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

UM ALGORITMO CONSTRUTIVO PARA O PROBLEMA DE SEQUENCIAMENTO DE RECUPERADORAS DE MINÉRIO

João Daniel Ferreira de Jesus-Junior[1,2], **Luciano Perdigão Cota**[1,3], **Marcone Jamilson Freitas Souza**[1,4]

[1]Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração,
Universidade Federal de Ouro Preto e Instituto Tecnológico Vale,
CEP 35.400-000, Ouro Preto (MG), Brasil

[2]Vale S.A,
CEP 29.090-860, Vitória (ES), Brasil

[3]Instituto Tecnológico Vale,
CEP 35.400-000, Ouro Preto (MG), Brasil

[4]Departamento de Computação – Universidade Federal de Ouro Preto (UFOP),
CEP 35.400-000, Ouro Preto (MG), Brasil

joao.daniel@vale.com, luciano.p.cota@itv.org, marcone@ufop.edu.br

RESUMO

Este trabalho trata um problema de sequenciamento de recuperadoras do Porto de Tubarão, da Vale. Neste problema, tem-se um conjunto de navios a serem embarcados e para cada navio, existe um conjunto de pilhas pré-definidas a serem embarcadas. O embarque das pilhas deve seguir uma ordem programada. As pilhas estão estocadas em áreas denominadas pátios de estocagem e neles existem recuperadoras que retiram o minério das pilhas e as enviam para embarque nos navios. O objetivo é sequenciar as recuperadoras minimizando o *makespan*. Para resolvê-lo, é proposto um algoritmo construtivo para apoiar a tomada de decisão, que atualmente é manual. Esse algoritmo avalia, dentre uma lista de tarefas candidatas, aquelas que possuem o menor tempo de término, tendo por objetivo a minimização do tempo de conclusão da última pilha recuperada. Nos experimentos computacionais foi utilizada uma instância real de abril de 2020. Como resultado, foi gerado um sequenciamento factível instantaneamente.

PALAVRAS CHAVE. Sequenciamento de Recuperadoras de Minério, Sequenciamento de Tarefas, Otimização, Portos.
Tópicos: PO na Indústria, OA – Outras aplicações em PO.

ABSTRACT

This paper addresses a reclaimer scheduling problem in Vale's Tubarão Port. In this problem, there is a set of ships to be boarded. For each ship, there is a set of predefined stockpiles to be boarded and a given order. The stockpiles are stored in storage yard areas where are the reclaimers. The material is reclaimed from stockpiles by reclaimers and sent to ships. The problem aims to sequence reclaimers in such a way that the makespan is minimized. For solving it, a constructive algorithm is proposed to support decision-making, which is currently manual. This algorithm evaluates, among a list of candidate jobs, those with the shortest completion time, to minimize the completion time of the last stockpile reclaimed. In computational experiments, a real instance of April 2020 was used. As a result, feasible scheduling was generated instantly.

KEYWORDS. Iron Ore Reclaimer Scheduling, Scheduling, Optimization, Ports.
Paper topics: OR in Industry. Other applications in OR

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

1. INTRODUÇÃO

A relevância da mineração na indústria nacional fez com que os primeiros trabalhos de otimização fossem aplicados ao setor de mineração na década de 1960 [Amaral e Pinto, 2010]. Ainda hoje existe uma vasta extensão teórica e prática a ser explorada. Métodos de "tentativa-e-erro" ainda são empregados no processo de planejamento e tomada de decisão para muitos processos de mineração [Pantuza et al., 2009]. No contexto de operações de portos, no qual este trabalho está inserido, os métodos de otimização objetivam o aumento de eficiência das operações nos portos, tratando principalmente problemas relacionados à saturação das estruturas, custos, armazenagem, estadia e fila de navios [Cereser, 2018].

Muitos trabalhos relacionados a portos foram desenvolvidos focados na otimização da alocação de navios em berços e na movimentação de contêineres [Pratap et al., 2016].

Lopes et al. [2011] tratam um problema de alocação de berços aplicando o método de solução *Greedy Randomized Adaptative Search Procedure* (GRASP) [Resende e Ribeiro, 2014] combinado com *Path Relinking* (PR) [Glover, 1996] para intensificação da busca. No método, cada iteração do GRASP é composta por uma fase construtiva e por uma fase de busca local que, baseada na solução construída, aplica movimentos iterativos até que o ótimo local seja encontrado. Em seguida, o PR é aplicado como uma estratégia de intensificação, explorando as trajetórias que conectam a solução da iteração atual à melhor solução obtida até então.

