

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

LUCAS CRISTIANO BERNARDONI

MODELAGEM DO CONTAINER STACKING PROBLEM UTILIZANDO UM MÉTODO
HEURÍSTICO

Joinville
2022

LUCAS CRISTIANO BERNARDONI

MODELAGEM DO CONTAINER STACKING PROBLEM UTILIZANDO UM MÉTODO
HEURÍSTICO

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientadora: Dra. Eng. Vanina
Macowski Durski Silva

Joinville
2022

AGRADECIMENTOS

Gostaria de agradecer primeiramente aos meus pais Sandro e Juliana por todo amor e ensinamentos que me proporcionaram, sendo a base de quem sou hoje. Também gostaria de agradecer aos meus irmãos Vitor e Artur pelos cuidados com um irmão mais novo.

À minha namorada Monique por todo o companheirismo durante a minha jornada, pelos momentos de tensão transformados em alegria e por muitas vezes ser o meu combustível para não desistir e seguir até o fim.

À professora Dra. Eng. Vanina Macowski Durski Silva por toda a orientação desde à iniciação científica até o presente trabalho, por acreditar no meu potencial sempre e, conseqüentemente, me ajudar a abrir portas inimagináveis na minha vida ao longo destes anos.

Aos meus amigos, membros do grupo Fort, pelos incontáveis finais de semana de risada e diversão que me ajudavam a lembrar como era bom estar em casa e, com isso, me motivar.

Aos meus amigos, membros da família TDO, pela criação de algo único e tão significativo durante a graduação e que perdurará pelo resto da vida, ajudamos uns aos outros a nos mantermos fortes até o fim.

Ao meu amigo Gabriel César Silveira por inúmeras disciplinas compartilhadas e aprovadas juntos, horas de conversas e disponibilidade em ouvir e opinar em momentos que precisei.

Aos meus gatos Luffy, Kim e Bella pela companhia, principalmente no período remoto, que contribuíram com o meu bem-estar durante períodos difíceis.

RESUMO

Este trabalho aborda um tema em evidência e ascensão na literatura: otimização de processos da cadeia logística. O problema específico tratado neste trabalho se refere à configuração (distribuição ou alocação) de contêineres em um terminal multimodal. A problemática se resume em contêineres alocados em um terminal de contêineres sem uma escolha adequada que considere as características dos contêineres e o estado do pátio em que são inseridos, implicando em realocações e deslocamento adicional dos equipamentos, atraso de conclusão de tarefas e mal aproveitamento do espaço físico, descrição que caracteriza o *Container Stacking Problem*. Desta forma, as consequências que caracterizam o problema em questão geram custos imprevistos e indesejados e a necessidade de evitar, ou ao menos reduzir, tais custos. Portanto, o objetivo deste trabalho é propor uma ferramenta que auxilie na determinação da posição ideal de um contêiner no terminal a fim de minimizar custo operacional da alocação do contêiner por meio do KPI de tempo total de manipulação (relacionado à distância total percorrida pelo equipamento). Neste trabalho, o terminal foi separado em seis diferentes níveis e com suas características, quatro heurísticas foram criadas para duas etapas, a escolha do contêiner no estoque e escolha de empilhamento em um bloco. Os resultados foram satisfatórios visto que foi possível definir os parâmetros que mais influenciam no processo de alocação de um contêiner.

Palavras-chave: Terminal de contêineres. Container Stacking Problem. Meta-Heurística.

ABSTRACT

The present work approaches a theme in evidence and rising in the literature: optimization of logistic chain processes. The specific problem addressed in this work refers to the configuration (distribution or allocation) of containers in a multimodal terminal. The problem boils down to containers allocated in a container terminal without an adequate choice that considers the characteristics of the containers and the state of the yard where they are inserted, implying in relocations and additional displacement of equipment, delay in completing tasks and poor use of the physical space, description that characterizes the *Container Stacking Problem*. In this way, the consequences that characterize the problem in question generate unforeseen and unwanted costs and the need to avoid, or at least reduce, such costs. Therefore, the objective of this work is to propose a tool that assists in determining the ideal position of a container in the terminal in order to minimize the operational cost of container allocation through the KPI of total handling time (related to the total distance traveled by the equipment). . In this work, the terminal was separated into six different levels and with its characteristics, four heuristics were created for two steps, the choice of the container in the stock and the choice of stacking in a block. The results were satisfactory since it was possible to define the parameters that most influence the container allocation process.

Keywords: Container Yard. Container Stacking Problem. Meta-Heuristics.

