



**Programa de Pós-Graduação em Instrumentação, Controle e
Automação de Processos de Mineração (PROFICAM)
Escola de Minas, Universidade Federal de Ouro Preto (UFOP)
Associação Instituto Tecnológico Vale (ITV)**

Dissertação

**SEQUENCIAMENTO DE RECUPERADORAS DE MINÉRIO NO PORTO DE
TUBARÃO**

João Daniel Ferreira Jesus Junior

**Ouro Preto
Minas Gerais, Brasil
2021**

João Daniel Ferreira Jesus Junior

**SEQUENCIAMENTO DE RECUPERADORAS DE MINÉRIO NO PORTO DE
TUBARÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Luciano Perdigão Cota,
D.Sc.

Coorientador: Prof. Marcene Jamilson Freitas
Souza, D.Sc.

Ouro Preto
2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

J586s Jesus Junior, João Daniel Ferreira de.
Sequenciamento de recuperadoras de minério no Porto de Tubarão.
[manuscrito] / João Daniel Ferreira de Jesus Junior. - 2021.
45 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Luciano Perdigão Cota.
Coorientador: Prof. Dr. Marcone Jamilson Freitas Souza.
Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.
Área de Concentração: Engenharia de Controle e Automação de Processos Mineraiis.

1. Automação industrial. 2. Otimização combinatória. 3. Administração da produção - Sequenciamento de tarefas. 4. Portos. I. Cota, Luciano Perdigão. II. Souza, Marcone Jamilson Freitas. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5:622.2

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
PROGR. POS GRAD. PROF. INST. CONT. E AUT.
PROCESSOS DE MIN.



FOLHA DE APROVAÇÃO

João Daniel Ferreira de Jesus Junior

Sequenciamento de recuperadoras de minério no porto de Tubarão

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM), Convênio Universidade Federal de Ouro Preto/Associação Instituto Tecnológico Vale - UFOP/ITV, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle e Automação na área de concentração em Instrumentação, Controle e Automação de Processos de Mineração.

Aprovada em 02 de setembro de 2021

Membros da banca

Doutor - Luciano Perdigão Cota - Orientador - Instituto Tecnológico Vale
Doutor - Marcone Jamilson Freitas Souza - Universidade Federal de Ouro Preto
Doutor - Thiago Antonio Melo Euzébio - Instituto Tecnológico Vale
Doutor - Geraldo Regis Mauri - Universidade Federal do Espírito Santo

Luciano Perdigão Cota, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 29/11/2021



Documento assinado eletronicamente por **Bruno Nazário Coelho, COORDENADOR(A) DE CURSO DE PÓS-GRAD EM INSTRUMENTAÇÃO CONTROLE E AUTOMAÇÃO DE PROCESSOS DE MINERAÇÃO**, em 01/12/2021, às 11:10, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0252289** e o código CRC **DD744DE1**.

*Dedico este trabalho a minha
esposa Vânia, meus filhos João
Paulo e Luísa, meus pais João e
Geni e minhas irmãs Daniela e
Thabata.*

Agradecimentos

A Deus por me conceder essa maravilhosa oportunidade e por estar junto a mim em todos os momentos de minha vida.

À minha esposa Vânia e ao meu filho João Paulo por todo apoio incondicional, principalmente nos momentos mais difíceis.

Aos meus pais João e Geni por dar a mim os incentivos necessários para os estudos. Às minhas irmãs Daniela e Thabata pelo carinho fraterno.

Aos meus líderes Milton Oliveira e Bruno Mazzini pela confiança e apoio durante o ingresso no curso de mestrado.

Aos meus professores Luciano Cota e Marcone Jamilson que compartilharam comigo seus conhecimentos ao longo de toda a pesquisa.

A todos meus colegas do mestrado pela ajuda e companheirismo durante a jornada.

Aos meus colegas de trabalho por acreditarem em mim e por cobrir as minhas atividades durante os períodos de estudos.

À VALE pela oportunidade e suporte.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale SA.

*“Ninguém vence sozinho, nem no campo, nem na vida!”
(Papa Francisco).*

Resumo

Resumo da Dissertação apresentada ao Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SEQUENCIAMENTO DE RECUPERADORAS DE MINÉRIO NO PORTO DE TUBARÃO

João Daniel Ferreira Jesus Junior

Setembro/2021

Orientadores: Luciano Perdigão Cota

Marcone Jamilson Freitas Souza

Este trabalho trata de um problema de sequenciamento de recuperadoras de minério no Porto de Tubarão da Vale. Nesse problema, existe um conjunto de navios a serem embarcados com pilhas pré-definidas. Em cada pilha, existe um tipo de produto e uma quantidade de minério disponível. As pilhas são embarcadas em porções de navios, e cada porção deve armazenar um produto e quantidade de minério pré-definidos. As pilhas estão armazenadas em pátios de estocagem e são recuperadas por recuperadoras de minério e embarcados nos navios atracados. O objetivo do problema é sequenciar as operações das recuperadoras para minimizar a soma dos tempos de atracação dos navios. Para tratar o problema, foi desenvolvido um algoritmo baseado em GRASP. Este algoritmo aplica o *Basic Variable Neighborhood Descent* como o método de busca local. O algoritmo proposto foi testado em instâncias reais e os resultados mostraram que ele pode fornecer soluções de alta qualidade para auxiliar a tomada de decisão.

Palavras-chave: Otimização, Sequenciamento de Tarefas, Portos.

Macrotema: Logística; **Linha de Pesquisa:** Tecnologias da Informação, Comunicação e Automação Industrial; **Tema:** Aumento de Produtividade na Cadeia de Transporte; **Área Relacionada da Vale:** Portos.

Abstract

Abstract of Dissertation presented to the Graduate Program on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

IRON ORE RECLAIMER SCHEDULING AT TUBARAO PORT

João Daniel Ferreira Jesus Junior

September/2021

Advisors: Luciano Perdigão Cota

Marcone Jamilson Freitas Souza

This work deals with a scheduling problem of reclaimers in Vale's Tubarão Port. In this problem, there is a set of ships to be boarded with pre-defined stockpiles. In each stockpile, there is a type of product and an available amount of iron ore. Stockpiles are boarded in ship holds, and each hold must store a pre-defined product and amount of iron ore. Stockpiles are stored in storage yards, which have reclaimers to reclaim iron ore from stockpiles and send it to berthed ships. The objective of the problem is to schedule the operations of reclaimers to minimize the sum of ships' berthing time. For treating the problem, we developed a GRASP-based algorithm. This algorithm applies the Basic Variable Neighborhood Descent as the local search method. The proposed algorithm was tested in real instances, and the results showed that it can provide high-quality solutions to support decision-makers.

Keywords: Optimization, Job scheduling, Ports.

Macrotheme: Logistics; **Research Line:** Information Technology, Communication and Industrial Automation; **Theme:** Increased productivity in the transport chain; **Related Area of Vale:**Ports.

Lista de Figuras

Figura 1.1	Fluxo operacional do Porto de Tubarão. Fonte: Autor (2021).	13
Figura 2.1	Exemplo de carregamento em um porto. Fonte: Autor (2021).	19
Figura 2.2	Exemplo de uma possível solução. Fonte: Autor (2021).	21
Figura 4.1	Exemplo da representação de uma solução. Fonte: Autor (2021).	26
Figura 4.2	Exemplo de lista de grupos de passos do embarque um navio. Fonte: Autor (2021).	29
Figura 4.3	Interface do sistema de suporte à decisão. Fonte: Autor (2021).	34
Figura 5.1	Gráfico <i>box plot</i> dos resultados do algoritmo por instância. Fonte: Autor (2021).	37
Figura 5.2	Sequenciamento das recuperadoras. Fonte: Autor (2021)	39

Lista de Tabelas

Tabela 2.1	Características das pilhas. Fonte: Autor (2021).	19
Tabela 2.2	Características das recuperadoras. Fonte: Autor (2021).	20
Tabela 2.3	Elegibilidade das recuperadoras. Fonte: Autor (2021).	20
Tabela 2.4	Características dos navios a serem embarcados. Fonte: Autor (2021).	21
Tabela 5.1	Características das instâncias. Fonte: Autor (2021).	35
Tabela 5.2	Resultados algoritmo proposto. Fonte: Autor (2021).	37
Tabela 5.3	Tempo de atracação por navio na melhor solução encontrada por instância (horas). Fonte: Autor (2021).	38
Tabela 5.4	Total embarcado por navio na melhor solução encontrada por instância (kt). Fonte: Autor (2021).	38

Sumário

1	Introdução	12
1.1	Contextualização	12
1.2	Motivação	13
1.3	Objetivos	14
1.3.1	Objetivo Geral	14
1.3.2	Objetivos Específicos	14
1.4	Organização do Trabalho	15
2	Caracterização do Problema	16
3	Revisão Bibliográfica	22
3.1	Sequenciamento de Tarefas em Máquinas	22
3.2	Otimização Aplicada a Portos de Granéis Sólidos	23
4	Metodologia	26
4.1	Representação da Solução	26
4.2	Algoritmo GRASP	26
4.3	Fase Construtiva	28
4.4	Estrutura de Vizinhança	32
4.5	Busca Local	33
4.6	Implementação do Sistema de Suporte à Decisão	33
5	Experimentos Computacionais	35
5.1	Calibração dos Parâmetros	36
5.2	Resultado do Algoritmo	36
6	Conclusões e Trabalhos Futuros	40
6.1	Conclusões	40
6.2	Trabalhos Futuros	40
	Referências Bibliográficas	41

1. Introdução

1.1. Contextualização

O Brasil é conhecido internacionalmente por ser um grande exportador de minério de ferro. Em 2020, foram exportados 341 milhões de toneladas de minério de ferro e seus concentrados, totalizando 25,8 bilhões de dólares, correspondendo a 12,3% de todos os bens exportados (COMEXVIS, 2021). Para viabilizar a exportação de grandes quantidades de minério de ferro no Brasil, é utilizada a cadeia logística “Mina-Ferrovia-Porto-Navegação”. Nessa cadeia, o porto constitui um ponto importante de recebimento e exportação, com minério proveniente de minas localizadas a aproximadamente 900km. Em 2020, o minério de ferro foi responsável pela movimentação de 51,5% de todo granel sólido em portos, correspondendo à 356 milhões de toneladas (ANTAQ, 2021).

A demanda crescente de granéis sólidos, em especial de minério de ferro (ZHANG e SINGH, 2021), fez com que as pesquisas relacionadas à movimentação de granéis sólidos em portos se intensificasse (ANGELELLI *et al.*, 2016; BURDETT *et al.*, 2019; HU e YAO, 2012). Assim como esses, o presente trabalho também trata da movimentação de granéis, especificamente o minério de ferro, nas operações do Porto de Tubarão. Este porto está localizado na cidade de Vitória, Brasil. Inaugurado em 1966, esse porto é destinado à exportação de minério de ferro (VALE, 2015). O porto tem capacidade de exportação de aproximadamente 100 milhões de toneladas de minério de ferro por ano.

A Figura 1.1 mostra o fluxo operacional típico da movimentação de minério no porto. Na área de descarga, os trens de minério chegam pela estrada de ferro Vitória-Minas (EFVM) e são descarregados pelos viradores de vagões. Esses viradores de vagões são equipamentos capazes de virar uma dupla de vagões para que o minério seja descarregado em silos ao nível do chão. Em seguida, o minério é enviado por meio de correias transportadoras até os pátios de estocagem, onde são empilhados por empilhadeiras. Empilhadeiras são equipamentos portuários semimóveis, que se locomovem sobre trilhos nos pátios, recebem o minério da correia transportadora e formam as pilhas no pátio de estocagem. As pilhas ficam armazenadas no pátio com a destinação já planejada para um determinado navio. Quando esse navio atraca no porto, o seu embarque pode ser iniciado.