Chen et al. [2014] tratam o problema de sequenciamento de movimentação de contêineres aplicando um modelo de programação matemática mais simples do que o encontrado na literatura até então e para solução em *solvers* padrão. Sammarra et al. [2007] tratam o mesmo problema, porém com aplicação de métodos heurísticos: o problema é decomposto em um problema de roteamento e sequenciamento. O problema de roteamento é resolvido com a aplicação da metaheurística Busca Tabu [Souza, 2012], enquanto o problema de sequenciamento é resolvido com uma técnica de busca local.

Com a crescente demanda de granéis sólidos impulsionados, principalmente pela China, o que inclui minério de ferro e carvão, vários trabalhos sobre sequenciamento de movimentação de granéis sólidos em portos têm sido intensificados.

Hu e Yao [2012] tratam um problema de sequenciamento de movimentação de minério em portos para minimizar o *makespan*, isto é, o tempo de término de recuperação da última pilha de minério. Um modelo de programação matemática para este problema é formulado, porém por ele ser considerado de natureza NP-difícil, algoritmos genéticos são também desenvolvidos para resolver o problema. Diferentemente do presente trabalho, os autores não consideram a precedência de recuperação de pilhas e nem a data de disponibilidade de recuperação das pilhas, entre outras características. Angelelli et al. [2016], por sua vez, investigam uma série de problemas relacionados ao sequenciamento de tarefas de recuperação em um porto de exportação de carvão. Os autores assumem algumas características básicas para o problema de sequenciamento e provam que ele é de natureza NP-difícil para diferentes variações. Diferentes algoritmos são propostos com execução em tempos polinomiais. Diferentemente do presente trabalho, os autores consideram que as recuperadoras são idênticas: possuem a mesma velocidade de locomoção e taxa de recuperação e não consideram a data de disponibilidade das tarefas.

Em Burdett et al. [2019], um porto de exportação de carvão é modelado com todas as suas características reais: chegada de trens a serem descarregados, chegada de navios a serem embarcados, a relação entre as máquinas de pátio como colisão e movimentação sobre os mesmos trilhos e preempção de tarefas. O problema é modelado como um *job shop* flexível e resolvido com algoritmos heurísticos baseados em *Simulated Annealing* [Kirkpatrick et al., 1983] e Algoritmos

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

Genéticos [Holland, 1992]. Para testar os métodos de solução propostos, várias instâncias de testes reais foram usadas.

Diferentemente de Burdett et al. [2019], que considera um problema de sequenciamento em um porto de movimentação de carvão mineral, o presente trabalho trata o sequenciamento de recuperação de minério com as características do Porto de Tubarão (ES), com o objetivo de minimizar o *makespan*. Para resolvê-lo, propõe-se um algoritmo de despacho que usa uma função guia avaliando os tempos de término das tarefas. Esse algoritmo foi aplicado a uma instância real do problema e mostrou-se capaz de gerar uma solução rapidamente.

O restante deste trabalho está organizado como segue. Na Seção 2, o problema é caracterizado e formulado. Na Seção 3 é apresentado o algoritmo de geração de solução inicial. Na Seção 4 são exibidos os resultados obtidos pelos experimentos computacionais e finalmente na Seção 5 são apresentadas as conclusões em relação à aplicação do algoritmo proposto. Sugestões de trabalhos futuros também são apresentadas.

2. CARACTERIZAÇÃO DO PROBLEMA

O problema de sequenciamento de recuperadoras de minério abordado neste trabalho possui as seguintes características:

(a) Há um conjunto SH de navios a serem embarcados, $SH = \{1, \cdots, |SH|\}$;

(b) Há um conjunto P de pilhas de minério a serem embarcadas, $P = \{1, \cdots, |P|\}$;

(c) Há um conjunto Y de pátios de estocagem onde as pilhas são estocadas, $Y = \{1, \cdots, |Y|\}$;

(d) Há um conjunto B_l de balizas do pátio $l \in Y$, $B_l = \{1, \cdots, |B_l|\}$;

(e) A localização de cada pilha $a \in P$ do pátio $l \in Y$ é representada no intervalo $[a^l_{ini}, a^l_{end}]$, onde a^l_{ini} e a^l_{end} são, respectivamente, as balizas inicial e final da pilha a do pátio l, em metros, sendo $1 \leq ini < end \leq |B_l|$;