LISTA DE FIGURAS

Figura 1 – Valores Totais em Média Diária.	11
Figura 2 – Movimentação de contêineres no Brasil entre 2010 até 2030.	12
Figura 3 – Esquema de um pátio com seus subníveis.	15
Figura 4 – Blocos que recebem contêineres específicos.	18
Figura 5 – Pátios preenchidos gradualmente.	20
Figura 6 – Fluxograma das etapas do estudo.	22
Figura 7 – Subdivisão de cada etapa do fluxograma da Figura 6.	23
Figura 8 – Pátio com X e Y máximos iguais a 15 e 10 u.c, respectivamente. . .	29
Figura 9 – Criação de um bloco em um pátio.	30
Figura 10 – Criação do estoque de contêineres.	31
Figura 11 – Destaque das pilhas para o bloco da Figura 9.	32
Figura 12 – Demonstração de uma alocação de contêiner.	33
Figura 13 – Criação de um ativo e representação completa do pátio.	34
Figura 14 – Criação de um ativo e representação completa do pátio.	35
Figura 15 – Caminhos possíveis para alocar na posição desejada.	38
Figura 16 – Caminhos para ativos de diferentes alcances.	39
Figura 17 – Vista lateral para os casos da Figura 18.	40
Figura 18 – Ativos de alcances iguais a 2 frente à pilhas de diferentes alturas. .	40
Figura 19 – Ambiente de implementação conforme Código 5.	45
Figura 20 – Caminho do ativo para uma alocação de contêiner.	47
Figura 21 – Configuração do pátio após alocação da Figura 20.	48
Figura 22 – Comparação entre pátios após diferentes heurísticas aplicadas com 120 contêineres.	52
Figura 23 – Pátio após a aplicação da heurística 1.3.	52
Figura 24 – Pátio após a aplicação da heurística 2.2.	53
Figura 25 – Comparação entre pátios após diferentes heurísticas aplicadas. . .	54
Figura 26 – Comparação entre os pátios de menores custos.	55
Figura 27 – Distâncias totais de todas heurísticas para 120 contêineres.	56

LISTA DE QUADROS

Quadro 1 – Métodos e objetivos de estudos recentes sobre CSP.	25
Quadro 2 – Relação entre objetivos e KPIs	27
Quadro 3 – Quantidade de cada KPI nos estudos apresentados	28
Quadro 4 – Características de um contêiner.	36

LISTA DE TABELAS

Tabela 1 – Informações dos contêineres do Código 5.	46
Tabela 2 – Distância total percorrida para as heurísticas 1.1, 1.2 e 1.3.	51
Tabela 3 – Distância total percorrida para as heurísticas 2.1, 2.2 e 2.3.	53
Tabela 4 – Distância total percorrida para as heurísticas 3.1, 3.2 e 3.3.	54
Tabela 5 – Distância total percorrida para as heurísticas 4.1, 4.2 e 4.3.	55

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	13
1.2	Justificativa do trabalho	13
1.2.1	Econômica e Operacional	13
1.2.2	Acadêmica	14
1.3	Estrutura do trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Container Stacking Problem	15
2.1.1	Estático x Dinâmico	16
2.2	Regras de Empilhamento	16
2.2.1	Regras de Atribuição de Bloco	17
2.2.2	Regras de Atribuição de Baia	18
2.2.3	Regras de Atribuição de Pilha	19
2.3	KPIs	20
2.3.1	Espaço de Armazenamento	20
2.3.2	Processo de Alocação	21
3	MÉTODO	22
3.1	Objeto de Estudo	23
3.2	Revisão Bibliográfica e Definição do Método	23
3.3	Desenvolvimento do Modelo e Implementação	28
3.3.1	Divisão do terminal como ambiente de implementação	28
3.3.1.1	Pátio	28
3.3.1.2	Bloco	29
3.3.1.3	Estoque	30
3.3.1.4	Pilha	31
3.3.1.5	Contêiner	32
3.3.1.6	Rua	33
3.3.1.7	Ativo	34
3.3.2	Características dos componentes de implementação	35
3.3.2.1	Características do contêiner e bloco	35
3.3.2.2	Características do ativo para alocação	37
3.3.3	Definição do modelo e implementação	41

3.3.3.1	Variáveis de tomada de decisão	41
3.3.3.2	Cálculo do custo	42
3.3.3.3	Considerações e Restrições	43
3.3.3.4	Demonstração do modelo e implementação	44
3.4	Definição de heurísticas	49
4	ANÁLISE DE RESULTADOS	51
4.1	Relação entre custos e heurísticas	51
5	CONCLUSÕES	57
	REFERÊNCIAS	58