Para o embarque do minério estocado é utilizado outro equipamento semimóvel, conhecido como recuperadora. Este equipamento recupera o minério da pilha e o envia por meio de correias transportadoras até os píeres. Os píeres são construções avançadas sobre o mar onde os navios são atracados para serem embarcados. Em cada píer há carregadores de navios, que são equipamentos semimóveis, cuja função é embarcar o minério, provenientes dos pátios, nos porões dos navios atracados.

O porto possui capacidade de embarcar minério em até três navios ao mesmo tempo. Por sua vez, cada navio pode ser embarcado com minério proveniente de até duas pilhas simul-

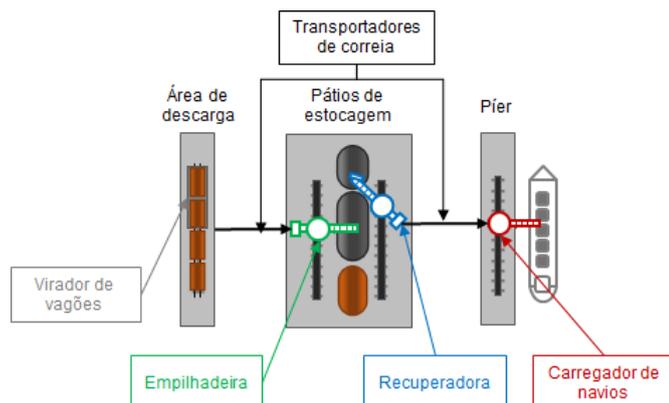


Figura 1.1: Fluxo operacional do Porto de Tubarão. Fonte: Autor (2021).

taneamente. A escolha de qual recuperadora será alocada na recuperação de uma determinada pilha é um problema complexo que impacta a eficiência do porto, visto que as pilhas embarcadas simultaneamente devem ser do mesmo produto e duas recuperadoras podem recuperar a mesma pilha simultaneamente. O sequenciamento das operações das recuperadoras é realizado para um horizonte de curto prazo (entre 1 e 10 dias) objetivando que o navio fique atracado o menor tempo possível. Para resolver o problema de sequenciamento tem-se como dados de entrada: as pilhas, os produtos das pilhas, a destinação de cada uma delas, as recuperadoras elegíveis para recuperar as pilhas e o posicionamento das empilhadeiras e recuperadoras. Outros aspectos também são considerados como interferências entre outros processos do porto como empilhamentos e transferência de materiais entre pátios.

A programação das operações de recuperação de pilhas pode ser modelada como um problema de sequenciamento de tarefas, em que as tarefas são essas operações de recuperação e as máquinas são as recuperadoras. O problema envolvido no sequenciamento dessas operações portuárias é de natureza combinatória, pertencendo à classe NP-difícil (HU e YAO, 2012). Em vista disso, é proposto tratá-lo por meio de um algoritmo baseado na meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) (GLOVER, 1996). Algoritmos baseados em GRASP têm sido aplicados com sucesso na resolução de vários problemas de otimização combinatória, como em Anokić *et al.* (2019), Guimarães *et al.* (2019) e Abu-Marrul *et al.* (2021).

1.2. Motivação

Portos de minério são um elo importante na da cadeia de exportação. Operações portuárias otimizadas garantem eficiência que se traduz em menos custos e mais navios embarcados.

A alocação de recuperadoras no Porto de Tubarão é um grande desafio e tem impacto relevante na eficiência desse porto. O tempo de estadia dos navios no porto pode ser reduzido, por exemplo, com um melhor sequenciamento das operações. O sequenciamento de recuperação no porto é um problema para o qual houve diversas tentativas de resolução baseadas em planilhas

eletrônicas. Porém, atualmente a equipe de programação de operação do porto ainda realiza algumas atividades de sequenciamento de operações de forma manual.

De nosso conhecimento, as operações de recuperação de minério no Porto de Tubarão têm características que ainda não foram abordadas na literatura, como por exemplo, (i) o envio de minério de duas pilhas com o mesmo tipo de produto simultaneamente para um mesmo navio, (ii) duas recuperadoras operando simultaneamente na mesma pilha e (iii) a minimização do tempo de estadia dos navios.

Do ponto de vista teórico, o estudo deste problema é de grande interesse, visto a dificuldade de resolver instâncias reais em tempo de tomada de decisão reduzido (GAREY e JOHNSON, 1979). Assim, existe um campo de pesquisa em aberto para o estudo de algoritmos que o resolvam de forma eficiente.

1.3. Objetivos

Nesta seção são apresentados os objetivos deste trabalho.

1.3.1. Objetivo Geral

O objetivo deste trabalho é desenvolver um sistema de suporte a decisão (SSD) que pode ser utilizado como ferramenta de apoio aos programadores das operações portuárias para sequenciar as operações de recuperação de minério no Porto de Tubarão.

1.3.2. Objetivos Específicos

Os objetivos específicos listados abaixo permitem o alcance do objetivo geral:

- Estudar trabalhos da literatura que tratam o sequenciamento de operações portuárias em portos de minério, carvão e outros granéis sólidos;
- Analisar e estudar técnicas de solução de problemas de otimização;
- Estudar problemas de sequenciamento;
- Caracterizar o problema baseado no porto em estudo;
- Tratar o problema buscando minimizar o tempo de espera e o tempo de operação dos navios;
- Considerar um horizonte de planejamento de curto prazo;
- Desenvolver um algoritmo construtivo para obter soluções de boa qualidade para o problema;

- Desenvolver estruturas de vizinhanças para explorar o espaço de soluções do problema;
- Desenvolver um SSD baseado na meta-heurística GRASP;
- Validar o SSD usando dados reais do Porto de Tubarão.

1.4. Organização do Trabalho

Os capítulos restantes estão organizados como segue:

- **Capítulo 2 - Caracterização do Problema:** Neste capítulo, o problema objeto de estudo é caracterizado;
- **Capítulo 3 - Revisão Bibliográfica:** Neste capítulo é apresentada uma breve revisão da literatura sobre problemas de sequenciamento e alguns trabalhos de otimização aplicados a portos de granéis sólidos;
- **Capítulo 4 - Metodologia:** Neste capítulo são descritos os algoritmos que serão utilizados para resolver o problema apresentado;
- **Capítulo 5 - Experimentos Computacionais:** Neste capítulo são apresentados os experimentos e os resultados obtidos;
- **Capítulo 6 - Conclusões e Trabalhos Futuros:** Este capítulo finaliza a dissertação apresentando as conclusões e a perspectiva de trabalhos futuros.

2. Caracterização do Problema

O problema de sequenciamento de operações portuárias aqui tratado envolve um conjunto Y de pátios de estocagem onde as pilhas são estocadas, $Y = \{1, \dots, |Y|\}$, um conjunto P de pilhas de minério a serem embarcadas em navios, $P = \{1, \dots, |P|\}$, um conjunto SH de navios a serem embarcados, $SH = \{1, \dots, |SH|\}$, um conjunto RC de máquinas recuperadoras, $RC = \{1, \dots, |RC|\}$ e um conjunto PR de produtos, da natureza do minério ou pelota, que podem ser embarcados, $PR = \{1, \dots, |PR|\}$. Este problema tem as seguintes características, as quais estão agrupadas pelos recursos envolvidos.

(1) Pátios

- (a) Cada pátio $l \in Y$ é dividido em regiões definidas como balizas B_l , $B_l = \{0, \dots, |B_l|\}$;

(2) Pilhas

- (a) A localização de cada pilha $a \in P$ do pátio $l \in Y$ é representada no intervalo $[a_{ini}^l, a_{end}^l]$, onde a_{ini}^l e a_{end}^l são, respectivamente, as balizas inicial e final da pilha a do pátio l , em metros, sendo $1 \leq ini < end \leq |B_l|$;
- (b) A localização média de uma pilha $a \in P$ no pátio $l \in Y$ é dada por $a_{med}^l = (a_{ini}^l + a_{end}^l)/2$;
- (c) Cada pilha $a \in P$ possui uma quantidade $sizePile_a$, em toneladas, de um único produto $q \in PR$;
- (d) De cada pilha $a \in P$ pode ser retirada uma quantidade $sizePile_a^{real}$, sendo $sizePile_a^{real} \leq sizePile_a$;
- (e) Ao começar a ser recuperada, uma pilha $a \in P$ pode ter sua recuperação interrompida e posteriormente retomada pela mesma ou por uma recuperadora diferente;

(3) Navios

- (a) Cada navio $h \in SH$ possui um conjunto de produtos $PR_h \subseteq PR$ que podem ser embarcados;
- (b) Cada navio $h \in SH$ possui um conjunto de porções BS_h a serem embarcados, $BS_h = \{1, \dots, \theta, \dots, |BS_h|\}$;
- (c) O embarque de cada navio $h \in SH$ é dividido em um conjunto SP_h de passos de embarque, sendo $SP_h = \{1, \dots, \phi, \dots, |SP_h|\}$. O conjunto de passos de embarque e a respectiva quantidade a ser embarcada em cada passo são definidos previamente pelo comandante do navio e visam a manter a estabilidade do navio durante o embarque;
- (d) Em cada passo $\phi \in SP_h$ é embarcado um único produto em um único porção $\theta \in BS_h$;
- (e) Para cada passo $\phi \in SP_h$ está programada uma quantidade $sizeStep_\phi$ que deve ser atendida, em toneladas;

- (f) Para cada passo $\phi \in SP_h$ pode haver o embarque simultâneo de sim_h pilhas, desde que as pilhas sejam do mesmo produto. O número máximo de pilhas que podem ser embarcadas simultaneamente é max_h .
 - (g) Para cada porção $\theta \in BS_h$ está programada uma quantidade $sizeBas_\theta$ de minério que deve ser atendida, em toneladas;
 - (h) Para cada navio $h \in SH$ está programada uma quantidade $sizeShip_h$ de minério que deve ser atendida, em toneladas, sendo $sizeShip_h = \sum_{\phi=1}^{|SP_h|} sizeStep_\phi$;
 - (i) 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 no navio $h \in SH$, previamente fixado, e após a sua data de atracação.
 - (j) Em cada navio $h \in SH$ pode ser embarcado mais de um produto diferente, sendo que cada porção $\theta \in BS_h$ só pode receber um único produto;
 - (k) Cada navio $h \in SH$ possui uma data de atracação $date_h^i$ e uma data prevista de desatracação $date_h^e$;
 - (l) A cada navio $h \in SH$ está associado um peso $\omega_h \in (0, 1]$, que define a prioridade de embarque desse navio.
- (4) Recuperadoras
- (a) Cada recuperadora $i \in RC$ possui uma velocidade de locomoção $speed_i^l$, em m/min;
 - (b) Cada recuperadora $i \in RC$ possui uma velocidade de giro de lança $speed_i^s$, em g/min;
 - (c) Cada recuperadora $i \in RC$ possui uma taxa de recuperação $rate_i$, em t/h;
 - (d) As recuperadoras do conjunto RC podem recuperar as pilhas do conjunto P e enviar para os navios do conjunto SH ;
 - (e) Uma recuperadora $i \in RC$ é elegível para recuperar uma pilha $a \in P$ do pátio $l \in Y$, e escreve-se, $e_{il} = 1$, se essa recuperadora i 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;
- (5) Tarefas de Recuperação de Pilhas para Embarque em Navios
- (a) Há um conjunto N de tarefas de recuperação de pilhas, $N = \{1, \dots, |N|\}$. Cada tarefa $j \in N$ representa o embarque de uma quantidade $sizeJob_j$ de minério recuperado de uma pilha $a \in P$ por uma recuperadora $i \in RC$;
 - (b) Cada tarefa $j \in N$ possui uma data de disponibilidade r_j^h , com $h \in SH, j \in N$, a partir da qual ela pode ser iniciada. A data de disponibilidade r_j^h da tarefa j equivale à data de atracação do navio h para a qual a pilha $a \in P$, associada à tarefa j , está prefixada;