(f) A localização média de uma pilha $a \in P$ no pátio $l \in Y$ é dada por $a^l_{med} = (a^l_{ini} + a^l_{end})/2$;

(g) Há um conjunto RC de máquinas recuperadoras que recuperam as pilhas do conjunto P e as enviam para os navios do conjunto SH, $RC = \{1, \cdots, |RC|\}$;

(h) Cada recuperadora $i \in RC$ possui uma velocidade de locomoção $speed_i$, em m/min, e uma taxa de recuperação $rate_i$, em ton/h;

(i) Uma recuperadora $i \in RC$ é elegível para recuperar uma pilha $a \in P$ do pátio $l \in Y$, e escrevemos, $e_{il} = 1$, se essa recuperadora i se locomove sobre os trilhos de uma berma que delimita o pátio l; caso contrário, $e_{il} = 0$. Uma berma é uma área que delimita dois pátios adjacentes;

(j) Cada pilha $a \in P$ possui uma quantidade $size_a$ de minério, em toneladas;

(k) Cada pilha $a \in P$ pode ser recuperada somente uma única vez;

(l) Cada navio $h \in SH$ possui um conjunto de pilhas reservadas $P_h \subseteq P$, sendo que cada pilha $a \in P_h$ pode ser embarcada somente em um navio previamente fixado e após a sua data de atracação. O embarque das pilhas em um navio deve seguir uma ordem de precedência $a' \prec a'' \prec \cdots \prec a^{|P_h|}$;

(m) Há um conjunto N de tarefas de recuperação de pilhas, $N = \{1, \cdots, |N|\}$. Cada tarefa $j \in N$ representa a recuperação completa de uma pilha $a \in P$ que será embarcada em um navio $h \in SH$ usando uma recuperadora $i \in RC$;

(n) Não é permitido preempção, isto é, uma vez iniciada a recuperação de uma pilha por uma recuperadora elegível, ela deve ser concluída por esta mesma recuperadora;

(o) Cada tarefa $j \in N$ possui uma data de disponibilidade r_j^h a partir da qual ela pode ser iniciada. A data de disponibilidade da tarefa equivale à data de atracação do navio $h \in SH$ para a qual a pilha $a \in P$, associada à tarefa $j \in N$, está prefixada;

(p) Cada tarefa $j \in N$ possui um tempo de processamento p_j^{ai} para ser executada. Tendo em vista que cada tarefa j representa a associação entre uma pilha $a \in P$, uma recuperadora $i \in RC$ e um navio $h \in SH$ predefinido, p_j^{ai} é calculado como $p_j^{ai} = size_a/rate_i$;

(q) Cada tarefa $k \in N$ associada à pilha a_2 possui um tempo de preparação $S_{ijk^{a_1 a_2}}$, que é o tempo necessário para a recuperadora $i \in RC$ se deslocar após ter executado a tarefa $j \in N$ associada à pilha a_1. Esse tempo de preparação é dado por $S_{ijk}^{a_1 a_2} = d_{a_1 a_2}/speed_i$, sendo $d_{a_1 a_2}$ a distância entre as localizações médias das pilhas a_1 e a_2, em metros;

(r) O tempo de locomoção da recuperadora $i \in RC$ para se posicionar na primeira tarefa k associada à pilha a_1 a ser executada, representado por $s_{i0k}a_0a_1$, em que 0 indica uma tarefa fictícia de recuperação de uma pilha a_0 cuja localização é $[0, 0]$ e $size_{a_0} = 0$;

O objetivo deste problema é encontrar um sequenciamento das $|P|$ pilhas nas $|RC|$ recuperadoras de forma a minimizar o *makespan*, denotado por $C_{\max}$, isto é, o tempo máximo de conclusão do sequenciamento. Este tempo máximo é o instante de tempo em que a última recuperadora recuperou a última pilha.

Para melhor compreensão do problema, considere como exemplo um porto com dimensões simplificadas como ilustrado na Figura 1. O porto possui três pátios de estocagem (Y1, Y2 e Y3), duas recuperadoras de minério (RC01 e RC02), dois berços de atracação (BE01 e BE02) e dois carregadores de navios (SL01 e SL02). Seis pilhas estão disponíveis para embarque (P01, P02, P03, P04, P05 e P06).