1 INTRODUÇÃO

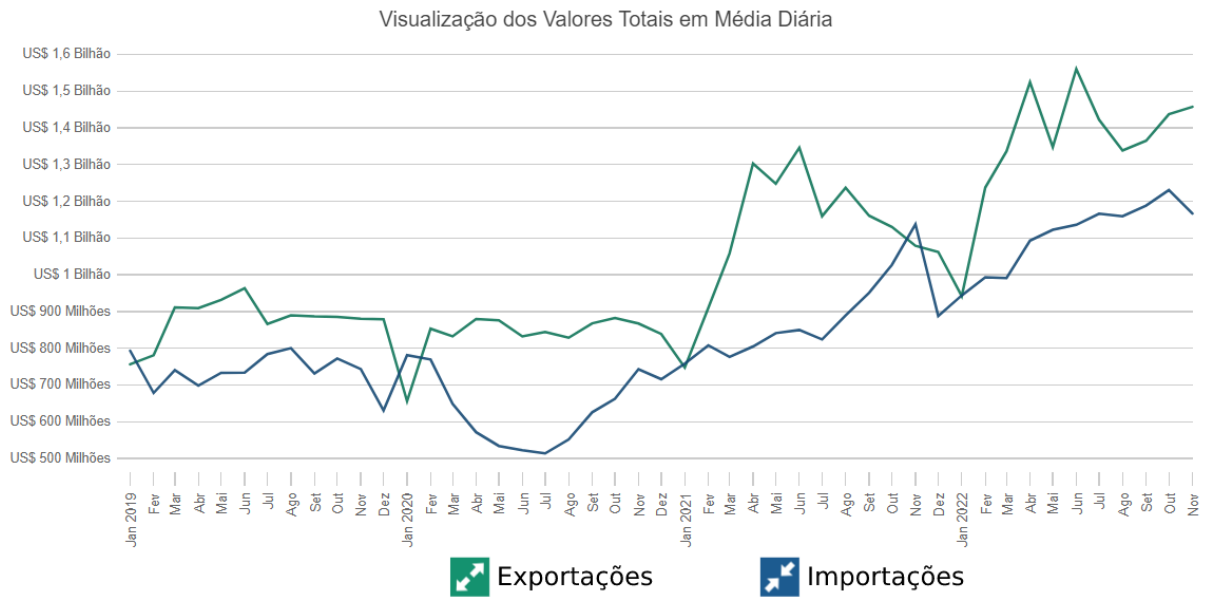
O constante desenvolvimento econômico-social global é reflexo da evolução tecnológica contínua. A busca por melhor qualidade de vida é evidente enquanto atividades cotidianas se tornam mais simplificadas, otimizadas ou até automatizadas. Entre tais atividades, a entrega de produtos adquiridos por *e-commerce* é um dos exemplos da facilidade e agilidade presente na atualidade. Neste sentido, surge a cadeia logística que estuda estes processos com a finalidade de definir abordagens para aprimorá-los ou reconhecer novos problemas que possam ser solucionados.

Em 2019, o valor total da logística global atingiu 6,6 trilhões de dólares com uma taxa de crescimento de 9,1% de acordo com Feng e Ye (2021). Já em 2021, o comércio global atingiu recorde de US\$ 28,5 trilhões um aumento de 25% em relação à 2020, a razão principal é o aumento de importações e exportações das maiores economias globais, aponta UNCTAD (2022). Logo, a movimentação dos produtos se torna de suma importância para efetivar os conceitos da cadeia logística e gerar uma economia saudável.

A movimentação de produtos através de navios e portos marítimos são responsáveis por aproximadamente 70% em valor e 80% em volume do comércio global, indica UNCTAD (2019). Evidentemente, a indústria marítima é o principal meio de transporte que apoia o comércio global e assim, devido ao rápido crescimento da carga containerizada, os terminais de contêineres marítimos tornaram-se os principais nós da rede de suprimentos.

Em 2021, o PIB do transporte no Brasil obteve um crescimento expressivo de 11,4% conforme CNT (2022), o que auxilia diretamente na movimentação dos produtos. A Figura 1 apresenta o valor total em dólares da média diária das importações e exportações no Brasil no período de janeiro de 2019 até novembro de 2022.

Figura 1 – Valores Totais em Média Diária.

