com multi depositos; o artigo [Belhaiza et al., 2017], que abordou o tema de PRV com múltiplas janelas de tempo utilizando uma heurística hibrida baseada em VNS com algoritmo genético; artigo [Belhaiza et al., 2017], que propôs a integração do planejamento da produção, gerenciamento de estoque, cronogramas de entrega e PRV aplicados às cadeias de suprimentos e resolvida com a aplicação de uma heurística baseada em VNS; no artigo [Chagas et al., 2020] é considerado uma frota de veículos que deve coletar itens em uma região e entrega-los em outra região. O artigo utilizou o método *Ejection Chain* para variar a vizinhança da busca.

3. Descrição do Problema

A Figura 1 ilustra um pequeno parque de esteiras onde temos origens e destinos (representados por retângulos), as arestas direcionadas representando as esteiras e os circulos pretos representando alimentadores de esteiras (usados para passar o minério de uma esteira para outra). Adicionalmente, note que existem duas esteiras TR1, partindo de um mesmo nó. Nesse caso, temos uma esteira que pode funcionar nos dois sentidos, onde um sentido bloqueia o outro. Existe também o alimentador móvel de esteira (ALM), que pode receber de duas esteiras, uma de cada vez, e enviar para outra esteira. Para exemplificar, considere a rota da Origem 2 até o Destino 2, passando pela seguinte sequência de equipamentos TR2-ALM1-TR10-TR13-TR15-TR17, representado pela linha

verde. Note que esta rota impossibilita o acionamento da rota Origem 3-TR3-ALM1-TR10-TR13-TR15-TR17-Destino 2, que é representado por uma linha vermelha contínua, na parte possível, e uma linha vermelha tracejada, na parte bloqueada. Como o alimentador de esteira móvel ALM1 seleciona um fluxo por vez, ou o fluxo de minério ocorre da esteira TR2 para esteira TR10 ou da esteira TR3 para esteira TR10. Outra forma de ocorrer bloqueios são as esteiras que possuem dois sentidos de funcionamento. Na Figura1 as esteiras TR1, TR2, TR3 e TR4 possuem dois sentidos de funcionamento, sendo usado apena um sentido por vez. Há alguns casos em que mais de uma esteira enviam minério para uma mesma esteira ao mesmo tempo, este caso é denominado interseção de rota. Um exemplo de interseção de rota são as esteiras TR6 e TR1 que enviam minério para TR5. Neste caso temos duas possibilidades: i) se TR6 e TR1 estiverem transportando minério de mesmo tipo, as duas podem se reunirem na TR5, desde que a soma das vazões das TR6 e TR1 não exceda a capacidade da TR5; ii) se TR6 e TR1 estiverem transportando minério de tipo diferente, TR5 só poderá receber de uma esteira e a outra ficará bloqueada durante a operação dessa demanda. Outro motivo de bloqueios entre rotas é o fato de uma origem não ser capaz de atender a mais de uma rota simultaneamente. Por fim, uma rota é capaz de atender a uma demanda por vez.