As características das pilhas estão contidos na Tabela 1. Na coluna "Pilha" estão presentes as identificações das pilhas. Na coluna "Pátio" são mostrados os pátios onde as pilhas estão estocadas. A coluna "Localização" exibe a localização de cada pilha, composta pelas posições das balizas inicial e final, respectivamente, em metros. A coluna "Localização média" mostra a localização média de cada pilha, em metros. A coluna "Quantidade", por sua vez, exibe a quantidade de material em cada pilha, em toneladas.

A Tabela 2 mostra as características das recuperadoras. A coluna "Recuperadora" exibe as identificações de cada recuperadora. Nas colunas "Velocidade" e "Taxa de recuperação" são exibidas as velocidades de locomoção e taxa de recuperação, em metros/minuto e toneladas/hora, respectivamente.

A Tabela 3 exibe a elegibilidade de cada recuperadora em recuperar pilhas em cada pátio. A coluna "Recuperadora" e "Pátio" mostram as identificações das recuperadoras e os pátios, respectivamente. A coluna "Elegibilidade" indica se a recuperadora i é elegível ou não para recuperar uma pilha no pátio l. O valor e_{il} = 1 indica que a recuperadora é elegível e o valor 0, caso contrário.

A Tabela 4 exibe as características dos navios a serem embarcados. As colunas "Navio", "Berço de Atracação" e "Data de Atracação" exibem as identificações, os berços de atracação e as

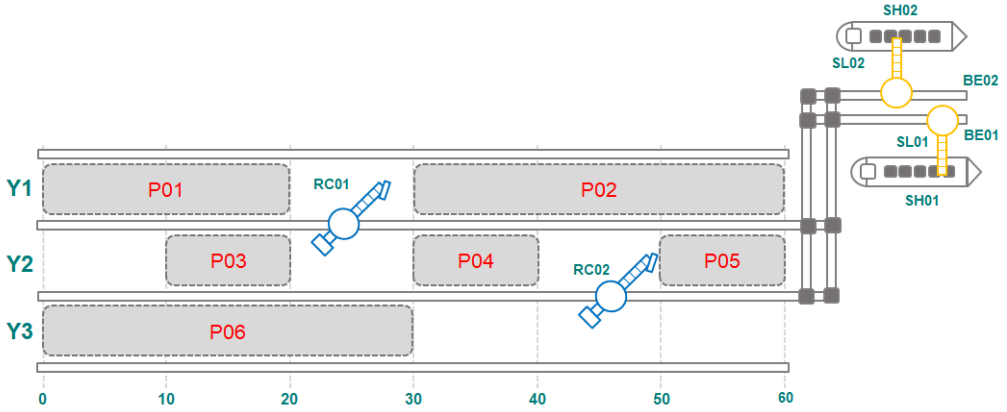


Figura 1: Exemplo de carregamento em um porto

Tabela 1: Características das pilhas

Pilha a	Pátio l	Localização $[a^l_{ini}, a^l_{end}]$	Localização média (m) a^l_{med}	Quantidade (ton) $size_a$
P01	Y1	[0,20]	10	400
P02	Y1	[30,60]	45	600
P03	Y2	[10,20]	15	200
P04	Y2	[30,40]	35	200
P05	Y2	[50,60]	55	200
P06	Y3	[0,30]	15	600

Tabela 2: Características das recuperadoras

Recuperadora i	Velocidade (m/min) $speed_i$	Taxa de recuperação (ton/h) $rate_i$
RC01	12	50
RC02	10	40

Tabela 3: Elegibilidade das recuperadoras

Recuperadora i	Pátio l	Elegibilidade e_{il}
RC01	Y1	1
RC01	Y2	1
RC01	Y3	0
RC02	Y1	0
RC02	Y2	1
RC02	Y3	1

datas de atracação, respectivamente. A coluna "Pilhas Reservadas" exibe as pilhas reservadas para cada navio na sequência de embarque previamente estabelecida.

Tabela 4: Características dos navios a serem embarcados

Navio	Berço de Atracação	Data de Atracação	Pilhas Reservadas
SH01	BE01	0	P01, P06 e P04
SH02	BE02	5	P03, P02 e P05

Na Figura 2 é ilustrada uma solução do problema, que consiste em um sequenciamento das seis pilhas nas duas recuperadoras.