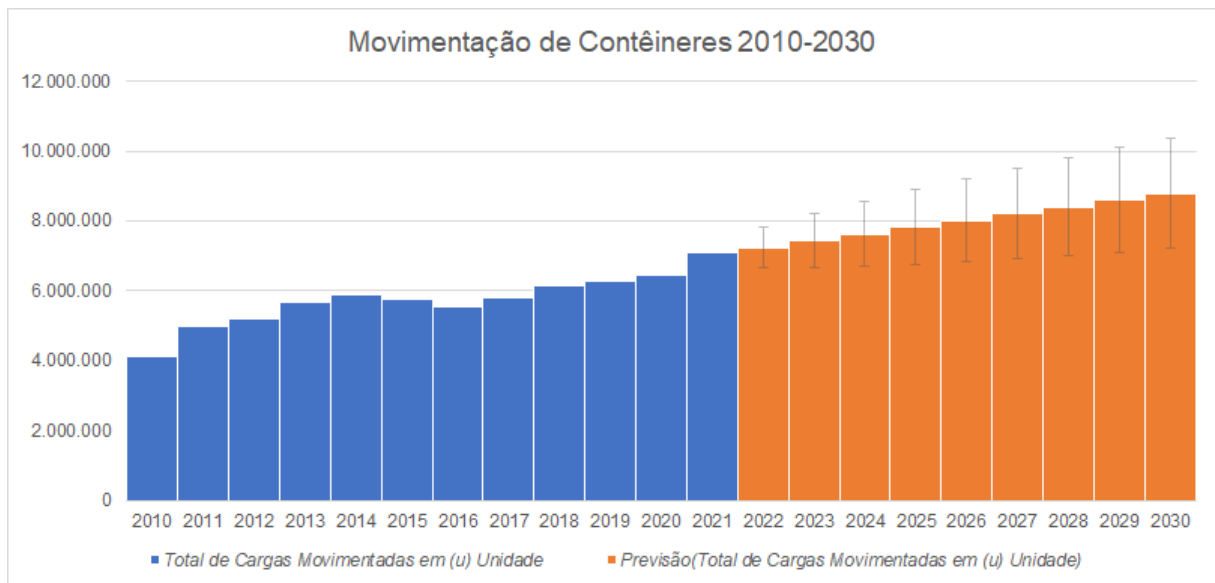


Fonte: Adaptado de govBR (2022).

O valor de exportações é superior ao de importações na maior parte do intervalo considerado e, na balança comercial, o superávit acontece quando as exportações excedem as importações, sendo fundamental para a economia de um país. Além disso, ao analisar as duas linhas individualmente, é perceptível o crescimento do valor total das exportações e importações ao longo dos anos causado pelo número cada vez maior de produtos movimentados.

Sabe-se que parte destes produtos são carregados em contêineres, e o aumento do transporte impacta no aumento da movimentação destes contêineres. Para verificar a evolução da movimentação dos contêineres no Brasil desde 2010 até 2021 e uma previsão de 2022 para 2030.

Figura 2 – Movimentação de contêineres no Brasil entre 2010 até 2030.



Fonte: Adaptado de ANTAQ (2022).

Entre 2010 e 2021, o crescimento constante é notável, com o auxílio da previsão este crescimento se torna ainda mais significativo com a quantidade total de cargas podendo chegar até 10 milhões. De 2010 a 2021, o total de cargas movimentadas quase dobrou de 4.134.685 para 7.088.134, resultando em um aumento de 71%.

Com grande representatividade na economia global, tanto os terminais portuários como os rodoviários e ferroviários exigem controle operacional e logístico para garantir um pátio de contêineres seguro, organizado e eficiente. Sem este controle, o terminal tende à realocações de contêineres desnecessárias, maior distância viajada pelos equipamentos (como por exemplo por reach stackers) e atraso na entrega e saída de contêineres. Tais consequências elevam os custos operacionais, e neste sentido tem-se o Problema de Empilhamento de Contêiner, ou do inglês, Container Stacking Problem (CSP), cujo seu estudo busca determinar as localizações ideais em um dado pátio para os contêineres recém chegados.

Portanto, considerando o CSP, este trabalho busca responder a seguinte questão: quais considerações influenciam na escolha de decisão da posição do contêiner no pátio e, a partir da comparação de diferentes métodos, qual algoritmo mais minimiza o custo operacional total para tal tomada de decisão?

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um algoritmo capaz de determinar a posição em que será alocado cada contêiner descarregado em um pátio de um terminal (portuário, ferroviário ou

rodoviário), a fim de minimizar o custo operacional por meio do KPI de tempo total de manipulação (relacionado à distância total percorrida pelo equipamento).

1.1.2 Objetivos Específicos

- Pesquisar sobre os modelos existentes para o Container Stacking Problem;
- Propor níveis de endereço padrão para um pátio de terminal;
- Definir parâmetros de composição do custo operacional do empilhamento/movimentação de contêineres
- Comparar o custo operacional de diferentes métodos de alocação de contêineres;
- Identificar o método mais adequado a fim de minimizar o custos total de operação.

1.2 JUSTIFICATIVA DO TRABALHO

1.2.1 Econômica e Operacional

Devido à quantidade significativa do crescimento da containerização explicada no início do Capítulo 1, a necessidade de uma melhor tomada de decisão para alocação de contêineres no pátio é evidente. O esforço para alocar um contêiner exige diversos recursos que podem ser altamente custosos, como: hora máquina do equipamento a ser utilizado, combustível e tempo de operação. Portanto, existe uma responsabilidade econômica a cada contêiner alocado a um terminal. Como o intuito deste trabalho é definir um modelo capaz de reduzir os custos da operação de alocação de um contêiner em um terminal, o interesse econômico compõe um dos pilares deste estudo.