- (c) Cada tarefa $j \in N$ possui um tempo de processamento p_j^i para ser executada, com $i \in RC$. Tendo em vista que cada tarefa j representa a recuperação de uma quantidade $sizeJob_j$ de uma pilha $a \in P$ por uma recuperadora i para um navio $h \in SH$ predefinido. O valor p_j^i é calculado pela relação $p_j^i = sizeJob_j / rate_i$;
- (d) 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 $S_{ijk}^{a_1 a_2} = d_{a_1 a_2} / speed_i^t$, sendo $d_{a_1 a_2}$ a distância entre as localizações médias das pilhas a_1 e a_2 , em metros;
- (e) O tempo de locomoção da recuperadora $i \in RC$ para se posicionar na primeira tarefa k associada à pilha a_1 a ser executada é representado por $S_{i0k}^{a_0 a_1}$. O índice 0 indica uma tarefa fictícia de recuperação de uma pilha a_0 cuja localização é $[0, 0]$ e $sizePile_{a_0} = 0$;

O objetivo do problema é encontrar um sequenciamento das $|P|$ pilhas nas $|RC|$ recuperadoras de forma a minimizar a soma ponderada dos tempos de estadia dos navios no porto. O fator de ponderação consiste em priorizar o atendimento dos navios que requerem uma quantidade maior de minério. A estadia de um navio no porto é calculada como a diferença do tempo de término de embarque e sua data de atracação. A função objetivo está formulada pela Equação (2.1), sendo C_h e r_h o tempo de conclusão do embarque e o tempo de atracação do navio $h \in SH$, respectivamente.

$$\min \sum_{h=1}^{|SH|} \omega_h (C_h - r_h) \quad (2.1)$$

Os navios com maior prioridade na função objetivo são aqueles que demandam a maior quantidade de minério a ser embarcado. Essa prioridade ω_h é calculada conforme a Equação (2.2), na qual $sizeShip_t$ é a quantidade de material a ser embarcada no navio $t \in SH$.

$$\omega_h = \frac{sizeShip_h}{\sum_{t=1}^{|SH|} sizeShip_t} \quad \forall h \in SH \quad (2.2)$$

Para melhor compreensão do problema, considere um porto de dimensões reduzidas mostrado na Figura 2.1. 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 contidas na Tabela 2.1. Na coluna “Pilha” está a identificação da pilha. A coluna “Pátio” informa o pátio onde a pilha está estocada. A coluna “Localização” exhibe a localização da pilha, composta pelas posições das balizas inicial e final, respectivamente, em metros. A coluna “Localização média” mostra a localização média da pilha, em metros. A coluna “Quantidade”, por sua vez, exhibe a quantidade de material da pilha, em toneladas.

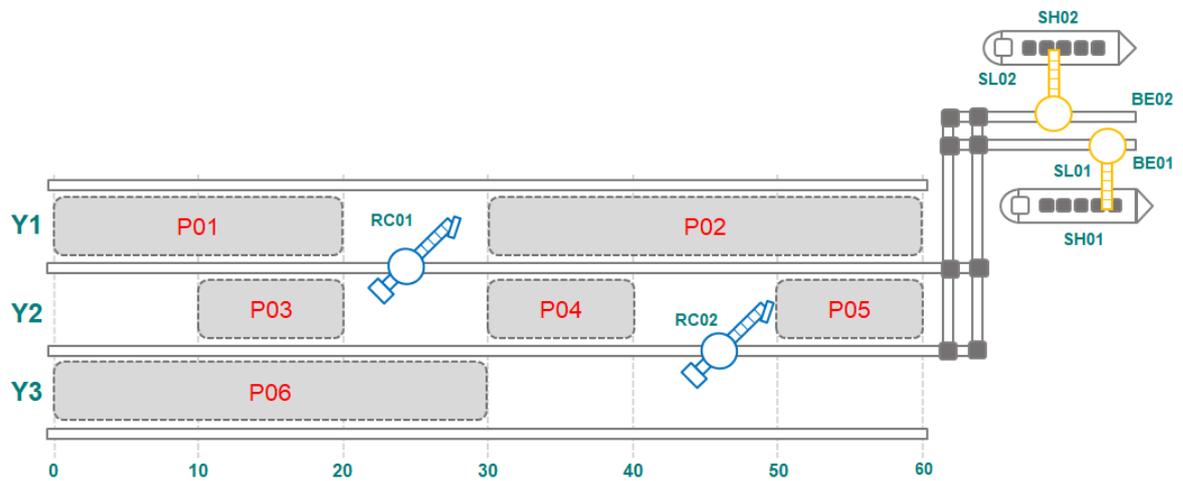


Figura 2.1: Exemplo de carregamento em um porto. Fonte: Autor (2021).

Tabela 2.1: Características das pilhas. Fonte: Autor (2021).

Pilha a	Pátio l	Localização $[a_{ini}^l, a_{end}^l]$	Localização média	Quantidade
			(m) a_{med}^l	(t) $size_a$
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

A Tabela 2.2 mostra as características das recuperadoras. A coluna “Recuperadora” exibe a identificação da recuperadora. Nas colunas “Velocidade” e “Taxa de recuperação” são exibidas a velocidade de locomoção e taxa de recuperação da recuperadora, em metros/minuto e toneladas/hora, respectivamente.

Tabela 2.2: Características das recuperadoras. Fonte: Autor (2021).

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

A Tabela 2.3 exibe a elegibilidade de cada recuperadora para recuperar pilhas em cada pátio. A coluna “Recuperadora” e “Pátio” mostram a identificação da recuperadora e do pátio, 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.

Tabela 2.3: Elegibilidade das recuperadoras. Fonte: Autor (2021).

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

A Tabela 2.4 exibe as características dos navios a serem embarcados. As colunas “Navio”, “Berço de Atracação” e “Data de Atracação” exibem a identificação, o berço de atracação e a data de atracação do navio, respectivamente. A coluna “Pilhas Reservadas” exibe as pilhas reservadas para o navio. Por fim, a coluna “Passos e Porões” exibem os passos de embarque dos porões com as respectivas quantidades a serem embarcadas definidas previamente pelo comandante do navio. Por exemplo, o navio SH01 tem como berço de atracação o berço BE01, data de atracação no instante 0 min. Para este navio estão reservadas as pilhas P01, P06 e P04. O navio possui os porões 1, 2 e 3 a serem embarcados, compostos por somente um passo cada, respectivamente os passos 1, 2 e 3. Para cada porão estão programadas 400 toneladas totalizando para o navio, 1200 toneladas.

Na Figura 2.2 é ilustrada uma solução do problema, que consiste em um sequenciamento das seis pilhas nas duas recuperadoras.

Tabela 2.4: Características dos navios a serem embarcados. Fonte: Autor (2021).

Navio	Berço de Atracação	Data de Atracação (min)	Pilhas Reservadas	Passos e Porões		
				Passo	Porão	Quantidade Programada (t)
SH01	BE01	0	P01, P06 e P04	1	1	400
				2	2	400
				3	3	400
SH02	BE02	5	P03, P02 e P05	1	1	500
				2	2	500

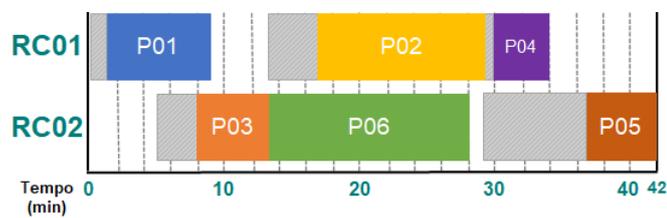


Figura 2.2: Exemplo de uma possível solução. Fonte: Autor (2021).

Como pode ser observado na Figura 2.2, na recuperadora RC01 foram alocadas as pilhas P01, P02 e P04, respectivamente nessa ordem. As pilhas P03, P06 e P05 foram alocadas nessa ordem na recuperadora RC02. As partes hachuradas representam os tempos de deslocamento das recuperadoras entre as tarefas.

3. Revisão Bibliográfica

Neste capítulo são apresentados trabalhos relacionados ao Sequenciamento de Tarefas na Seção 3.1, enquanto que na Seção 3.2 é realizada uma revisão de trabalhos em sequenciamento de operações de portos de granéis sólidos.

3.1. Sequenciamento de Tarefas em Máquinas

O problema de sequenciamento de tarefas é amplamente abordado na literatura e aplicado em vários segmentos de serviços e indústrias. Em empresas de serviços, por exemplo, há um problema que faz parte da rotina de vários aeroportos no mundo: o sequenciamento de aeronaves nos portões de embarque (PINEDO, 2012). Considere que há vários portões de embarque e centenas de aviões aterrissando e decolando diariamente em um aeroporto. Os portões desse aeroporto estão sujeitos a várias restrições, tais como capacidade de atendimento de pequenas ou grandes aeronaves (*widebodies*), capacidade de atendimento de pequenas ou grandes quantidades de passageiros e ocupação, tendo em vista que um portão acomoda somente uma aeronave por vez. As aeronaves devem ser direcionadas aos portões que possam acomodá-las minimizando tempos de embarque e desembarque, atrasos de decolagem e congestionamentos de aeronaves a espera de permissões de pouso.

Faeda (2015) trata um problema de sequenciamento de tarefas em máquinas paralelas não relacionadas considerando restrições de precedência entre as tarefas e tempos de preparação dependentes da sequência. O objetivo do problema é minimizar o *makespan* e para resolvê-lo, são propostos três modelos de programação linear inteira mista (PLIM) e sete heurísticas construtivas. O autor também propõe a utilização do método *Proximity Search* (PS) (FISCHETTI e MONACI, 2014) que procura determinar soluções ótimas para o problema a partir de uma solução inicial e um modelo PLIM. Algoritmos baseados em meta-heurísticas também foram desenvolvidos.

Em Tran *et al.* (2016), os autores estudam um problema de sequenciamento de máquinas paralelas não relacionadas com tempo de preparação dependente da sequência com o objetivo de minimizar o *makespan*. Dois algoritmos para a solução do problema são propostos: O primeiro é baseado na decomposição de Benders e o segundo é um algoritmo *Branch&Check*. A operação desses algoritmos é dividida em duas fases. Na primeira fase, uma formulação PLIM é usada para alocar as tarefas às máquinas. Na segunda fase, um otimizador para o subproblema do caixeiro viajante é utilizado para encontrar soluções ótimas em cada máquina. Esses algoritmos são capazes de provar que as soluções encontradas são ótimas.

Afzalirad e Rezaeian (2016), por sua vez, estudam um problema de sequenciamento de máquinas paralelas não relacionadas com restrição de recursos, tempos de preparação dependentes da sequência, datas de disponibilidade das tarefas, elegibilidade de máquinas e restrições de precedência com objetivo de minimizar o *makespan*. Para resolver o problema, é proposto

um PLIM. Para solução de instâncias de grande porte do problema, é proposto um algoritmo genético (AG) (HOLLAND, 1992) e um sistema imunológico artificial (SIA) (KUMAR *et al.*, 2009). Para calibrar os parâmetros desses dois algoritmos, é utilizado o método de Taguchi (TAGUCHI, 1986).

Em Cota *et al.* (2018) e Cota *et al.* (2019), os autores se dedicam a um problema de otimização multiobjetivo de *green scheduling* para minimização do *makespan* e o consumo total de energia consumida pelas máquinas. O problema trata do sequenciamento de máquinas paralelas não relacionadas com tempo de preparação dependentes da sequência. Em Cota *et al.* (2018), os autores definem o problema e propõem um modelo PLIM para resolver instâncias de pequeno porte. Já em Cota *et al.* (2019), é proposto um algoritmo heurístico multiobjetivo, chamado MO-ALNS/D, que combina os algoritmos clássicos *Adaptive Large Neighborhood Search* (ALNS) e MOEA/D para resolver instâncias de grande porte do problema.

Fanjul-Peyro *et al.* (2019) estudam um problema de sequenciamento de máquinas paralelas com tempos de preparação dependentes da sequência. O objetivo do problema é a minimização do *makespan*. Para resolver o problema, os autores formulam um novo modelo PLIM a partir do modelo proposto em Avalos-Rosales *et al.* (2015), em que algumas inequações que caracterizam as restrições são redefinidas. Além do novo modelo PLIM, é proposto um algoritmo baseado em programação matemática. Esse algoritmo é inspirado no método *Branch&Check* de Tran *et al.* (2016) com nova metodologia e combinação com o PLIM proposto pelos autores.