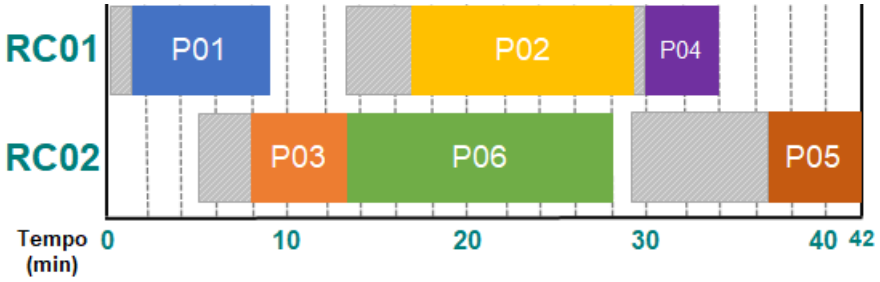


Figura 2: Exemplo de uma solução do RCSP

Como pode ser observado na Figura 2, na recuperadora RC01 foram alocadas as pilhas P01, P02 e P04, respectivamente. As partes hachuradas representam os tempos de deslocamento das recuperadoras entre as tarefas. Para a obtenção desse sequenciamento foram utilizados os dados das tabelas previamente apresentadas. Nessa solução, as recuperadoras RC01 e RC02 demandaram, respectivamente, 34 e 42 minutos para concluírem o embarque de todos os navios. Assim, o *makespan* dessa solução é 42.

3. ALGORITMO HEURÍSTICO PROPOSTO

Para construção de uma solução inicial, é proposto um algoritmo de despacho, nomeado DRC (das iniciais em inglês *Dispatch Reclaimer*). O algoritmo DRC avalia uma lista de tarefas candidatas LC, sendo que uma tarefa é considerada como candidata se a associação pilha a, localizada no pátio l, e recuperadora i, é factível, ou seja, se $e_{il} = 1$. Seu pseudocódigo é mostrado no Algoritmo 1.

O Algoritmo 1 recebe como entrada o conjunto Y de pátios, o conjunto P de pilhas, o conjunto RC de recuperadoras, o conjunto SH de navios e as funções *findRC*$(.)$ e *findP*$(.)$. A função *findRC(j)* retorna a recuperadora i associada à tarefa j passada como parâmetro. A função *findP*(j) retorna a pilha a associada à tarefa j passada como parâmetro.

Na linha 1 do Algoritmo 1, a função *ObterTarefasCandidatas*$(.)$ retorna um conjunto LC de tarefas candidatas. Após a obtenção das tarefas candidatas, a função *ProcessaTarefasCandidatas*$(.)$ faz o processamento destas tarefas e retorna na linha 2 um conjunto LC' de tarefas candidatas processadas. Este processamento consiste no cálculo do tempo de preparação, data de início e tempo de conclusão de cada tarefa tarefa candidata. Na linha 3 a solução inicial é inicializada com um conjunto vazio. Na linha 4, por sua vez, um conjunto P' de pilhas é inicializado com todas as pilhas P a serem recuperadas. O processo iterativo (linhas 5 a 13) ocorre enquanto existirem pilhas

Algoritmo 1: DRC

Entrada: Conjunto Y de pátios, Conjunto P de pilhas, Conjunto RC de recuperadoras, Conjunto SH de navios, funções *findRC*$(.)$ e *findP*$(.)$
Saída: Solução inicial s

1 $LC \leftarrow$ *ObterTarefasCandidatas*(Y, P, RC, SH);
2 $LC' \leftarrow$ *ProcessaTarefasCandidatas*(LC, Y, P, RC, SH);
3 $s \leftarrow \emptyset$;
4 $P' \leftarrow P$;
5 **enquanto** $(|P'| > 0)$ **faça**
6 **para cada** $(i \in RC)$ **faça**
7 $LC_i \leftarrow \{j \in LC' \mid$ *findRC*$(j) = i\}$;
8 Selecione $j \in LC_i$, sendo $j = \arg\min\{C_j \mid j \in LC_i\}$;
9 $s \leftarrow s \cup \{j\}$;
10 $LC' \leftarrow$ *ReprocessaTarefasCandidatas*(LC', s);
11 $P' = P' \setminus \{$*findP*$(j)\}$;
12 **fim**
13 **fim**
14 **retorna** s;