Com a implementação de diferentes métodos para alocação de contêineres em um terminal, este trabalho propõe adicionalmente a geração de um relatório de alocação que relacione os custos, espaço aproveitado e premissas consideradas; desta forma, o usuário é capaz de analisar se a abordagem que emprega para alocar os contêineres no terminal condiz com seu interesse ou se existem possíveis melhorias a serem implementadas.

A importância econômica está diretamente relacionada ao processo operacional do terminal logístico, porém, a abordagem operacional possui especificidades enquanto a econômica tem como objetivo único reduzir o custo global de operação logística. A questão operacional é abordada neste trabalho a partir da proposição de uma ferramenta computacional (em um aplicativo/software) onde é possível que o usuário visualize na tela diferentes opções para alocar os contêineres de acordo com as preferências escolhidas. Por exemplo, se o terminal não permitir que contêineres de clientes diferentes se misturem na mesma pilha, basta acessar a opção “Clientes” e selecionar quais os clientes permitidos para uma área específica. Logo, essa ferramenta expande as possibilidades de alocação para diversos terminais ou até mesmo para diferentes situações de um mesmo terminal. Caso se deseje verificar

como seria a alocação dos contêineres para diferentes permissões, basta alterar as opções na tela mencionada e configurar conforme necessário. Tais alterações oferecem flexibilidade e customização para os dados de entrada e, conseqüentemente, de saída, permitindo saber o custo da operação dadas as especificidades adotadas.

1.2.2 Acadêmica

A interdisciplinaridade é uma das principais maneiras de firmar o conhecimento de diferentes áreas. Saber aplicar conceitos distintos de forma conjunta e com êxito garante um domínio dos temas. Por este motivo, este trabalho de conclusão de curso em Engenharia Mecatrônica busca utilizar conceitos de Programação, Pesquisa Operacional e Logística, ao propor uma ferramenta para o Container Stacking Problem.

Primeiramente, foi necessário compreender os conceitos que envolvem Logística para que, desta forma, fosse possível abordar a problemática deste trabalho com embasamento bibliográfico e uma visão geral dos processos necessários para propor uma solução compatível com cenários reais.

Posteriormente, a Pesquisa Operacional introduziu o trabalho para sua parte principal. É responsável por elencar os aspectos do problema de maneira palpável com o propósito de auxiliar na tomada de melhores decisões.

Por fim, a partir da linguagem Python e das bibliotecas *numpy* e *matplotlib*, o espaço que simula um terminal de contêineres foi criado. Desta forma, foi possível unir o ambiente virtual bidimensional, as variáveis de entrada e as heurísticas criadas, gerando uma análise das distâncias totais percorridas.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 6 seções incluindo esta introdutória. Primeiramente, o trabalho é contextualizado a partir da sua importância nos dias de hoje, sua problemática, os objetivos que desejam ser alcançados e seus motivos. Na segunda seção, os mais recentes e principais estudos sobre o Container Stacking Problem são abordados, comparando os principais parâmetros de alocação de contêiner considerados nos modelos bem como o método de solução e algoritmo utilizado para a resolução do problema. Já na terceira seção, é descrito o método utilizado para o desenvolvimento deste trabalho para garantir que haja a possibilidade de replicação a fim de validar o estudo. A quinta seção aborda os resultados gerados a partir dos métodos da seção interior, comparando diferentes métodos para a alocação dos contêineres do pátio. Por fim, os resultados são discutidos e analisados com o intuito de definir qual o melhor método, os parâmetros mais impactantes e possíveis melhorias.