3.2. Otimização Aplicada a Portos de Granéis Sólidos

Hu e Yao (2012) estudam um problema de sequenciamento de movimentação de minério em portos com o objetivo de minimizar o *makespan*. Os autores definem uma tarefa como sendo o empilhamento ou a recuperação completa de uma pilha em um pátio. As tarefas são realizadas por empilhadeiras-recuperadoras (ER). Estes equipamentos são capazes de empilhar e recuperar em momentos distintos. No problema considerado, há um tempo de preparação da tarefa para que a máquina se posicione na pilha relativa à próxima tarefa depois de recuperar (ou empilhar) a última pilha. Os autores assumem que cada ER e tarefa estão disponíveis no início do sequenciamento, que cada ER já está posicionada na pilha a ser empilhada (ou recuperada) e que, uma vez começada, a tarefa não pode ser interrompida enquanto a pilha não for totalmente recuperada (ou empilhada). Para resolver o problema, os autores propõem um modelo de programação matemática. Porém, pelo fato de o problema ser NP-difícil (GAREY e JOHNSON, 1979), eles também desenvolveram algoritmos genéticos (AG) (HOLLAND, 1992) para tratar instâncias maiores do problema. Diferentemente deste trabalho, os autores não consideram a recuperação parcial das pilhas.

Boland *et al.* (2012) abordam um problema de planejamento de pátio de estocagem de carvão mineral aplicado ao Porto de Newcastle, Austrália. O objetivo é minimizar o tempo

médio de atraso de partida dos navios. Os autores caracterizam as operações do porto e propõem algoritmos construtivos e os aplicam a instâncias reais do problema.

Sun e Tang (2013) estudam um problema de sequenciamento de equipamentos de transporte de insumos siderúrgicos tais como minério e carvão em uma indústria siderúrgica. O problema tratado consiste em sequenciar recuperadoras que retiram os insumos dos pátios e os enviam às unidades fabris da siderúrgica por meio de correias transportadoras. A operação de recuperação e transporte do material do pátio até a unidade é considerada uma tarefa. Os autores assumem que as recuperadoras e as unidades fabris são a origem e o destino, respectivamente, das tarefas. As tarefas de transporte entre os pátios e as unidades possuem datas de disponibilidade, que são os instantes nos quais as unidades estão aptas a receberem os insumos. Os autores também consideram o tempo de preparação entre tarefas, que são os tempos demandados pelas recuperadoras para se locomover entre pilhas recuperadas consecutivamente. Para modelar o problema, eles propõem um modelo de programação matemática com o objetivo de minimizar o custo total dos atrasos. Para resolvê-lo, eles aplicam o método de decomposição de Benders (BENDERS, 2005).

Em Angelelli *et al.* (2016), os autores investigam algumas variações de um problema de sequenciamento que podem surgir na operação de recuperadoras em um porto de exportação de carvão. Como características do problema, eles consideram que duas recuperadoras podem ser movimentar sobre os trilhos de uma mesma berma entre dois pátios. No início do sequenciamento, todas as pilhas a serem recuperadas já estão formadas nos pátios, as recuperadoras estão posicionadas nos extremos opostos dos pátios e podem recuperar pilhas simultaneamente em qualquer pátio. Porém, elas não podem passar uma sobre a outra, o que caracteriza uma colisão. As recuperadoras voltam para suas respectivas posições de origem ao final da recuperação de todas as pilhas. Algumas propriedades do problema são analisadas e, para algumas variantes, os autores fazem provas de complexidade dos algoritmos propostos. Para uma das variantes, em particular, eles apresentam um algoritmo de aproximação com fator igual a $4/3$ do valor da solução ótima. De forma semelhante, Kalinowski *et al.* (2017) estudam o mesmo problema. Esses autores, porém, não assumem que todas as pilhas estão disponíveis para recuperação no início do sequenciamento. Para resolver o problema, eles apresentam um modelo PLIM. Além disso, eles mostram que, dada uma sequência de recuperação, o problema pode ser resolvido com um algoritmo de tempo pseudo-polinomial.

Burdett *et al.* (2019, 2020) tratam um problema de sequenciamento de operações em um porto de exportação de carvão. O porto é modelado com as seguintes características: chegada de trens a serem descarregados, chegada de navios a serem embarcados, controle anticollisão (*collision avoidance*) de máquinas de pátio, permissão de preempção de tarefas e produtos diferentes podem ser embarcados simultaneamente em um navio. Os autores o modelam como um problema de sequenciamento *job shop* flexível. Dado que o problema é NP-difícil, eles desenvolvem algoritmos heurísticos baseados em *Simulated Annealing* (KIRKPATRICK *et al.*, 1983) e Algoritmos Genéticos. Diferentemente dos autores, neste trabalho há controle do tipo

de minério a ser embarcado simultaneamente em um navio, uma vez que no Porto de Tubarão é permitido embarcar apenas um tipo de produto por vez.

Em Ünsal (2020) são estudadas variantes de um problema de sequenciamento de recuperadoras de granéis sólidos, com e sem interferência de operações de empilhamento nas operações de sequenciamento das recuperadoras. O autor considera como características do problema: tempo de preparação dependente da sequência, restrições de elegibilidade e precedência. Para resolver o problema, é proposta uma formulação PLIM e algumas variantes.

Os trabalhos descritos nesta seção tratam de problemas de sequenciamento de operações portuárias de granéis sólidos. Autores como Hu e Yao (2012); Ünsal (2020) propõem primeiramente uma formulação PLIM para solução de instâncias pequenas e métodos heurísticos para instâncias reais e maiores dos problemas. Trabalhos, como em Burdett *et al.* (2019, 2020), são de natureza NP-difícil; desta forma, os autores propõem somente métodos heurísticos e meta-heurísticos para tratá-los. Porém, nenhum dos autores propõem métodos baseados na meta-heurística GRASP. Trabalhos como em Boland *et al.* (2012); Hu e Yao (2012) possuem características semelhantes a este trabalho, por exemplo, tempo de preparação entre tarefas; entretanto, com menos características operacionais. Por sua vez, os trabalhos Burdett *et al.* (2019, 2020); Ünsal (2020) incluem outras características operacionais como tarefas de empilhamentos e controle anticolisão não tratadas neste trabalho.

4. Metodologia

A Seção 4.1 mostra como uma solução é representada e avaliada. A Seção 4.2 detalha o algoritmo proposto baseado na metaheurística GRASP. A Seção 4.3 apresenta o algoritmo da fase de construção do GRASP. A estrutura de vizinhança utilizada para explorar o espaço de busca é descrita na Seção 4.4. A Seção 4.5 descreve o procedimento de busca local utilizado. Por fim, a Seção 4.6 apresenta o sistema de suporte à decisão.

4.1. Representação da Solução

Neste problema, uma solução s é representada por uma lista de recuperadoras com $|RC|$ posições, onde cada posição representa uma recuperadora e a cada recuperadora está associada uma lista de tarefas por ela realizada.

A Figura 4.1 ilustra a representação de uma possível solução para o problema do porto de dimensões reduzidas caracterizado no Capítulo 2. Nesta figura, as tarefas de recuperação estão representadas por retângulos coloridos proporcionais à quantidade de minério a ser recuperada. As linhas que ligam duas tarefas consecutivas representam o tempo de preparação cujo tamanho é proporcional ao tempo de locomoção da recuperadora entre duas pilhas. As tarefas de 400t da pilha P01, 600t da pilha P02 e 200t da pilha P04 estão alocadas à recuperadora RC01 nesta ordem. Para a recuperadora RC02 estão alocadas nesta ordem, as tarefas de recuperação de 200t da pilha P03, 600t da pilha P06 e, por fim, 200t da pilha P05.

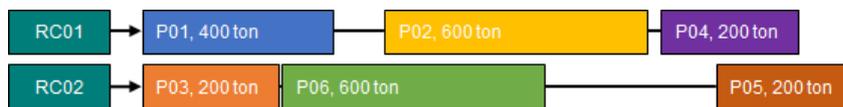


Figura 4.1: Exemplo da representação de uma solução. Fonte: Autor (2021).

Uma solução s é avaliada pela função de avaliação $f(s)$ dada pela Equação 4.1, a ser minimizada.

$$f(s) = \sum_{h=1}^{|SH|} \omega_h (C_h(s) - r_h) \quad (4.1)$$

Nesta equação, $C_h(s)$ representa o tempo de conclusão do embarque do navio h na solução s , r_h é o instante de atracação do navio e ω_h é o fator de prioridade do navio. Para a solução ilustrada na Figura 4.1, o valor da função de avaliação é 35,4.

4.2. Algoritmo GRASP

O algoritmo proposto para resolver este problema é baseado na meta-heurística GRASP. O algoritmo contém duas fases que são aplicadas em sequência até que o critério de parada

seja satisfeito. A primeira fase consiste em um algoritmo construtivo parcialmente guloso, que combina estratégias gulosas e aleatórias. A segunda é a fase de busca local, na qual a solução construída na fase anterior é refinada. A melhor das soluções refinadas é retornada pelo algoritmo. Seu pseudocódigo está descrito pelo Algoritmo 1.

Algoritmo 1: Pseudocódigo do algoritmo GRASP

Entrada: Parâmetro α , Função gulosa $g(\cdot)$, Função de avaliação $f(\cdot)$, Número $|P|$ de pilhas, Número médio $|RC|^{med}$ de recuperadoras elegíveis por pilha, Parâmetro k

Saída: Solução GRASP s

```

1  $s^* \leftarrow Construtivo(0, g(\cdot))$  ;
2  $f^* \leftarrow f(s^*)$  ;
3  $iterMax \leftarrow k \times |P| \times |RC|^{med}$  ;
4 enquanto ( $i \leq iterMax$ ) faça
5   |  $s \leftarrow Construtivo(\alpha, g(\cdot))$  ;
6   |  $s \leftarrow BVND(s, f(\cdot))$  ;
7   | se ( $f(s) < f^*$ ) então
8     |   |  $s^* \leftarrow s$  ;
9     |   |  $f^* \leftarrow f(s)$  ;
10    |   |  $i \leftarrow 0$  ;
11    |   fim
12    | senão
13    |   |  $i \leftarrow i + 1$  ;
14    |   fim
15  fim
16  $s \leftarrow s^*$  ;
17 retorna  $s$  ;
```

O Algoritmo 1 possui como parâmetros de entrada α , a função adaptativa gulosa $g(\cdot)$, a função de avaliação $f(\cdot)$, Número $|P|$ de pilhas, Número médio $|RC|^{med}$ de recuperadoras elegíveis por pilha e o parâmetro k . O algoritmo inicia com uma solução construída de forma totalmente gulosa pelo procedimento $Construtivo(\cdot)$ na linha 1. Em seguida, na linha 2, o valor da função de avaliação da melhor solução, f^* , é iniciado com o valor da função de avaliação retornado pelo procedimento da linha anterior. O parâmetro $iterMax$, que representa o número máximo de iterações sem melhora do algoritmo, é definido na linha 3. As linhas 4 a 15 definem o laço iterativo do algoritmo. A cada iteração, uma solução s é construída por um algoritmo construtivo parcialmente guloso usando o procedimento $Construtivo(\cdot)$ na linha 5. Então, na linha 6, a solução s é refinada pelo método de busca local BVND descrito na Seção 4.5. Caso a solução refinada seja melhor do que a solução encontrada até então (linha 7), a melhor solução s^* e o melhor valor da função f^* são atualizados nas linhas 8 e 9, e o número i de iterações sem melhora é reiniciado com o valor 0 na linha 10. Se não houver melhora, o número de iterações sem melhora é incrementado na linha 13. O laço de repetições é executado enquanto o número máximo de iterações sem melhora não é atingido. Ao fim do algoritmo, a melhor solução encontrada durante a busca é retornada na linha 17.