a serem recuperadas e embarcadas. Para cada recuperadora i (linha 6), seleciona-se na linha 7 o conjunto de tarefas candidatas LC_i associadas à recuperadora i e na linha 8, a próxima tarefa j a ser sequenciada na recuperadora i. A próxima tarefa j a ser sequenciada na recuperadora i é aquela cujo tempo de conclusão é o menor dentre todas a tarefas do conjunto LC_i. Na linha 9, a tarefa j é inserida na solução parcial s e, em seguida, na linha 10, o conjunto LC' é atualizado, isto é, as tarefas são reprocessadas levando em conta as tarefas já sequenciadas. Na linha 11, a pilha associada à tarefa j selecionada na linha 8 é removida do conjunto de pilhas P' a serem recuperadas. Por fim, na linha 14, retorna-se a solução construída.

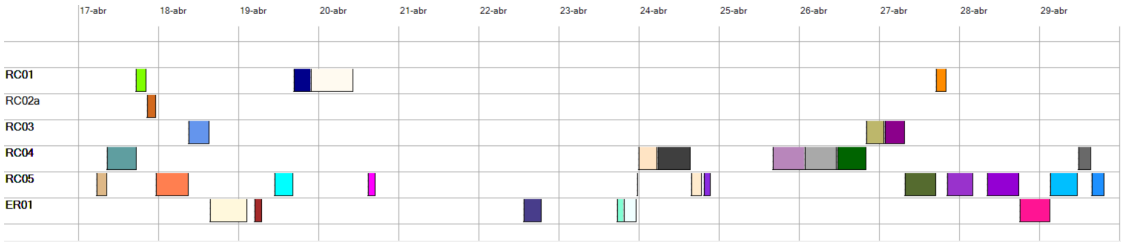
4. EXPERIMENTOS COMPUTACIONAIS

O Algoritmo DRC apresentado na Seção 3 foi implementado em uma aplicação desenvolvida na linguagem de programação C# sobre a plataforma *.NET Framework 4.6.1* [Microsoft, 2020], utilizando o ambiente de programação *Visual Studio 2017 Community*. Os testes foram realizados em um computador *Dell Inspiron 15 Série 7000* com processador *Intel Core i7-8550U*, com 16GB de mémoria RAM e sistema operacional *Windows 10* de 64 bits.

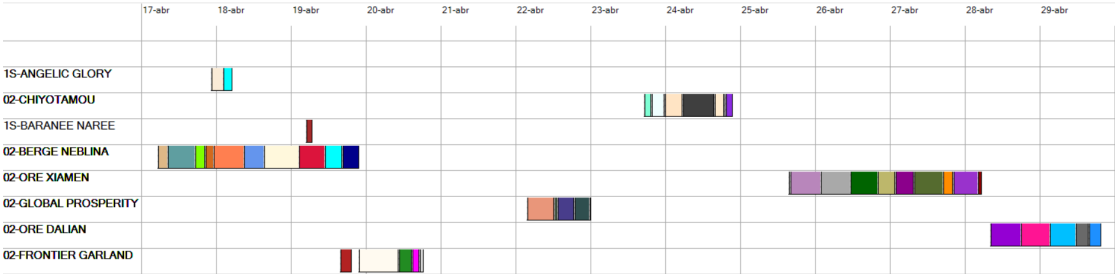
Para testar o algoritmo construtivo proposto utilizou-se uma instância real do Porto de Tubarão, da Vale S.A. Essa instância é composta por 11 pátios de estocagem, 8 recuperadoras de minério com velocidade de locomoção entre 15 e 25 m/min e taxa de recuperação entre 5000 e 7000 ton/h. Por sua vez, foram considerados 8 navios para embarque, com data de atracação variando entre 320 e 16337 min (5,3 e 272,3 h). A data de atracação considera uma janela de tempo de sequenciamento a partir de 17/04/2020 00:00. Foram consideradas 46 pilhas para embarque nos navios, sendo que as quantidades de minério nas pilhas variaram entre 2000 e 76700 ton. A quantidade de pilhas reservadas para cada navio variou entre 1 e 10 pilhas.

O resultado do sequenciamento das recuperadoras é ilustrado, na forma de gráficos de *Gantt*, nas imagens da Figura 3.