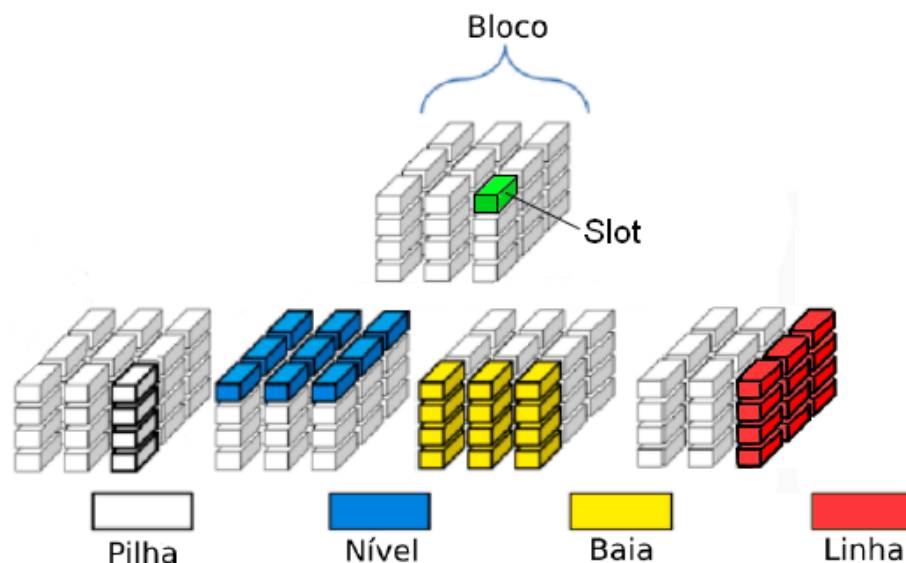
2 FUNDAMENTAÇÃO TEÓRICA

2.1 CONTAINER STACKING PROBLEM

O CSP é responsável por determinar a posição ideal de alocação de contêineres que chegam a um determinado pátio de acordo com os espaços disponíveis enquanto considera o carregamento eficiente e operações de entrega em um terminal de contêineres. Em outras palavras, determinar a localização para alocar contêineres para operações de retirada futuras minimizando as realocações Gunawardhana, Perera e Thibbotuwawa (2021).

De acordo com a Figura 3, um pátio consiste em várias áreas retangulares demarcadas como blocos de pátio. Cada bloco de pátio compreende várias baias; cada baia compreende várias linhas e cada linha compreende vários níveis verticais. A localização final de um contêiner é o slot, e ele representa um código que inclui detalhes do bloco, baia, linha e nível específicos Ting e Wu (2017).

Figura 3 – Esquema de um pátio com seus subníveis.



Fonte: Adaptado de Rong et al. (2020).

Para Lersteau et al. (2021), o CSP com o objetivo de minimizar o número de itens bloqueados de acordo com a ordem de chegada (e saída) enquanto considera a redução de consumo de energia e garantindo uma diminuição no número de violações de empilhamento (alocações que geram realocações) de 62,6% para 0,9%. A redução expressiva demonstra o impacto que o CSP pode ter em um terminal de contêineres.

Para Wang et al. (2022) o problema de alocação 3D em pátios com dimensão

de tempo (3DYAPT), busca determinar os locais de alocação de contêineres em um determinado bloco de armazenamento para satisfazer os requisitos de lotes de contêineres recebidos. O objetivo é minimizar a área bidimensional do bloco de armazenamento ocupada temporariamente por contêineres dentro de um determinado horizonte de planejamento (dimensão do tempo).

Também considerando a dimensão temporal, WANG (2021) a divide em duas partes. Em primeiro lugar, as visitas aos portos de diferentes embarcações têm padrões diferentes em termos de períodos de tempo, o que adiciona restrições complexas ao alocar espaços de armazenamento em diferentes períodos. Em segundo lugar, os contêineres transportados pela mesma embarcação sempre chegam ao pátio durante um intervalo de tempo.

Já Oelschlägel e Knust (2021) assumem que os itens (contêineres) chegam em uma determinada sequência em uma área de armazenamento parcialmente preenchida e devem ser atribuídos a pilhas sujeitas à restrição de que nem todos os itens podem ser empilhados em cima de todos os outros itens. Seu estudo tem como objetivos minimizar o número de pilhas usadas e o número de itens mal alocados em relação a tempos de retirada, respectivamente.

A busca por solução ao CSP é feita com base em várias características em relação aos contêineres (tamanho, tipo, porto de destino, etc.) e pátio de empilhamento (capacidade, alturas de empilhamento, distâncias de segurança, etc.) Rekik, Elkosantini e Chabchoub (2015a). Além disso, os mesmos autores também afirmam que o CSP foi desenvolvido com base em várias regras de empilhamento de contêineres (estratégias), indicadores-chave de desempenho (KPIs) e diferentes restrições de empilhamento.

2.1.1 Estático x Dinâmico

O CSP pode ser tratado de acordo com o momento em que as informações dos contêineres a serem alocados são recebidas assim como o momento em que a decisão de qual posição alocá-los é tomada originando o Static Container Stacking Problem (SCSP) e Dynamic Container Stacking Problem (DCSP) Rekik e Elkosantini (2019).

2.2 REGRAS DE EMPILHAMENTO

Para melhorar o desempenho dos terminais, várias regras de empilhamento (estratégias) têm sido desenvolvidas na literatura Ma e Kim (2012). A eficiência de cada regra varia de terminal para terminal Asperen, Borgman e Dekker (2013). De acordo com Ma e Kim (2012), uma regra de empilhamento pode estar relacionada à seleção de um bloco, uma baía ou uma pilha. Assim, categorizam-se as existentes regras de

empilhamento em três famílias principais: Regras de Atribuição de Blocos, Regras de Atribuição de baias e Regras de Atribuição de Pilha Rekik e Elkosantini (2019).