4.3. Fase Construtiva

O algoritmo construtivo parcialmente guloso utilizado na fase de construção do GRASP tem seu pseudocódigo descrito pelo Algoritmo 2.

Algoritmo 2: Algoritmo da fase de construção GRASP

Entrada: Parâmetro α , Função gulosa $g(\cdot)$
Saída: Solução por construção s

```
1  $s \leftarrow \emptyset$ ;  
2  $M \leftarrow \text{ConstroiGrupos}()$ ;  
3 para cada  $(\mu \in M)$  faça  
4    $CL \leftarrow \text{ConstroiListaCandidatos}(\mu)$ ;  
5   enquanto  $(|CL| > 0)$  faça  
6      $g_{\min} = \min\{g(j) \mid j \in CL\}$ ;  
7      $g_{\max} = \max\{g(j) \mid j \in CL\}$ ;  
8      $RCL \leftarrow \{j \in CL \mid g(j) \leq g_{\min} + \alpha(g_{\max} - g_{\min})\}$ ;  
9     Selecione uma tarefa  $j \in RCL$  aleatoriamente ;  
10     $s \leftarrow s \cup \{j\}$ ;  
11     $CL \leftarrow \text{ConstroiListaCandidatos}(\mu)$ ;  
12  fim  
13 fim  
14 retorna  $s$ ;
```

No início do algoritmo, uma solução s é inicializada. Logo em seguida, uma lista M de grupos de passos é construída pela função descrita no Algoritmo 3. Para cada grupo $\mu \in M$, uma lista de candidatos CL é gerada por meio da função descrita no Algoritmo 4.

A seguir, entra-se em um laço de repetição até que todas as tarefas candidatas de CL do grupo μ sejam avaliadas. A cada iteração, forma-se uma Lista Restrita de Candidatos RCL a partir da lista CL . Cada tarefa $j \in CL$, que obedece a inequação $g(j) \leq g_{\min} + \alpha(g_{\max} - g_{\min})$, é incluída em RCL . Nesta inequação, $g(j)$ é o valor da função g aplicada à tarefa j , g_{\min} e g_{\max} são, respectivamente, o menor e maior valor da função g dentre todas as tarefas da CL . A descrição desta função é apresentada mais adiante. Por sua vez, α é um parâmetro do método, com valor no intervalo $[0, 1]$, que define o grau de aleatoriedade do procedimento construtivo. Após a construção da RCL , uma tarefa $j \in RCL$ é selecionada aleatoriamente para compor a solução s e, posteriormente, a lista de candidatos CL do grupo μ é atualizada. O procedimento termina quando todos os grupos da lista M sejam analisados. Ao final do algoritmo, uma solução construída s é retornada.

O algoritmo utilizado para implementar a função *ConstroiGrupos* tem seu pseudocódigo descrito pelo Algoritmo 3. Esse algoritmo constrói uma lista M de grupos de passos. Cada grupo $\mu \in M$ é definido por uma tupla (h, G) , sendo $h \in SH$ um navio e G um conjunto de passos consecutivos de um mesmo produto a serem embarcados no navio h .

Na linha 1 o conjunto M de grupos é inicializado com vazio. Entre as linhas 2 e 13

Algoritmo 3: ConstroiGrupos

Saída: Lista M de grupos de passos

```
1  $M \leftarrow \emptyset$ ;  
2 para cada ( $h \in SH$ ) faça  
3    $G \leftarrow \emptyset$ ;  
4   para cada ( $\phi \in SP_h$ ) faça  
5      $G \leftarrow G \cup \{\phi\}$ ;  
6      $q \leftarrow \text{Produto}(\phi)$ ;  
7      $q_{prox} \leftarrow \text{Produto}(\phi + 1)$ ;  
8     se ( $q \neq q_{prox}$ ) então  
9        $M \leftarrow M \cup \{(h, G)\}$ ;  
10       $G \leftarrow \emptyset$ ;  
11     fim  
12   fim  
13 fim  
14 retorna  $M$ ;
```

é executado um laço de repetição sobre todos os navios $h \in SH$. Na linha 3 o conjunto G de passos é inicializado com vazio. Entre as linhas 4 e 12 um laço de repetição é executado sobre todos os passos do navio h . A seguir, o passo ϕ é inserido no conjunto G na linha 5. Em seguida, o produto a ser embarcado no passo ϕ é identificado (isto é, atribuído à q) e na próxima linha, o produto a ser embarcado no próximo passo ($\phi + 1$) é identificado e atribuído à q_{prox} . Se a condição verificada na linha 8 for verdadeira, então os passos consecutivos ϕ e $\phi + 1$ são de produtos diferentes. Portanto, encerra-se uma sequência de passos consecutivos do produto q , a tupla (h, G) é inserida no conjunto M (linha 9) e G é reinicializado (linha 10).

Para facilitar a compreensão do Algoritmo 3, considere o embarque de um navio exemplo. O embarque desse navio está previamente dividido em 8 passos $(1, 2, \dots, 7, 8)$, conforme ilustrado na Figura 4.2. Por exemplo, o passo 1 embarcará 800 toneladas do produto A no porão 1, o passo 2 embarcará 800 toneladas do produto B no porão 2 e o passo 3 embarcará 700 toneladas do produto A no porão 1.

Navio h	Passo ϕ	1	2	3	4	5	6	7	8
	Porão θ	1	2	1	3	4	2	3	1
	Quantidade $sizeStep_\phi$	800	800	700	800	900	800	900	1200
	Produto q	A	B	A	A	B	B	A	A
1	Grupo $\mu = (h, G)$	$(1, \{1\})$	$(1, \{2\})$	$(1, \{3, 4\})$		$(1, \{5, 6\})$		$(1, \{7, 8\})$	

Figura 4.2: Exemplo de lista de grupos de passos do embarque um navio. Fonte: Autor (2021).

O grupo 1 é formado somente pelo passo 1, que embarcará o produto A. Na sequência, o grupo 2 é formado apenas pelo passo 2, que embarcará o produto B. O grupo 3 é formado pelos

passos 3 e 4, já que eles são consecutivos e de um mesmo produto (A). O grupo 4 é formado pelos passos 5 e 6 do produto B. Por fim, o grupo 5 é formado pelos passos consecutivos 7 e 8 do produto A. Ao final, tem-se $M = \{(1, \{1\}), (1, \{2\}), (1, \{3, 4\}), (1, \{5, 6\}), (1, \{7, 8\})\}$.

O Algoritmo 4 mostra como construir uma lista CL de tarefas candidatas a partir de um grupo μ de passos.

Algoritmo 4: ConstroiListaCandidatos

Entrada: Grupo μ de passos
Saída: Lista CL de tarefas candidatas

```

1  $CL \leftarrow \emptyset$ ;
2  $q \leftarrow \text{Produto}(\mu)$ ;
3  $P^q \leftarrow \{a \in P \mid (\text{Produto}(a) = q) \text{ e } (\text{sizePile}_a^{\text{real}} < \text{sizePile}_a)\}$ ;
4 para cada  $a \in P^q$  faça
5      $\text{sizeJob} \leftarrow \text{ObtemQuantidade}(\mu, a)$ ;
6      $RC^a \leftarrow \{i \in RC \mid \text{elegivel}(a, i)\}$ ;
7     para cada  $i \in RC^a$  faça
8          $j \leftarrow (a, i, \text{sizeJob})$ ;
9          $CL \leftarrow CL \cup \{j\}$ ;
10    fim
11 fim
12 retorna  $CL$ ;

```

Na linha 1 é atribuída uma lista vazia a CL . Na linha 2 é atribuído à q o produto a ser embarcado pelos passos do grupo μ . Em seguida, na linha 3, a lista de pilhas que contém o produto q e que não foram totalmente embarcadas é atribuída à P^q . Um laço de repetição ocorre entre as linhas 4 e 11 para todas as pilhas $a \in P^q$. Na linha 5, a quantidade a ser recuperada na pilha a é obtida pela função $\text{ObtemQuantidade}(\mu, a)$, descrita pelo Algoritmo 5, e atribuída à sizeJob . Em seguida, na linha 6, é formada a lista RC^a de recuperadoras elegíveis para recuperar a pilha a . Entre as linhas 7 e 10 é executado um laço de repetição sobre todas as recuperadoras $i \in RC^a$. Na linha 8 é, então, formada uma tarefa j pela associação da pilha a com a recuperadora i e a quantidade sizeJob a ser recuperada dessa pilha. Em seguida, na linha 9, a tarefa j é inserida na lista de candidatos CL . Ao final do algoritmo, a lista CL é retornada na linha 12.

O pseudocódigo do algoritmo que implementa a função $\text{ObtemQuantidade}(\mu, a)$ está descrito no Algoritmo 5.

O algoritmo possui como parâmetro de entrada um grupo μ de passos e uma pilha a e como parâmetro de saída a quantidade sizeJob a ser recuperada. Na linha 1, é atribuído à tupla (h, G) o grupo μ , sendo h o navio e G o conjunto de passos consecutivos. A quantidade programada para ser embarcada pelo grupo é calculada na linha 2 e atribuída à sizeGroup_μ . A quantidade já embarcada por este grupo μ é calculada pela soma das quantidades embarcadas anteriormente nos passos $\phi \in G$ e atribuída à $\text{sizeGroup}_\mu^{\text{real}}$ na linha 3. Por sua vez, a quantidade restante a embarcar pelo grupo é calculada como a diferença entre a quantidade programada e a quantidade já embarcada na linha 4 e atribuída à $\text{sizeGroup}_\mu^{\text{remain}}$. De forma análoga aos cálculos

Algoritmo 5: ObtemQuantidade

Entrada: Grupo μ de passos, Pilha a
Saída: Quantidade $sizeJob$ a ser recuperada

- 1 $(h, G) \leftarrow \mu$;
- 2 $sizeGroup_{\mu} \leftarrow \sum_{\phi=1}^{|G|} sizeStep_{\phi}$;
- 3 $sizeGroup_{\mu}^{real} \leftarrow \sum_{\phi=1}^{|G|} QuantidadeEmbarcada(\phi)$;
- 4 $sizeGroup_{\mu}^{remain} \leftarrow sizeGroup_{\mu} - sizeGroup_{\mu}^{real}$;
- 5 $sizePile_a^{real} \leftarrow QuantidadeEmbarcada(a)$;
- 6 $sizePile_a^{remain} \leftarrow sizePile_a - sizePile_a^{real}$;
- 7 **se** $(sizePile_a^{remain} > sizeGroup_{\mu}^{remain})$ **então**
- 8 | $sizeJob \leftarrow \frac{sizeGroup_{\mu}^{remain}}{max_h}$;
- 9 **fim**
- 10 **senão**
- 11 | $sizeJob \leftarrow sizePile_a^{remain}$;
- 12 **fim**
- 13 **retorna** $sizeJob$;

das quantidades já embarcadas e restantes do grupo μ , as quantidades embarcada e restante da pilha a são calculadas nas linhas 5 e 6. Se a quantidade restante na pilha for maior que a restante do grupo μ a ser embarcada, então, na linha 8, a quantidade $sizeJob$ a ser embarcada é definida pela divisão entre a quantidade restante a ser embarcada no grupo μ e o número de embarques simultâneos permitido para o navio h (max_h). Caso contrário, na linha 11, $sizeJob$ recebe a quantidade restante na pilha. Ao final do algoritmo, a quantidade a recuperar é retornada na linha 13.

Para ilustrar o funcionamento do Algoritmo 4, considere a lista $M = \{(1, \{1\}), (1, \{2\}), (1, \{3, 4\}), (1, \{5, 6\}), (1, \{7, 8\})\}$, gerada no exemplo da Figura 4.2. Como dados adicionais, considere que existem quatro pilhas disponíveis. As pilhas 1 e 2 são formadas pelo produto A , sendo $sizePile_1 = 2000t$ e $sizePile_2 = 2500t$. Já as pilhas 3 e 4 são formadas pelo produto B , sendo $sizePile_3 = 1200t$ e $sizePile_4 = 1300t$. As pilhas 1 e 3 podem ser recuperadas pelas recuperadoras 1 e 2, a pilha 2 pode ser recuperada apenas pela recuperadora 1 e, por fim, a pilha 4 apenas pela recuperadora 2, sendo as taxas de recuperação $rate_1 = 200t/h$ e $rate_2 = 250t/h$.