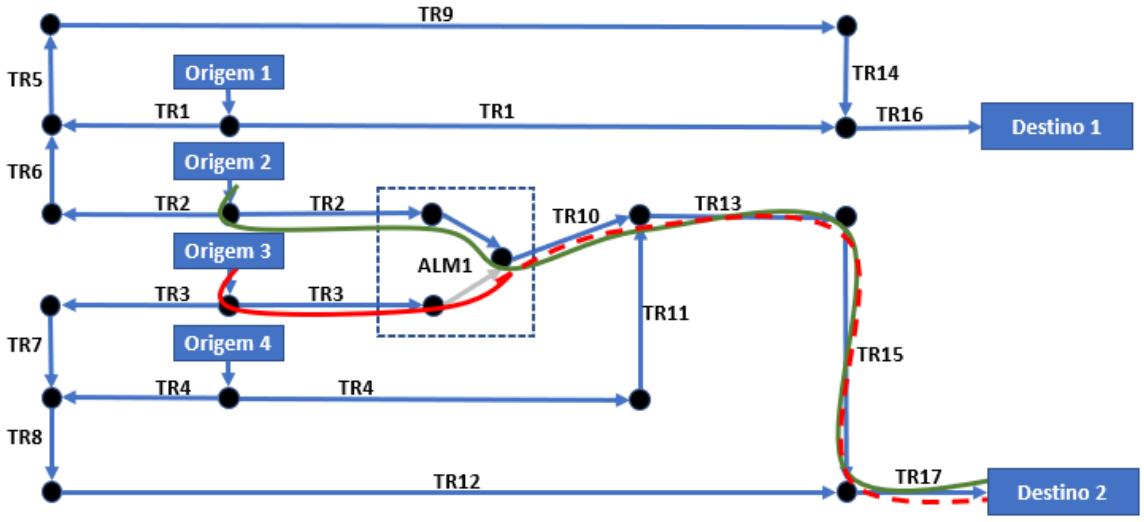

Figura 1: Cenário de Rotas com bloqueio

No artigo de [Nishi et al., 2020] foi apresentado um PAB, sequenciando os atendimentos de navios de forma similar ao PRVVDJT. No presente artigo é proposto a modelagem de um PAD, onde é considerado que um conjunto de rotas é similar a um conjunto de veículos, as demandas a serem atendidas pelas rotas são tratadas como os clientes. Após o atendimento do cliente, o veículo vai para uma garagem virtual sinalizando do fim o atendimento, sendo assim a modelagem deverá incluir uma rota virtual para cada rota existente para representar o fim de atendimento da demanda.

A Figura 2 mostra um grafo ilustrando um cenário com 3 rotas e 4 demandas. Assim, o grafo possui 3 nós para cada rota, 4 nós para cada demanda e mais 3 nós para cada rota virtual identificando a finalização de cada atendimento. A rota 1, atende as demandas 5 e 4 concluindo o atendimento na rota virtual 8, a rota 3 atende as demandas 6 e 7 concluindo o atendimento na rota virtual 10 e a rota 2 não será usada.

4. Modelo Matemático Proposto

Foi realizado a modelagem do PAD, onde a função objetivo é minimizar o tempo de execução das demandas. A seguir são apresentadas as notações, parâmetros e variáveis do modelo:

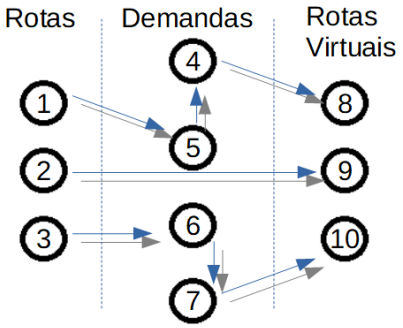

Figura 2: Grafo de atendimento de demandas

Parâmetros:

nr: número total de rotas de equipamentos;

nd: número de demandas;

a_i: momento de chegada da demanda i;

vr_k: vazão do equipamento de menor capacidade da rota k;

cd_i: carga da demanda i;

md_i: tipo de minério da demanda i;

od_i: origem da demanda i;

odr_k: combinação de origem e destino da rota k;

odd_i: combinação de origem e destino da demanda i;

b_{kr}: matriz binária de bloqueio, sendo 1 se a rota k bloqueia o uso da rota r;

it_{kr}: matriz binária de interseção, sendo 1 se a rota k e r podem trabalhar juntas para atender uma demanda. Cabe destacar que as restrições devem limitar a capacidade de vazão das rotas, pela capacidade da esteira mais restrita.

Conjuntos

R: conjunto de rotas reais $1...nr$;

D: conjunto de demandas $(nr + 1)...(nr + nd)$;

RV: conjunto de demandas e rotas virtuais $(nr + 1)...(2 * nr + nd)$;

T: conjunto com todos os índices;

Variáveis de Decisão

X_{ij}^k: variável binária. Assume valor 1 se a demanda j é atendida após a demanda i na rota k, e 0 caso contrário;

DA_i: variável binária. Assume valor 1 se a demanda i foi atendida por alguma rota, e 0 caso contrário;

DAR_i: variável inteira que define qual rota k atenderá a demanda i;

TI_i: momento do início do atendimento da demanda i;

TF_i: momento do fim do atendimento da demanda i;

$TMAX$: momento em que termina o atendimento da lista de demandas;

A Figura 3 ilustra os conjuntos do modelo.