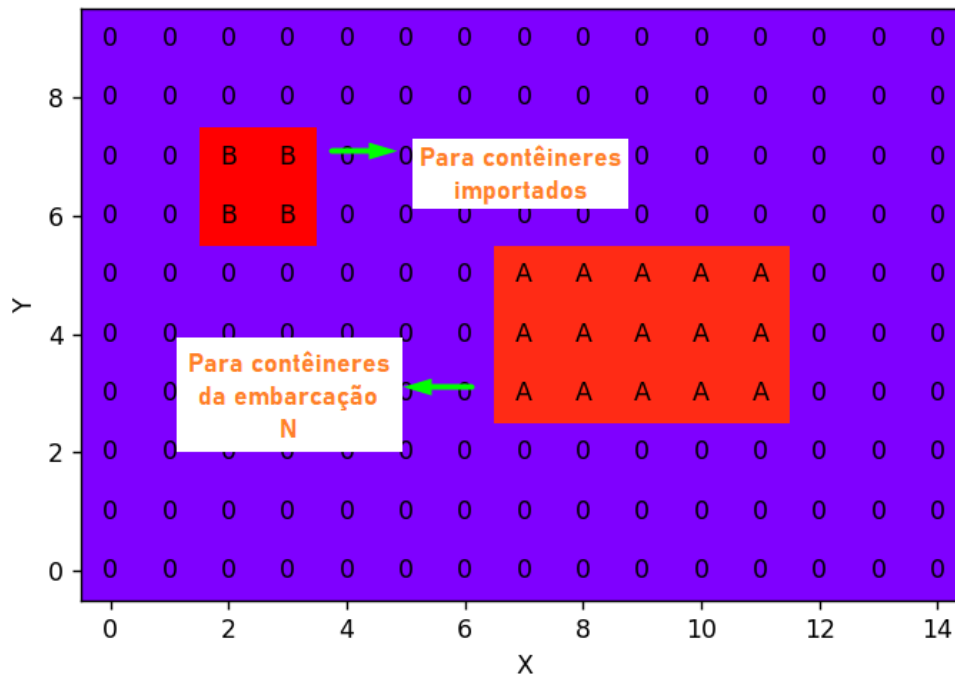
2.2.1 Regras de Atribuição de Bloco

As Regras de Atribuição de Blocos (estratégias) estão relacionadas à seleção do bloco “apropriado” para contêineres de entrada/saída. Essas regras incluem:

- **Áreas Dedicadas:** consiste em reservar um bloco específico para tipos específicos de contêineres, como bloco para contêineres importados ou um bloco para alguns tipos de contêineres de uma embarcação específica Chen e Lu (2012).
- **Sem Restrição:** o bloco recebe quaisquer tipos de contêineres de quaisquer embarcações.
- **Diferentes Prioridades em Blocos para Diferentes Berços de Atracação:** fixar contêineres importados próximo ao portão de chegada e contêineres que serão exportados próximo ao berço de atracação que serão carregados Ma e Kim (2012).
- **Número Máximo de Caminhões Internos e Rodoviários em um Bloco:** seleciona blocos onde o número de caminhões em espera não exceda um limite fixo Ma e Kim (2012).
- **Lógica Fuzzy:** seleciona o bloco com o maior valor de interesse que é determinado para cada bloco viável de acordo com um conjunto de critérios Ries, González-Ramírez e Miranda (2014).

A Figura 4 apresenta a regra de Áreas Dedicadas visto que será a regra empregada neste trabalho. A representação do pátio abaixo é a mesma utilizada durante todo o trabalho e será explicada com detalhes no próximo capítulo.

Figura 4 – Blocos que recebem contêineres específicos.



Fonte: Autor (2022).

Os eixos x e y representam as dimensões do pátio enquanto os espaços em vermelho representam tais áreas ou blocos específicos para diferentes tipos de contêineres. Na Figura 4, o bloco "B" consiste em receber contêineres importados enquanto o bloco "A" em receber contêineres de uma embarcação específica.

2.2.2 Regras de Atribuição de Baia

As Regras de Atribuição de Baias tratam da seleção da baia no bloco pré-selecionado pela regra anterior. Várias regras de empilhamento de baias foram estudadas na literatura, como:

- **Localização Concentrada:** consiste em reservar alguma área de armazenamento para um determinado grupo de contêineres e atribuindo contêineres do mesmo grupo a baias adequadas Ma e Kim (2012) Chen e Lu (2012).
- **Localização Mais Próxima:** designar contêineres recém chegados para a baia mais próxima do berço de embarcação e também contêineres que sairão para a baia mais próxima do portão Woo e Kim (2011).
- **Regra de Sequência:** selecionar baias vazias para contêineres recém chegados Ma e Kim (2012).
- **Lógica Fuzzy:** mesma lógica empregada para a Lógica Fuzzy para Regras de

Atribuição de Bloco Ries, González-Ramírez e Miranda (2014).