O gráfico de *Gantt* ilustrado na Figura 3(a) mostra a visão do sequenciamento das tarefas nas recuperadoras. A primeira coluna mostra as identificações das recuperadoras. No cabeçalho



(a) Visão das recuperadoras



(b) Visão dos navios

Figura 3: Sequenciamento das recuperadoras

das demais colunas estão os dias da janela de sequenciamento programada. O sequenciamento das pilhas em cada recuperadora também é exibido nessas colunas. O tempo de recuperação de cada pilha é representado por um retângulo com cores diferentes para cada pilha, enquanto que os tempos de locomoção são representados por retângulos hachurados na cor cinza.

O gráfico de *Gantt* ilustrado na Figura 3(b), por sua vez, exibe o resultado do mesmo sequenciamento, porém tendo como referência os navios a serem embarcados. A primeira coluna exibe as identificações de cada navio. As demais informações são as mesmas da figura anterior, destacando-se que são mantidas as mesmas cores utilizadas para identificar as pilhas.

O *makespan* obtido pelo algoritmo *DRC* foi de 18.460 minutos, isto é, 12 dias, 19 horas e 40 minutos. O tempo de execução foi de apenas 124 ms.

A Tabela 5 detalha o sequenciamento obtido. A coluna "Pilha" exibe as identificações das pilhas. A coluna "Tempo de Locomoção" mostra os tempos de locomoção das recuperadoras entre a pilha recuperada anteriormente e a pilha atual, em minutos. A coluna "Data de Disponibilidade" exibe as datas em que as pilhas estão disponíveis para recuperação, em minutos. A coluna "Data de Início da Recuperação", por sua vez, mostra as datas em que as pilhas começaram a ser recuperadas, em minutos. A coluna "Tempo de Recuperação" mostra os tempos de recuperação das pilhas, em minutos. A coluna "Data de Término da Pilha" exibe as datas de términos de recuperação das pilhas, em minutos. Nesta tabela, as linhas são ordenadas pela "Data de Término da Pilha".

Tabela 5: Resultado do Sequenciamento

Pilha a	**Tempo de Locomoção** $S_{ijk}^{a_1 a_2}$	**Data de Disponibilidade** r_j^h	**Data de Início da Recuperação** t_j^a	**Tempo de Recuperação** p_j^{ai}	**Data de Término da Pilha** C_j
H1115	5	320	325	188	513
F3746	16	320	529	522	1051
B1016	0	320	1037	198	1235
C3944	20	320	1255	136	1391
P2025	9	1349	1358	218	1576
J2129	10	1349	1586	158	1744
I3441	9	320	1400	574	1974
B1727	14	320	1994	372	2366
F2736	12	320	2378	657	3035
E0307	10	3166	3176	128	3304
I4249	9	320	3044	488	3532
H4956	6	320	3538	325	3863
J1020	4	3830	3834	214	4048
B3645	18	320	3894	291	4185
A0420	19	3830	4204	744	4948
I1623	10	3830	4958	248	5206
I0815	16	3830	5222	128	5350
I0001	7	3830	5357	85	5442
J3041	8	7420	7428	522	7950
P2025a	8	7420	7958	42	8000
E2937	11	7420	8012	325	8337
J2129a	4	7420	8341	274	8615
P0719	3	7420	8618	34	8652
E1927	4	9680	9684	128	9812
E1927a	0	9680	9812	232	10044
I2433	6	9680	10050	17	10067
G2430	5	9680	10074	342	10416
F0816	6	9680	10422	599	11021
I5056	9	9680	11030	177	11207
P0105	4	9680	11211	34	11245
H1620	14	9680	11259	118	11377
P0105a	0	12450	12450	25	12475
G3138	9	12450	12490	574	13064
F2736a	1	12450	13065	578	13643
G1623	4	12450	13647	522	14169
B1727a	0	12450	14170	330	14500
B3645a	12	12450	14512	352	14864
H4148	10	12450	14874	551	15425
B1016a	0	12450	15418	198	15616
H4956a	3	12450	15619	477	16096
I2433a	10	12450	16106	60	16166
I2433b	9	16337	16346	574	16920
F4756	11	16337	16931	548	17479
H3340	3	16337	17482	492	17974
G3947	9	16337	17986	235	18221
H2132	4	16337	18225	235	18460

Como pode ser observado na Tabela 5, a pilha H1115 está disponível para recuperação no instante 320 minutos. Como o tempo de locomoção da recuperadora RC05 (vide Figura 3(a)) é de 5 minutos a partir de sua posição inicial, a recuperação é iniciada no instante 325 minutos e concluído no instante 513 minutos. A última pilha a ser recuperada para embarque é a H2132, sendo sua recuperação encerrada no instante 18460 minutos.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho tratou o problema de sequenciamento de recuperadoras de minério. Para resolvê-lo, desenvolvemos um algoritmo heurístico construtivo, nomeado DRC, que avalia dentre uma lista de tarefas candidatas, aquelas que possuam o menor tempo de término, tendo por objetivo a minimização do tempo de conclusão da última pilha a ser recuperada.