A lista de candidatos CL para o primeiro grupo $\mu \in M$, $\mu = (1, \{1\})$, é obtida conforme a seguir. Após a execução das linhas 1, 2 e 3, $CL = \emptyset$, $q = A$ e $P^q = \{1, 2\}$. Na execução da primeira iteração do laço entre as linhas 4 e 11, $a = 1$. Após a chamada da função $ObtemQuantidade((1, \{1\}), 1)$ na linha 5, $sizeJob = 800$. $RC^a = \{1, 2\}$, após a execução da linha 6. Na execução da primeira iteração do laço entre as linhas 7 e 10, $i = 1$. Após a execução das linhas 8 e 9, $j = (1, 1, 800)$ e $CL = \{(1, 1, 800)\}$. Na execução da segunda iteração, $i = 2$, $j = (1, 2, 800)$ e $CL = \{(1, 1, 800), (1, 2, 800)\}$. Na execução da segunda iteração do laço entre as linhas 4 e 11, $a = 2$. Na execução das próximas linhas, $sizeJob = 800$ e $RC^a = \{1\}$. O laço entre as linhas 7 e 10 possui somente uma iteração, $i = 1$. Após a execução das linhas 8 e 9,

$j = (2, 1, 800)$ e $CL = \{(1, 1, 800), (1, 2, 800), (2, 1, 800)\}$.

Para compreensão do Algoritmo 2 entre as linhas 6 e 10, são apresentados os passos para inserção de uma tarefa j na solução s . É utilizada a lista $M = \{(1, \{1\}), (1, \{2\}), (1, \{3, 4\}), (1, \{5, 6\}), (1, \{7, 8\})\}$, $\mu = (1, \{1\})$, $CL = \{(1, 1, 800), (1, 2, 800), (2, 1, 800)\}$, $\alpha = 0,3$ e a função gulosa $g(\cdot)$ dada pela Equação (4.2).

$$g(k) = \max\{r_k, C_i + S_{ijk}^{a_1 a_2}\} + p_{ik} \quad \forall i \in RC, k = (i, a_2, sizeJob_k) \in CL, a_1, a_2 \in P \quad (4.2)$$

na qual r_k é o instante de disponibilidade da tarefa k , C_i é o tempo de término da recuperadora i antes de alocar a tarefa $k \in CL$, p_{ik} é o tempo de processamento da tarefa k na recuperadora i e $S_{ijk}^{a_1 a_2}$ é o tempo necessário para a recuperadora i se deslocar da tarefa $j = (a_1, i)$ para a tarefa $k = (a_2, i)$. Para simplificar a compreensão, o instante de disponibilidade da tarefa k é 0, o tempo de término da recuperadora i é 0 e o tempo de deslocamento entre as tarefas é 0, ou seja, $r_k = 0 \quad \forall k \in N$, $C_i = 0 \quad \forall i \in RC$ e $S_{ijk}^{a_1 a_2} = 0 \quad \forall a_1, a_2 \in P, \forall i \in RC, \forall j, k \in N$. Aplicando a função gulosa $g(\cdot)$ às tarefas da lista CL , tem-se:

$$g((1, 1, 800)) = \max\{0, 0 + 0\} + 800/200 = 4$$

$$g((1, 2, 800)) = \max\{0, 0 + 0\} + 800/250 = 3,2$$

$$g((2, 1, 800)) = \max\{0, 0 + 0\} + 800/200 = 4$$

Nos passos seguintes, obtém-se:

$$g_{\min} = 3,2 \text{ e } g_{\max} = 4, g(j) \leq 3,44$$

$$RCL = \{(1, 2, 800)\}$$

Como RCL possui somente um elemento, portanto $j = (1, 2, 800)$. Desta forma, $s = \{(1, 2, 800)\}$.

4.4. Estrutura de Vizinhança

Para exploração do espaço de busca de soluções, foram propostas duas estruturas de vizinhança, denominadas \mathcal{N}_1 e \mathcal{N}_2 . Ambas são descritas a seguir:

1. \mathcal{N}_1 - **Múltipla inserção**: Esta vizinhança contém os vizinhos da solução corrente gerados a partir do movimento de realocação de uma tarefa de uma recuperadora para uma outra posição da mesma recuperadora ou para uma recuperadora diferente. Um novo vizinho é gerado somente se a nova solução obtida for factível.
2. \mathcal{N}_2 - **Troca em máquinas diferentes**: Esta vizinhança contém os vizinhos da solução corrente gerados a partir da aplicação do movimento de troca de duas tarefas entre recuperadoras diferentes. Um novo vizinho é gerado somente se a nova solução obtida for factível.

4.5. Buscal Local

O método de busca local utilizado na linha 6 do Algoritmo 1 é baseado na heurística *Basic Variable Neighborhood Descent* (BVND) (HANSEN *et al.*, 2017) com a estratégia *First Improvement* (FI). As estruturas de vizinhanças são ordenadas de acordo com a complexidade da busca. Essa estratégia é comumente aplicada em métodos VND (HANSEN e MLADENOVIC, 2014). Sendo o movimento de troca composto por dois movimentos de inserção, a primeira vizinhança é a de inserção (\mathcal{N}_1) e a segunda é de troca (\mathcal{N}_2). O método BVND é caracterizado por fazer trocas sistemáticas de estruturas de vizinhanças, explorando-as em ordem sequencial. A solução retornada pelo método é um ótimo local segundo todas as estruturas de vizinhanças exploradas. Seu pseudocódigo está descrito pelo Algoritmo 6.

Algoritmo 6: Heurística BVND

Entrada: Solução inicial s , Função de avaliação $f(\cdot)$, Conjunto de vizinhanças $\mathcal{N}(\cdot)$

Saída: Solução refinada s

```
1  $k \leftarrow 1$ 
2 enquanto ( $k \leq |\mathcal{N}|$ ) faça
3    $s' \leftarrow \text{FI}(s, f(s), \mathcal{N}_k)$ ;
4   se ( $f(s') < f(s)$ ) então
5      $s \leftarrow s'$ ;
6      $k \leftarrow 1$ ;
7   fim
8   senão
9      $k \leftarrow k + 1$ ;
10  fim
11 fim
12 retorna  $s$ ;
```

O Algoritmo 6 recebe como parâmetros uma solução inicial s , uma função de avaliação $f(\cdot)$ e o conjunto de vizinhanças $\mathcal{N}(\cdot)$. Um laço de repetição entre as linhas 2 e 11 ocorre enquanto há uma estrutura de vizinhança a ser explorada. Na linha 3, é aplicada a estratégia FI, implementada pelo método $\text{FI}(s, f(s), \mathcal{N}_k)$, sobre a vizinhança k da solução corrente s . Se a solução vizinha s' encontrada é melhor do que a solução corrente s (linha 4), então s recebe s' e retorna-se a busca à primeira estrutura de vizinhança ($k \leftarrow 1$). Caso contrário, passa-se para a próxima estrutura de vizinhança (linha 9). O laço é interrompido quando não houver mais vizinhanças para serem exploradas. No final do algoritmo é retornada a solução refinada s .

4.6. Implementação do Sistema de Suporte à Decisão

Para implementar o algoritmo proposto, foi desenvolvido um SSD na linguagem C# sobre a plataforma .NET Framework 4.6.1 (MICROSOFT, 2020), utilizando o ambiente de

desenvolvimento Visual Studio 2017 Community.

A Figura 4.3 exibe a interface do SSD. A interface do sistema permite que o usuário entre com os seguintes parâmetros: (i) Número de execuções da instância de teste no campo [*Executions*], (ii) Parâmetro k no campo [k], (iii) Parâmetro α no campo [*Grasp Alpha*]. O campo [*Grasp Iterations*] exibe o número máximo de iterações sem melhora *iterMax* utilizado no Algoritmo 1. Após a entrada dos parâmetros, o usuário clica no botão [*GRASP*] e o sistema exibe a solução encontrada na forma de um gráfico de *Gantt* e o valor da função objetivo no campo [*Fo*].

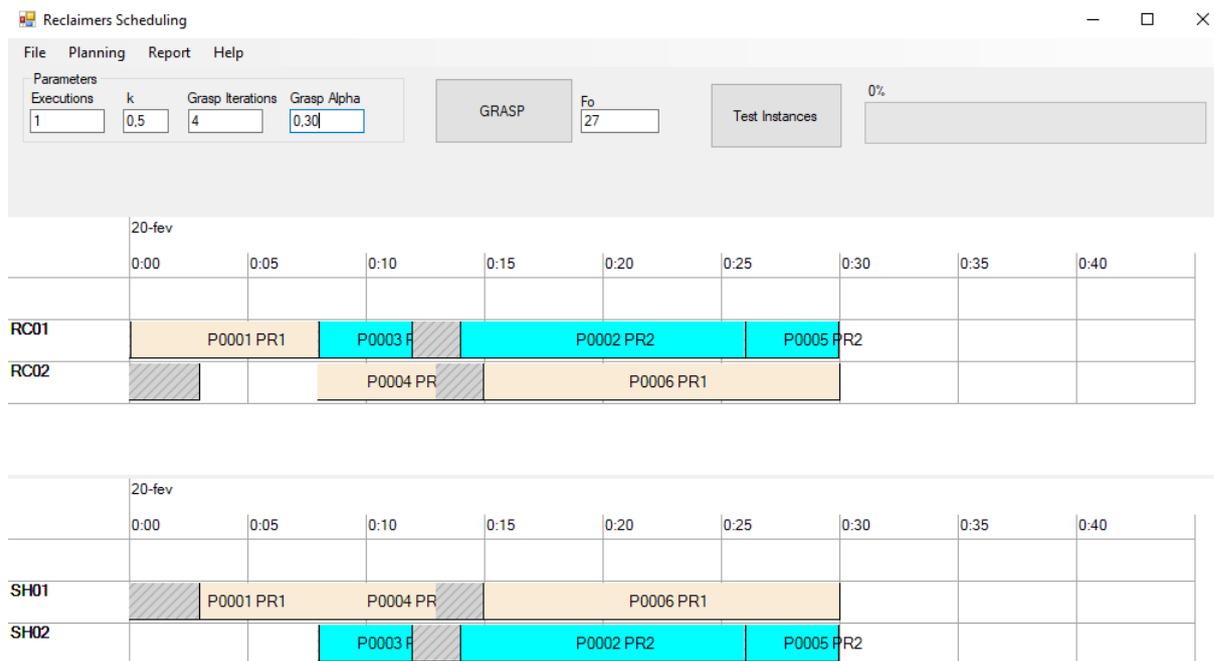


Figura 4.3: Interface do sistema de suporte à decisão. Fonte: Autor (2021).

5. Experimentos Computacionais

Para testar o algoritmo proposto, foram selecionadas 19 instâncias de teste a partir dos dados operacionais do Porto de Tubarão. O porto possui 23 pátios de estocagem e 11 recuperadoras de minério de ferro. As instâncias de teste possuem até 56 pilhas e até 10 navios a serem embarcados. A quantidade de minério de ferro a ser embarcada varia entre 55kt e 1600kt e o horizonte de planejamento varia entre 2 e 224 horas. A Tabela 5.1 detalha as características da instância. A coluna “Instância” mostra a identificação da instância. As colunas “# Pilhas”, “# Produtos” e “# Navios” apresentam o número de pilhas, o número de produtos e navios, respectivamente. A coluna “Total embarcado” informa a quantidade de minério de ferro embarcado em kt. Finalmente, a coluna “Horizonte de planejamento” exibe o horizonte de planejamento em horas.

Tabela 5.1: Características das instâncias. Fonte: Autor (2021).