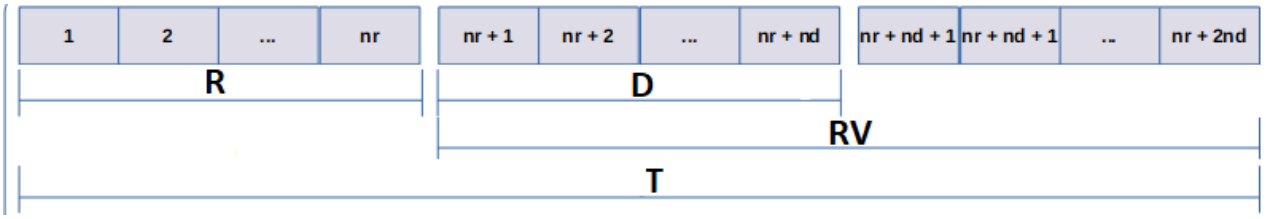

Figura 3: Conjuntos de elementos do modelo

Função Objetivo

$$\min : TMAX \tag{1}$$

Restrições

$$DA_i = 1; \quad \forall i \in D \tag{2}$$

$$DA_i = \sum_{k \in R} \sum_{j \in RV} x_{ij}^k; \quad \forall i \in D \tag{3}$$

$$\sum_{\substack{k \in R: \\ odr_k = odd_i}} \left(\left(\sum_{j \in D} x_{ij}^k \right) + x_{i,(k+nd+nr)}^k \right) = DA_i; \quad \forall i \in D \tag{4}$$

$$\left(\sum_{j \in D} x_{kj}^k \right) + x_{k,(k+nr+nd)}^k = 1; \quad \forall k \in R \tag{5}$$

$$\left(\sum_{i \in D} x_{i,(k+nr+nd)}^k \right) + x_{k,(k+nr+nd)}^k = 1; \quad \forall k \in R \tag{6}$$

$$\left(\sum_{i \in D} x_{(k+nr+nd),i}^k \right) + x_{(k+nr+nd),k}^k = 0; \quad \forall k \in R \tag{7}$$

$$\left(\sum_{j\in D} x^k_{jk}\right) + x^k_{(k+nr+nd),k} = 0; \quad \forall k \in R \tag{8}$$

$$\left(\left(\sum_{j\in D} x^k_{ij}\right) + x^k_{i,(k+nr+nd)}\right) - \left(\left(\sum_{j\in D} x^k_{ji}\right) + x^k_{ki}\right) = 0, \forall k \in R; \quad \forall i \in D \tag{9}$$

$$\textbf{Se } x^k_{ji} = 1 \textbf{ então } TI_i + (cd_i/vr_i) \leq TI_j; \quad \forall k \in R, \forall i \in D, \forall j \in D \tag{10}$$

$$\begin{aligned}\textbf{Se } (b_{kr} = 1 \textbf{ ou } ((it_{kr} = 1) \textbf{ e } (md_i \neq md_j)) \textbf{ ou } od_i = od_j) \textbf{ e} \\ (DAR_i = k) \textbf{ e } (DAR_j = r) \textbf{ e } (TI_i \leq TI_j) \textbf{ então } (TF_i \leq TI_j); \\ \forall k, r \in R : k \neq r, \forall i, j \in D : i \neq j\end{aligned} \tag{11}$$

$$\begin{aligned}\textbf{Se } (b_{kr} = 1 \textbf{ ou } ((it_{kr} = 1) \textbf{ e } (md_i \neq md_j)) \textbf{ ou } od_i = od_j) \textbf{ e} \\ (DAR_i = k) \textbf{ e } (DAR_j = r) \textbf{ e } (TI_i \geq TI_j) \textbf{ então } (TF_i \geq TI_j); \\ \forall k, r \in R : k \neq r, \forall i, j \in D : i \neq j\end{aligned} \tag{12}$$

$$\textbf{Se } DAR_i = k \textbf{ então } TF_i = TI_i + (cd_i/vr_i), \forall k \in R, \forall i \in D \tag{13}$$

$$TI_i \geq a_i, \forall i \in D \tag{14}$$

$$TMAX \geq TF_i, \forall i \in D \tag{15}$$

$$\textbf{Se } x^k_{ji} = 1 \textbf{ então } DAR_i = k, \forall k \in R, \forall i \in D, \forall j \in RV \tag{16}$$

Descrição:

A função objetivo, representada pela Expressão (1) busca minimizar o tempo de atendimento de todas as demandas. As restrições da Equação (2) garantem que todas as demandas *i* sejam atendidas. As restrições da Equação (3) atribuem valor à variável DA_i, onde essa variável será 1 se a demanda *i* for atendida e 0 caso contrário. A Equação (4) garante que a demanda *i* seja atendida por apenas uma rota. Além disso, esta restrição obriga as rotas a atenderem somente as demandas que possuem a mesma combinação de origem e destino, orientadas pelos parâmetros odr_k e odd_i. Uma primeira demanda a ser atendida é definida pelas restrições da Equação (5), já as restrições da Equação (6) determinam que uma demanda será a última a ser atendida em cada rota, caso a rota seja utilizada. As restrições das Equações (7) e (8) definem que o fluxo de atendimento das demandas comece em rotas reais e terminem em rotas virtuais respectivamente e as restrições da Equação (9) garantem a conservação do fluxo nas demais demandas. Na Equação (10) é calculado a variável decisão TI_i que é o momento de atendimento das demandas respeitando a sequência de demandas, para isto é calculado o tempo de execução das demandas dividindo a carga cd_i pela vazão vr_k do equipamento de menor capacidade da rota k o que resulta em um tempo medido em horas. Note

que essa expressão possui uma condição lógica, fazendo com que se torne não linear. Contudo, restrições lógicas são automaticamente linearizadas pelo software CPLEX durante a resolução do modelo. Assim, manteremos a apresentação das restrições exatamente com implementada no solver. Isso será feito para as demais restrições com expressões lógicas. As Equações (11) e (12) utilizam a matriz de bloqueio b_{kr} para não permitir que rotas *k* e *r* operem simultaneamente e a matriz de interseção it_{kr} representam as rotas que podem operar simultaneamente, se o tipo de minério determinado pelos parâmetros md_i e md_j for do mesmo tipo, além disto as origens das demandas od_i e od_j não podem ser iguais em caso de operação simultânea, dessa forma garante o sequenciamento correto dos tempos TI_i e TF_i. A Equação (13) calcula TF_i, que é o momento do fim de atendimento da demanda *i*. A Equação (14) define que o momento do inicio do atendimento TI_i é após o momento de chegada da demanda a_i. A Equação (15) determina que TMAX é maior igual que o momento final de atendimento de todas as demandas. Por fim, as restrições da Equação (16) atribuem a variável DAR_i a rota que atendem à demanda *i*.

5. Heurística

O método escolhido para o desenvolvimento da heurística foi o VNS. Para cada solução corrente existe uma Lista de Possibilidades (LP) para o atendimento das demandas por novas rotas. Assim, cada substituição de uma rota está no conjunto de vizinhos da solução. Quando uma demanda é atendida por uma rota obtém-se o tempo de serviço (TS). Para a variação de vizinhança, caracteristica do VNS, considere que a variável K é incrementada a cada iteração que a função objetivo não diminuiu e é reiniciada com o valor 1 quando ocorre diminuição da função objetivo ou quando K for maior do que o número de demandas: *i*) caso K seja par, a heurística ordena a solução corrente por TS de forma crescente e altera as rotas das K primeiras demandas; *ii*) caso K seja impar, a solução é ordenada de forma decrescente por TS e altera as rotas das K primeiras demandas.