Testamos o algoritmo DRC em uma instância real do Porto de Tubarão, em Vitória (ES). Com a aplicação do algoritmo foi possível encontrar uma solução aplicável e em menos de um segundo. Atualmente, o sequenciamento das recuperadoras é feito de forma manual, demandando um tempo significativo dos analistas de programação das operações do porto. A solução gerou como resultado um sequenciamento de recuperação de pilhas para embarque nos navios que respeita todas as restrições que foram caracterizadas neste trabalho.

Como trabalhos futuros pretende-se incluir outras características das operações do porto, como empilhamento nos pátios e embarque simultâneo de pilha para um mesmo navio, entre outras. Além disso, pretende-se também desenvolver um algoritmo metaheurístico para fornecer soluções de melhor qualidade.

Agradecimentos

Os autores agradecem à Universidade Federal de Ouro Preto, à Vale, ao Instituto Tecnológico Vale e às agências CNPq (processo 303266/2019-8), FAPEMIG (processo PPM-CEX 676/17) e CAPES (código de financiamento 001), pelo apoio ao desenvolvimento deste trabalho.

Referências

Amaral, M. d. e Pinto, R. P. (2010). Planejamento de operações de lavra em minas a céu aberto com alocação de equipamentos de carga e de transporte. *Anais do XLII SBPO*, p. 2123–2134.

Angelelli, E., Kalinowski, T., Kapoor, R., e Savelsbergh, M. W. (2016). A reclaimer scheduling problem arising in coal stockyard management. *Journal of Scheduling*, 19(5):563–582. ISSN 10946136.

Burdett, R. L., Corry, P., Yarlagadda, P. K., Eustace, C., e Smith, S. (2019). A flexible job shop scheduling approach with operators for coal export terminals. *Computers and Operations Research*, 104:15–36. ISSN 03050548.

Cereser, B. L. H. (2018). *O Problema de Alocação de Máquinas e Berços Portuários*. Tese de doutorado, Universidade Estadual de Campinas.

Chen, J. H., Lee, D.-H., e Goh, M. (2014). An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem. *European Journal of Operational Research*, 232(1):198–208.

Glover, F. (1996). Tabu search and adaptive memory programming - advances, applications and challenges. In Barr, R. S., Helgason, R. V., e Kennington, J. L., editors, *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, p. 1–75. Kluwer Academic Publishers.

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

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1):66–73.

Hu, D. e Yao, Z. (2012). Stacker-reclaimer scheduling in a dry bulk terminal. *International Journal of Computer Integrated Manufacturing*, 25(11):1047–1058. ISSN 0951192X.

Kirkpatrick, S., Gelatt, C. D., e Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680. ISSN 0036-8075. URL `https://science.sciencemag.org/content/220/4598/671`.

Lopes, A., Schulz, V., e Mauri, G. (2011). Grasp com path relinking para o problema de alocação de berços. *Pesquisa Operacional para o Desenvolvimento*, 3(3):218–229.

Microsoft (2020). Overview of the .net framework.

Pantuza, G., Souza, M. J. F., e Cabral, I. E. (2009). Uma nova formulação de programação matemática para o problema de planejamento de lavra. *Revista GEPROS*, (4):125–137.

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. ISSN 1366588X.

Resende, M. G. C. e Ribeiro, C. C. (2014). *GRASP: Greedy Randomized Adaptive Search Procedures*, p. 287–312. Springer US, Boston, MA. ISBN 978-1-4614-6940-7.

Sammarra, M., Cordeau, J. F., Laporte, G., e Monaco, M. F. (2007). A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling*, 10(4-5):327–336. ISSN 10946136.

Souza, M. J. F. (2012). Busca tabu. In Gaspar-Cunha, A., Antunes, C. H., e Takahashi, R., editors, *Manual de Computação Evolutiva e Metaheurísticas*, p. 177–200. Imprensa da Universidade de Coimbra, Portugal. ISBN 978-989-26-0150-2.

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