Instância	# Pilhas	# Produtos	# Navios	Total embarcado (kt)	Horizonte de planejamento (h)
01	8	2	3	410,500	59,1
02	12	5	3	507,450	37,4
03	18	5	4	724,314	25,0
04	22	7	5	835,314	8,3
05	24	7	7	757,546	29,6
06	12	4	4	250,096	29,6
07	25	4	4	622,600	22,0
08	34	5	5	790,750	33,3
09	8	4	2	114,550	53,6
10	32	7	5	900,610	100,0
11	7	4	2	80,000	77,6
12	16	3	3	351,587	22,3
13	19	2	2	685,650	2,1
14	35	5	5	804,542	70,3
15	23	4	5	785,469	224,5
16	41	6	9	1189,552	70,3
17	8	1	1	393,400	3,4
18	56	10	10	1685,514	100,0
19	14	2	2	465,000	56,5

Os testes do SSD foram realizados em um computador Dell Inspiron 15-7000 Series com um processador Intel Core i7-8550U, com 16GB de RAM e sistema operacional Windows 10 de 64 bits.

A Seção 5.1 descreve como os parâmetros dos algoritmos foram obtidos. A Seção 5.2 mostra os resultados do algoritmo.

5.1. Calibração dos Parâmetros

Dado o caráter estocástico do algoritmo proposto e a diversidade de instâncias, seus parâmetros (α e k) foram calibrados para obter seu melhor desempenho. Para esta tarefa foi utilizado o pacote *irace* (LÓPEZ-IBÁÑEZ *et al.*, 2016). Das 19 instâncias usadas para teste, foram selecionadas três com tamanhos diferentes (pequeno, médio e grande) para a calibração. Os valores dos parâmetros testados foram: $\alpha \in \{0, 20; 0, 30; 0, 40\}$ e $k \in \{0, 30; 0, 40; 0, 50; \dots; 0, 80\}$. Ao final do ajuste, os valores retornados pelo pacote foram: $\alpha = 0, 30$ e $k = 0, 50$.

5.2. Resultado do Algoritmo

O algoritmo proposto foi executado 30 vezes para cada uma das 19 instâncias. A Tabela 5.2 exibe os resultados do algoritmo para cada execução em relação à função objetivo dada pela Equação (4.1). A coluna “Instância” mostra a identificação da instância. As colunas “Melhor” e “Média” apresentam os melhores resultados e a média, respectivamente, em 30 execuções do algoritmo. As colunas “Desv. pad.” e “RPD” mostram o desvio padrão e o desvio percentual relativo (RPD). Por fim, a coluna “Tempo médio” mostra o tempo médio de execução do algoritmo em segundos.

A métrica RPD é avaliada de acordo com a Equação abaixo:

$$RPD_i = \frac{\bar{f}_i - f_i^*}{f_i^*} \quad (5.1)$$

onde \bar{f}_i e f_i^* representam, respectivamente, a média e o melhor resultado encontrado pelo algoritmo na instância i .

A Figura 5.1 exibe um gráfico *box plot* dos resultados do algoritmo e que mostra a variabilidade das soluções geradas. Como pode ser observado, o maior valor para a mediana do RPD foi menor do que 6%.

A Tabela 5.3 mostra o menor tempo de atracação obtido por cada navio na melhor solução encontrada em cada instância. A coluna “Instância” identifica a instância. As colunas “1^o, 2^o, ..., 10^o” apresentam o menor tempo de atracação do navio, em horas, ordenado pelo instante de atracação. Desta forma, os tempos de atracação dos dois primeiros navios são de 17,8 e 11,2 horas na melhor solução da instância 3.

Por sua vez, a Tabela 5.4 exibe as mesmas características, assim como a tabela anterior, porém relacionada à quantidade de minério embarcada em cada navio, em kt.

As Figuras 5.2 (a) e (b) ilustram o sequenciamento das recuperadoras para uma solução da instância 3 sob o horizonte das recuperadoras e dos navios, respectivamente. Na figura, cada retângulo colorido representa uma pilha a ser embarcada em um navio. Os retângulos hachurados em cinza representam os tempos de preparação. Por exemplo, a pilha “B2345 SFHT” foi a última programada para ser recuperada pela RC01 e embarcada no navio SH03-02.

Tabela 5.2: Resultados algoritmo proposto. Fonte: Autor (2021).

Instância ID	Função objetivo			RPD	Tempo médio (s)
	Melhor	Médio	Desv. pad.		
01	1857,00	1907,53	25,19	2,7%	0,25
02	2040,00	2090,67	52,95	2,5%	1,47
03	2011,00	2115,80	60,53	5,2%	6,88
04	1970,00	2067,53	53,56	5,0%	23,13
05	1614,00	1654,03	35,54	2,5%	28,09
06	585,00	590,10	4,54	0,9%	1,56
07	1492,00	1507,93	13,15	1,1%	22,26
08	1406,00	1441,83	30,93	2,5%	84,71
09	241,00	250,37	6,27	3,9%	0,24
10	1496,00	1541,60	28,25	3,0%	40,98
11	128,00	131,10	3,82	2,4%	0,08
12	1244,00	1280,33	18,15	2,9%	3,82
13	2293,00	2339,63	23,84	2,0%	7,08
14	1723,00	1745,47	10,57	1,3%	206,43
15	1199,00	1208,07	9,44	0,8%	22,37
16	1169,00	1176,47	4,50	0,6%	307,19
17	2534,00	2570,23	23,34	1,4%	0,18
18	1555,00	1604,90	18,65	3,2%	556,33
19	1766,00	1775,93	2,18	0,6%	1,61

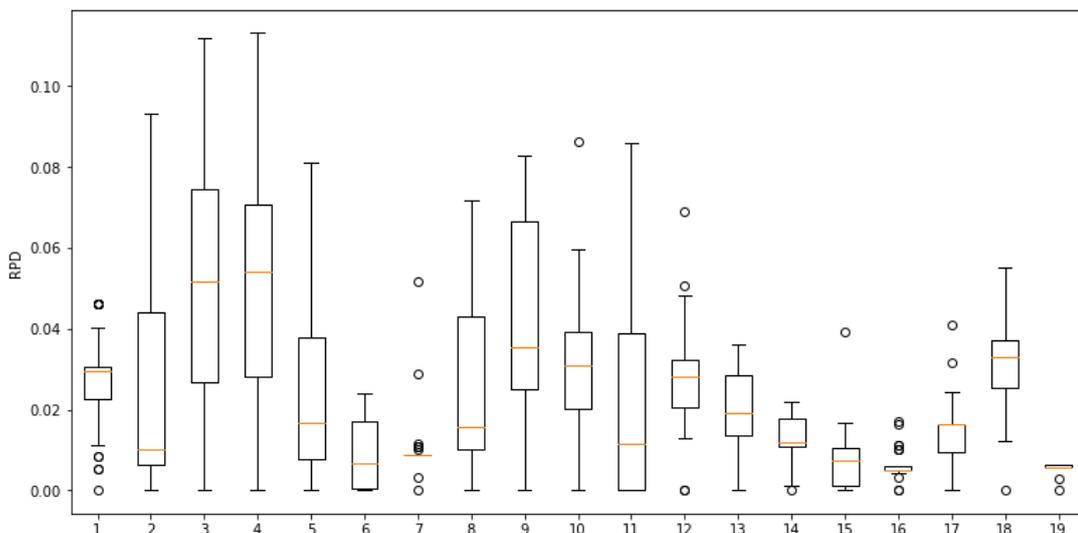


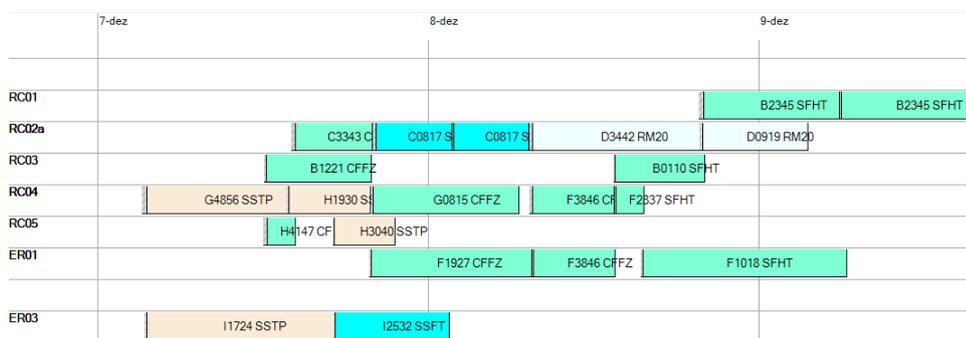
Figura 5.1: Gráfico *box plot* dos resultados do algoritmo por instância. Fonte: Autor (2021).

Tabela 5.3: Tempo de atracação por navio na melhor solução encontrada por instância (horas).
Fonte: Autor (2021).

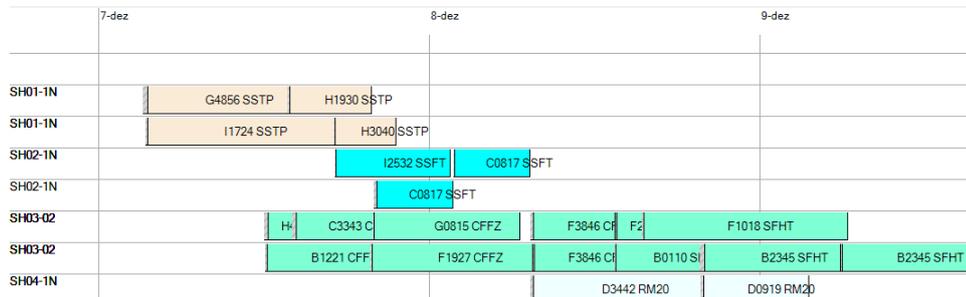
Instância ID	Ordem de atracação do navio									
	1 ^o	2 ^o	3 ^o	4 ^o	5 ^o	6 ^o	7 ^o	8 ^o	9 ^o	10 ^o
1	42,4	7,5								
2	13,6	47,0								
3	17,8	11,2	47,1	20,0						
4	18,1	11,1	47,1	20,0	21,1					
5	10,4	7,9	13,6	8,4	46,9	12,1				
6	9,9	7,2	8,4	12,1						
7	33,3	21,8	9,7	5,8						
8	34,1	21,8	15,6	8,1	11,5					
9	14,1									
10	15,0	32,6	13,9	13,6	27,7					
11	9,8									
12	27,7	9,8	6,0							
13	42,3	31,7								
14	40,8	27,7	13,4	10,0	6,0					
15	43,5	12,1	32,4							
16	43,5	27,7	12,1	13,4	30,4	10,0	6,0			
17	42,2									
18	31,8	32,4	24,0	14,6	10,4	8,7	9,3	10,0	28,3	22,0
19	36,8	14,6								

Tabela 5.4: Total embarcado por navio na melhor solução encontrada por instância (kt). Fonte: Autor (2021).

Instância ID	Ordem de atracação do navio									
	1 ^o	2 ^o	3 ^o	4 ^o	5 ^o	6 ^o	7 ^o	8 th	9 ^o	10 ^o
1	374,500	36,000								
2	114,450	393,000								
3	164,250	85,750	393,000	81,314						
4	164,250	85,750	393,000	81,314	111,000					
5	71,316	57,530	114,450	39,300	393,000	81,950				
6	71,316	57,530	39,300	81,950						
7	297,000	202,600	80,000	43,000						
8	297,000	202,600	120,000	74,150	97,000					
9	114,550									
10	114,550	292,060	130,000	103,000	261,000					
11	80,000									
12	233,087	99,500	48,000							
13	393,400	292,250								
14	376,395	233,087	99,500	77,500	48,000					
15	393,400	99,819	292,250							
16	393,400	233,087	99,819	99,500	269,246	77,500	48000			
17	393,400									
18	297,000	309,000	202,000	117,000	81,314	81,950	58,800	81,150	263,600	193,700
19	341,000	124,000								



(a) Horizonte das recuperadoras



(b) Horizonte dos navios

Figura 5.2: Sequenciamento das recuperadoras.