Na Figura 4 observamos o pseudocódigo da heurística proposta nesse trabalho. Na linha 5 um loop principal é iniciado e o critério de parada é definido pela variável *iteracoes*. Na linha 6 outro loop é iniciado, enquanto a variável *K* for menor do que a variável *numero_demandas* o loop continuará. A linha 7 testa se a variável *K* é um número par, caso verdadeiro a variável *lista_demanda_alterar* é atribuída com o retorno da função *n_menor_tempo_servico* que possui como argumento de entrada a solução corrente (*s*) e a variável *K*. Assim, a função *n_menor_tempo_servico* ordena a solução *s* do menor para o maior TS e retorna para a variável *lista_demanda_alterar* os k primeiros item da lista ordenada. Caso o teste da linha 7 for falso, a heurística aplicará a função *n_maior_tempo_servico* para ordenar a solução *s* do maior para o menor TS e retornará para a variável *lista_demanda_alterar* os k primeiros itens da lista ordenada. Na linha 12 a variável *s'* é atribuída com o retorno da função *troca_rota* que possui as variáveis *s*, *lista_demanda_alterar* e *vizinhos* como argumento de entrada. Essa função substitui as rotas listadas em *lista_demanda_alterar* utilizando as possibilidades de atendimento de demandas listadas por *vizinhos*. Na linha 13 a variável *s"* é atribuída pelo retorno da função *sequenciamento_bloqueio* que possui como argumentos de entrada as variáveis *s'* , *matriz_bloqueio* e *matriz_intersecao*. Esta função realiza a sequência das demandas listadas na variável *s'* e não permite operações simultânea de rotas que se bloqueiam (*matriz_bloqueio*) ou se interseccionam (*matriz_interseção*) com tipos diferentes de material. A linha 14 testa se o retorno da função *ver_tmax_solucao* é menor do que a variável *tmax*, caso o teste for verdadeiro as variáveis *tmax*, *s*, *k* e *vizinhos* são atualizadas, caso contrário a variavel *k* é incrementada. Adicionalmente, se *k* for maior do que *numero_demandas* o algoritmo aceita uma solução que não melhore *tmax*, para tentar sair de um possível mínimo local. Esse processo é repetido até que o critério de parada da linha 5 ocorra.

```
Procedimento VNS
Início
    1.  S ← solucao_aleatória;
    2.  tmax←ver_tmax_solucao(s);
    3.  vizinhos ← verificar_vizinhos(s);
    4.  enquanto (iteracoes <= criterio_parada)
    5.          k ← 1;
    6.          enquanto (k <= numero_de_demandas)
    7.              Se( k MOD 2 == 0)
    8.                lista_demanda_alterar = n_menor_tempo_servico (S, k);
    9.              Senão
    10.               lista_demanda_alterar = n_maior_tempo_servico (S, k);
    11.             fim-se;
    12.             S' ← troca_rota (S, lista_demanda_alterar, vizinhos);
    13.             S'' ← sequenciamento_bloqueio (S', matriz_bloqueio, matriz_intersecao);
    14.             se(ver_tmax_solucao(S'') < tmax)
    15.                tmax←ver_tmax_solucao(S'');
    16.                S←S'';
    17.                k←1;
    18.                vizinhos ← verificar_vizinhos(S'');
    19.             senão
    20.                K++;
    21.             fim-se;
    22.             iteracoes ++;
    23.         fim-enquanto;
    24.         iteracoes ++;
    25.         S←S'';
    26.         vizinhos ← verificar_vizinhos(S'');
    27.     fim-enquanto;
    28. retorne S;
fim VNS
```

Figura 4: Pseudo código da heurística proposta

6. Resultados Numéricos

Para verificar o funcionamento do modelo PLIM proposto na Seção 4 foi utilizado um computador com um processador Intel Core (TM) i5 7ª geração de 2 núcleos e 8Gb de memória RAM executando a versão 12.8 do *solver* CPLEX. Além disto a heurística VNS proposta na Seção 5 foi implementada utilizando a linguagem C#. A subseção 6.1 exibe os resultados dos experimentos que comparam os resultados obtidos pelo CPLEX e pela heurística. A subseção 6.2 exibe os resultados de experimentos da heurística com instâncias maiores.

6.1. Exemplo de Teste

A Tabela 1 mostra o resultado dos experimentos realizados utilizando os dados do cenário da Figura 1. Como pode ser observado, nessa rede existem 8 rotas possíveis. Neste experimento foram resolvidas 8 instâncias, com um número crescente de demandas. Nos testes, o tempo de processamento do CPLEX foi limitado em 1 hora, definido após entrevista com operadores de um porto, e o critério de parada da heurística foi 5000 iterações. A coluna *Instância* é o identificador

Instância	N° demandas	CPLEX fun. objet.	CPLEX proc.	CPLEX GAP	VNS fun. objet.	VNS proc.
1	7	6,0h	2444s	0%	6h 51min	3,7s
2	8	7h 5min	3600s	3,1%	8h 51min	3,6s
3	10	11h 24min	3600s	65,0%	9h 19min	3,7s
4	17	20h 54min	3600s	85,4%	14h 19min	4,4s
5	50	-	-	-	41h 30min	4,5s
6	100	-	-	-	88h 48min	7,2s
7	150	-	-	-	132h 48min	12,14s
8	180	-	-	-	141h 30min	15,5s