Fonte: Autor (2021)

6. Conclusões e Trabalhos Futuros

6.1. Conclusões

Este trabalho abordou o problema de sequenciamento de recuperadoras de minério no Porto de Tubarão com o objetivo de minimizar a soma ponderada das diferenças entre a data de conclusão e atracação dos navios.

O sequenciamento de recuperadoras no Porto de Tubarão é um problema importante para a sua eficiência operacional, pois o porto é capaz de embarcar vários navios com demandas diferentes simultaneamente. Essas demandas são caracterizadas pela natureza e quantidade de minério de ferro a ser embarcado. Demandas diferentes e simultâneas dos navios fazem com que haja competição de recursos para recuperação de minério de ferro das pilhas.

Este problema é de natureza combinatória e pertencente à classe de problemas NP-difícil, portanto, para resolvê-lo, foi proposto o desenvolvimento de um SSD que implementa a metaheurística GRASP. Este algoritmo aplica o BVND como um método de busca local. O BVND explora o espaço da solução por meio de vários movimentos de inserção e troca em diferentes máquinas. Os testes do algoritmo foram realizados com 19 instâncias reais de Porto de Tubarão. Os parâmetros do algoritmo foram calibrados devido ao caráter estocástico do GRASP e das diferentes instâncias testadas.

Nos testes, foi possível encontrar boas soluções, em todas as instâncias, com um desvio percentual relativo de no máximo 5,2%. Além da boa qualidade da solução gerada pelo algoritmo, o tempo gasto para encontrá-la foi pequeno, inferior a 10 minutos, considerado adequado para a tomada de decisão. Os resultados obtidos validam, assim, o algoritmo proposto para solução do problema e aplicação real no suporte à programação de recuperadoras do porto.

6.2. Trabalhos Futuros

Como trabalhos futuros, são propostas as seguintes atividades:

- Adaptação da aplicação computacional em outros portos de granéis sólidos e pátios de estocagem das plantas de beneficiamento que utilizam recuperadoras;
- Inclusão de mais características operacionais do porto como empilhamento e recirculação de pilhas de minério e suas interferências com as operações de recuperação. A recirculação de pilhas consiste na recuperação de pilhas que são enviadas para empilhamento em outras áreas dos pátios de estocagem.

Referências Bibliográficas

- ABU-MARRUL, V., MARTINELLI, R., HAMACHER, S., et al.. “**Matheuristics for a parallel machine scheduling problem with non-anticipatory family setup times: Application in the offshore oil and gas industry**”, *Computers and Operations Research*, v. 128, pp. 105162, 2021. ISSN: 03050548. doi: 10.1016/j.cor.2020.105162.
- AFZALIRAD, M., REZAEIAN, J. “**Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions**”, *Computers and Industrial Engineering*, v. 98, pp. 40–52, 2016. ISSN: 03608352. doi: 10.1016/j.cie.2016.05.020.
- ANGELELLI, E., KALINOWSKI, T., KAPOOR, R., et al.. “**A reclaimer scheduling problem arising in coal stockyard management**”, *Journal of Scheduling*, v. 19, n. 5, pp. 563–582, 2016. ISSN: 10946136. doi: 10.1007/s10951-015-0436-y.
- ANOKIĆ, A., STANIMIROVIĆ, Z., STAKIĆ, D., et al.. “**Metaheuristic approaches to a vehicle scheduling problem in sugar beet transportation**”, *Operational Research*, n. 0123456789, 2019. ISSN: 18661505. doi: 10.1007/s12351-019-00495-z.
- ANTAQ. *Anuário Estatístico*. Relatório técnico, Agência Nacional de Transportes Aquaviários, 2021. Disponível em <http://www.gov.br/anta>, Acessado em 12/07/2021.
- AVALOS-ROSALES, O., ANGEL-BELLO, F., ALVAREZ, A. “**Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times**”, *The International Journal of Advanced Manufacturing Technology*, v. 76, n. 9-12, pp. 1705–1718, 2015. doi: 10.1007/s00170-014-6390-6.
- BENDERS, J. F. “**Partitioning procedures for solving mixed-variables programming problems**”, *Computational Management Science*, v. 2, n. 1, pp. 3–19, 2005. doi: 10.1007/s10287-004-0020-y.
- BOLAND, N., GULCZYNSKI, D., SAVELSBERGH, M. “**A stockyard planning problem**”, *EURO Journal on Transportation and Logistics*, v. 1, n. 3, pp. 197–236, 2012. ISSN: 2192-4376. doi: 10.1007/s13676-012-0011-z.

- BURDETT, R. L., CORRY, P., YARLAGADDA, P. K., et al.. “**A flexible job shop scheduling approach with operators for coal export terminals**”, *Computers and Operations Research*, v. 104, pp. 15–36, apr 2019. ISSN: 03050548. doi: 10.1016/j.cor.2018.11.019.
- BURDETT, R. L., CORRY, P., EUSTACE, C., et al.. “**A flexible job shop scheduling approach with operators for coal export terminals – A mature approach**”, *Computers and Operations Research*, v. 115, pp. 104834, 2020. ISSN: 03050548. doi: 10.1016/j.cor.2019.104834.
- COMEXVIS. “**Exportação de Minério de Ferro e Seus Concentrados**”. 2021. Comércio Exterior, Ministério da Economia. Disponível em <http://comexstat.mdic.gov.br/pt/comex-vis>. Acessado em 12/07/2021.
- COTA, L. P., COELHO, V. N., GUIMARÃES, F. G., et al.. “**Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times**”, *International Transactions in Operational Research*, 2018. doi: 10.1111/itor.12566.
- COTA, L. P., GUIMARÃES, F. G., RIBEIRO, R. G., et al.. “**An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem**”, *Swarm and Evolutionary Computation*, v. 51, pp. 100601, dec 2019. doi: 10.1016/j.swevo.2019.100601.
- FAEDA, F. M. *Métodos de Resolução do Problema De Sequenciamento em Máquinas Paralelas não-relacionadas com Restrições de Precedência e Tempos De Preparação*. Tese de Doutorado, Universidade Federal de Viçosa, 2015.
- FANJUL-PEYRO, L., RUIZ, R., PEREA, F. “**Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times**”, *Computers & Operations Research*, v. 101, pp. 173 – 182, 2019. ISSN: 0305-0548. doi: 10.1016/j.cor.2018.07.007.
- FISCHETTI, M., MONACI, M. “**Proximity search for 0-1 mixed-integer convex programming**”, *Journal of Heuristics*, v. 20, n. 6, pp. 709–731, 2014. doi: 10.1007/s10732-014-9266-x.
- GAREY, M. R., JOHNSON, D. S. *Computers and intractability: A Guide to the Theory of NP-Completeness*, v. 174. Freeman San Francisco, 1979.
- GLOVER, F. “**Tabu search and adaptive memory programming - Advances, applications and challenges**”. Em: Barr, R. S., Helgason, R. V., Kennington, J. L. (Eds.), *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer*

Science and Operations Research, Kluwer Academic Publishers, pp. 1–75, 1996. doi: 10.1007/978-1-4615-4102-8_1.

GUIMARÃES, I. F., OUAZENE, Y., DE SOUZA, M. C., et al.. “**Flowshop scheduling problem with parallel semi-lines and final synchronization operation**”, *Computers and Operations Research*, v. 108, pp. 121–133, 2019. ISSN: 03050548. doi: 10.1016/j.cor.2019.04.011.

HANSEN, P., MLADENOVIĆ, N. “**Variable Neighborhood Search**”. Em: Burke, E., Kendall, G. (Eds.), *Search Methodologies*, Introductory Tutorials in Optimization and Decision Support Techniques, Springer, cap. 12, pp. 313–337, New York, USA, 2014. doi: 10.1007/978-1-4614-6940-7.

HANSEN, P., MLADENOVIĆ, N., TODOSIJEVIĆ, R., et al.. “**Variable neighborhood search: basics and variants**”, *EURO Journal on Computational Optimization*, v. 5, pp. 423–454, 2017. doi: 10.1007/s13675-016-0075-x.

HOLLAND, J. H. “**Genetic Algorithms**”, *Scientific American*, v. 267, n. 1, pp. 66–73, 1992.

HU, D., YAO, Z. “**Stacker-reclaimer scheduling in a dry bulk terminal**”, *International Journal of Computer Integrated Manufacturing*, v. 25, n. 11, pp. 1047–1058, nov 2012. ISSN: 0951192X. doi: 10.1080/0951192X.2012.684707.

KALINOWSKI, T., KAPOOR, R., SAVELSBERGH, M. W. “**Scheduling reclaimers serving a stock pad at a coal terminal**”, *Journal of Scheduling*, v. 20, n. 1, pp. 85–101, 2017. ISSN: 10946136. doi: 10.1007/s10951-016-0495-8.

KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. “**Optimization by Simulated Annealing**”, *Science*, v. 220, n. 4598, pp. 671–680, 1983. ISSN: 0036-8075. doi: 10.1126/science.220.4598.671.

KUMAR, R. S., ASOKAN, P., KUMANAN, S. “**Artificial immune system-based algorithm for the unidirectional loop layout problem in a flexible manufacturing system**”, *The International Journal of Advanced Manufacturing Technology*, v. 40, n. 5-6, pp. 553–565, 2009. doi: 10.1007/s00170-008-1375-y.

LÓPEZ-IBÁÑEZ, M., DUBOIS-LACOSTE, J., PÉREZ CÁCERES, L., et al.. “**The irace package: Iterated racing for automatic algorithm configuration**”, *Operations Research Perspectives*, v. 3, pp. 43–58, 2016. ISSN: 22147160. doi: 10.1016/j.orp.2016.09.002.

MICROSOFT. “**Overview of The .NET Framework**”. 2020.

PINEDO, M. L. *Scheduling - Theory, Algorithms, and Systems*. 4a ed. New York, 2012. doi: doi.org/10.1007/978-3-642-46773-8_5.

- SUN, D., TANG, L. “**Benders approach for the raw material transportation scheduling problem in steel industry**”, *IEEE International Conference on Control and Automation, ICCA*, pp. 481–484, 2013. ISSN: 19483449. doi: 10.1109/ICCA.2013.6565191.
- TAGUCHI, G. *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Quality Resources, 1986. ISBN: 9283310845.
- TRAN, T. T., ARAUJO, A., BECK, J. C. “**Decomposition Methods for the Parallel Machine Scheduling Problem with Setups**”, *INFORMS Journal on Computing*, v. 28, n. 1, pp. 83–95, 2016. doi: 10.1287/ijoc.2015.0666.
- VALE. “**Porto de Tubarão, em Vitória, completa 49 anos de história**”. 2015. Vale. Disponível em <http://www.vale.com/brasil/pt/aboutvale/news/paginas/porto-tubarao-vitoria-completa-49-anos-historia.aspx>. Acessado em 14/03/2021.
- ZHANG, M., SINGH, S. “**China 2020 iron ore imports hit record as infrastructure spending lifted demand**”. 2021. Reuters. Disponível em <https://cutt.ly/5kpHDrC>. Acessado em 01/02/2021.
- ÜNSAL, “**Reclaimer Scheduling in Dry Bulk Terminals**”, *IEEE Access*, v. 8, pp. 96294–96303, 2020. doi: 10.1109/ACCESS.2020.2997739.

Apêndice A: Publicações

A seguir é listada a publicação originada por este trabalho de pesquisa que foi aceita em um evento científico e publicada nos seus anais:

1. **Título:** UM ALGORITMO CONSTRUTIVO PARA O PROBLEMA DE SEQUENCIAMENTO DE RECUPERADORAS DE MINÉRIO

Autores: João Daniel Ferreira Jesus Junior, Luciano Perdigão Cota, Marcone Jamilson Freitas Souza

Evento: LII Simpósio Brasileiro de Pesquisa Operacional

Local: João Pessoa (online), PB, Brasil

Período: 3 a 5 de Novembro de 2020