Tabela 1: Comparativo do CPLEX com a Heurística

Instancia	Nº rotas por demandas	Tempo função objetivo	Tempo processamento
9	1	129h 54min	56s
10	2	36h 36min	67s
11	4	33h 18min	91s
12	7	29h 54min	91s
13	10	26h 42min	125s
14	15	23h 18min	142s

Tabela 2: Experimento em um cenário real

da instância gerada, a coluna *N° demandas* indica o numero de demandas utilizada na instância, a coluna *CPLEX fun. objet.* é o tempo da função objetivo que representa o tempo de atendimento de todas as demandas encontrado pelo CPLEX, a coluna *CPLEX proc.* é o tempo de processamento gasto pelo CPLEX da instância, a coluna *CPLEX GAP* mostra o desvio estimado até a solução ótima, a coluna *VNS fun. objet.* é o tempo de atendimento de todas as demandas encontrado pelo VNS e a coluna *VNS proc.* é o tempo de processamento gasto pelo VNS.

O CPLEX obteve a solução ótima apenas para a 1ª instância, com 7 demandas, e alcançou um resultado melhor do que a heurísitca apenas nas instâncias 1 e 2. Nas instâncias 3 e 4 a heurística obteve um resultado melhor do que o CPLEX, devido ao Gap das soluções do CPLEX. A partir da instancia 5 o CPLEX não conseguiu apresentar solução viável no prazo estabelecido, sendo assim só foram anotados os valores da função objetivo apresentados pela heurística.

6.2. Aplicação em um caso real

O cenário real utilizado nos experimentos foi um porto localização em Vitória-ES, Brasil. A combinação de todos os equipamentos instalados nesse porto formam aproximadamente 5000 rotas de equipamentos que transportam diferentes tipos de minério. A Tabela 2 exibe o resultado dos experimentos utilizando os dados do cenário real. A proposta é resolver uma lista de 180 demandas, porém a cada instância foi elevado o número de rotas possíveis para atender a cada demanda. Como esperado, foi observado que o número de possibilidade de rota por demanda é inversamente proporcional a função objetivo. Isto ocorre porque com mais opções de rotas é possível contornar os bloqueios das rotas, diminuindo assim o tempo máximo de atendimento. Adicionalmente, observa-se que a heurística demonstrou ser capaz de atender a necessidade do caso real de uma empresa. A coluna *Instância* é o identificador da instância gerada, a coluna *Nº rotas por demandas* indica o numero de rotas possiveis para cada demanda, a coluna *Tempo função objetivo* é o tempo da função objetivo que representa o tempo de atendimento de todas as demandas encontrado pelo VNS e a coluna *Tempo processamento* é o tempo de processamento gasto pelo VNS.

7. Conclusão

O sequenciamento do número de demandas que chegam em um terminal portuário de minério de ferro pode ser tratado como um Problema de Alocação de Recursos devido a quantidade finita de equipamentos onde se pretende executar uma lista de tarefas. Nesse trabalho, foi proposto um modelo de programação linear inteira e uma heurística, considerando as combinações de origens e destinos para cada rota onde todas as demandas devem ser atendidas. Como restrição, foi contemplando uma matriz de bloqueio que não permite a execução simultânea de rotas incompatíveis e uma matriz de interseção que mapeia as rotas que podem funcionar simultaneamente, desde que transportem o mesmo tipo de material. Para a solução das instância do PLIM proposto foi utilizado o software CPLEX. Observou-se que o CPLEX encontrou soluções apenas para instancias de pequeno porte e com poucas demandas. A heurística encontrou soluções de boa qualidade, quando comparada com a soluções obtidas pelo CPLEX. Porém, o método heurístico apresentou um tempo computacional relativamente baixo, possibilitando a sua aplicação em ambientes reais.

Referências

Agra, A. e Oliveira, M. (2018). Mip approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *European Journal of Operational Research*, 264(1):138–148.

Belassiria, I., Mazouzi, M., Elfezazi, S., e Torbi, I. (2019). A heuristic methods for stacker and reclaimer scheduling problem in coal storage area. In *2019 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, p. 1–7. IEEE.

Belhaiza, S., M'Hallah, R., e Brahim, G. B. (2017). A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, p. 1319–1326. IEEE.

Chagas, J. B., Silveira, U. E., Santos, A. G., e Souza, M. J. (2020). A variable neighborhood search heuristic algorithm for the double vehicle routing problem with multiple stacks. *International Transactions in Operational Research*, 27(1):112–137.

Correcher, J. F., Alvarez-Valdes, R., e Tamarit, J. M. (2019a). New exact methods for the time-invariant berth allocation and quay crane assignment problem. *European Journal of Operational Research*, 275(1):80–92.

Correcher, J. F., Van den Bossche, T., Alvarez-Valdes, R., e Berghe, G. V. (2019b). The berth allocation problem in terminals with irregular layouts. *European Journal of Operational Research*, 272(3):1096–1108.

de Paula, M. R., Boland, N., Ernst, A. T., Mendes, A., e Savelsbergh, M. (2019). Throughput optimisation in a coal export system with multiple terminals and shared resources. *Computers & Industrial Engineering*, 134:37–51.

Ernst, A. T., Oğuz, C., Singh, G., e Taherkhani, G. (2017). Mathematical models for the berth allocation problem in dry bulk terminals. *Journal of Scheduling*, 20(5):459–473.

Iris, Ç., Pacino, D., e Ropke, S. (2017). Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 105:123–147.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

Lahyani, R., Gouguenheim, A.-L., e Coelho, L. C. (2019). A hybrid adaptive large neighbourhood search for multi-depot open vehicle routing problems. *International Journal of Production Research*, 57(22):6963–6976.

Mauri, G. R., Ribeiro, G. M., Lorena, L. A. N., e Laporte, G. (2016). An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Computers & Operations Research*, 70:140–154.

Menezes, G. C., Mateus, G. R., e Ravetti, M. G. (2016). A hierarchical approach to solve a production planning and scheduling problem in bulk cargo terminal. *Computers & Industrial Engineering*, 97:1–14.

Menezes, G. C., Mateus, G. R., e Ravetti, M. G. (2017). A branch and price algorithm to solve the integrated production planning and scheduling in bulk ports. *European Journal of Operational Research*, 258(3):926–937.

Nishi, T., Okura, T., Lalla-Ruiz, E., e Voß, S. (2020). A dynamic programming-based matheuristic for the dynamic berth allocation problem. *Annals of operations research*, 286(1):391–410.

on Trade, U. N. C. e Development (2018). *Review of Maritime Transport 2018*. URL `https://www.un-ilibrary.org/content/publication/cd4440fc-en`.

Pratap, S., Daultani, Y., Tiwari, M., e Mahanty, B. (2018). Rule based optimization for a bulk handling port operations. *Journal of Intelligent manufacturing*, 29(2):287–311.

Pratap, S., Kumar B, M., Saxena, D., e Tiwari, M. K. (2016). Integrated scheduling of rake and stockyard management with ship berthing: a block based evolutionary algorithm. *International Journal of Production Research*, 54(14):4182–4204.

Ruiz, E., Soto-Mendoza, V., Barbosa, A. E. R., e Reyes, R. (2019). Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm. *Computers & Industrial Engineering*, 133:207–219.

Tang, L., Sun, D., e Liu, J. (2016a). Integrated storage space allocation and ship scheduling problem in bulk cargo terminals. *IIE Transactions*, 48(5):428–439.

Tang, M., Gong, D., Liu, S., e Zhang, H. (2016b). Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem. *Advances in Production Engineering & Management*, 11(4).

Unsal, O. e Oguz, C. (2019). An exact algorithm for integrated planning of operations in dry bulk terminals. *Transportation Research Part E: Logistics and Transportation Review*, 126:103–121.

Venturini, G., Iris, Ç., Kontovas, C. A., e Larsen, A. (2017). The multi-port berth allocation problem with speed optimization and emission considerations. *Transportation Research Part D: Transport and Environment*, 54:142–159.

Vidović, M., Popović, D., Ratković, B., e Radivojević, G. (2017). Generalized mixed integer and vns heuristic approach to solving the multisize containers drayage problem. *International Transactions in Operational Research*, 24(3):583–614.

Zhang, B., Zheng, Z., e Wang, D. (2020). A model and algorithm for vessel scheduling through a two-way tidal channel. *Maritime Policy & Management*, 47(2):188–202.