Para este trabalho, não será implementado a regra de atribuição de Baia, utilizar-se-á somente a regra de atribuição de bloco e a regra de atribuição de pilha, descrita a seguir.

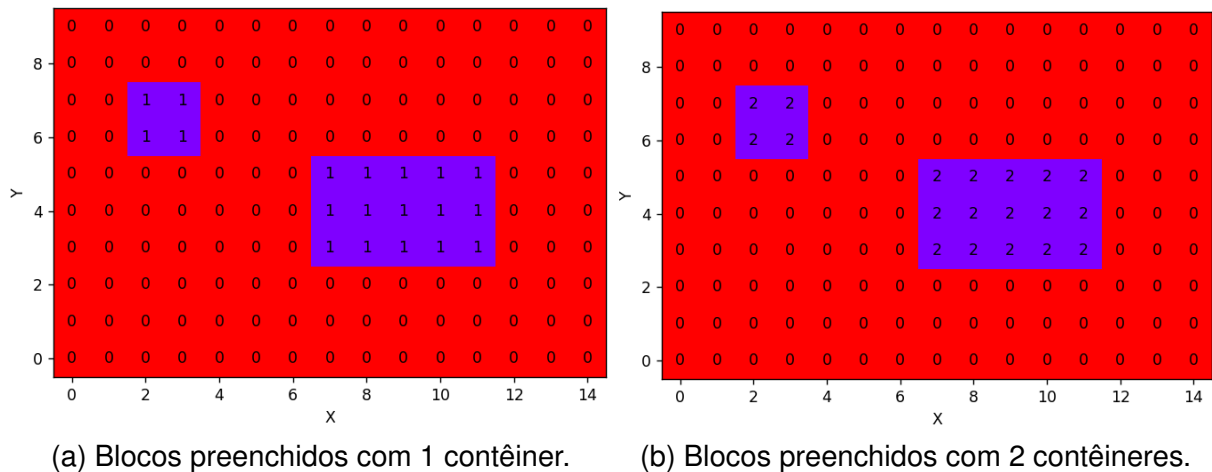
2.2.3 Regras de Atribuição de Pilha

As regras de Atribuição de Pilha são responsáveis por determinar o local de alocação exato na baia pré-selecionada (pela aplicação das Regras de Atribuição de baia) e no bloco pré-selecionado (pela aplicação das Regras de Atribuição de Bloco). Algumas das regras de atribuição de pilha:

- Empilhamento Aleatório: uma pilha aleatória da baia é selecionada até que a altura máxima seja atingida AZIZ e ZAKARIA (2003).
- Nivelamento: a pilha é preenchida camada por camada. Todas as posições do chão vazias são preenchidas com contêineres primeiro, antes que os contêineres sejam empilhados sobre outros Duinkerken, Evers e Ottjes (2001).
- Segregação: evitar o empilhamento de contêineres de entrada em cima de contêineres que já estão empilhados Huynh (2008).
- Altura Máxima da Pilha Restante: colocar o contêiner em pilhas com os níveis mais altos Rekik, Elkosantini e Chabchoub (2015b).
- Lógica Fuzzy: mesma lógica empregada para a Lógica Fuzzy para Regras de Atribuição de Bloco e Regras de Atribuição de Baia Ries, González-Ramírez e Miranda (2014).

A Figura 5 apresenta a regra de Nivelamento com a representação do pátio empregada neste trabalho.

Figura 5 – Pátios preenchidos gradualmente.



Fonte: Autor (2022)

Primeiramente, nota-se que as figuras diferem no valor apresentado em cada bloco e este valor representa o número de contêineres presentes em determinada posição. Na Figura 5a, os dois blocos foram completamente preenchidos com apenas 1 contêiner (todas posições nos blocos com valores iguais a 1), seguindo a regra de Nivelamento. A Figura 5b representa mais uma etapa da regra de Nivelamento, preenchendo novamente todas as posições dos blocos com mais uma camada de contêineres, resultando em 2 contêineres para todo o bloco.

Pela regra de Nivelamento não é permitido que duas pilhas tenham altura com diferença maior que 1 pois significa que uma pilha recebeu algum contêiner antes que toda camada fosse preenchida.

Para este trabalho, a regra de atribuição de pilha empregada será explicada ao final do capítulo 3.

2.3 KPIS

Key Performance Indicators (KPIs) são ferramentas empregadas para medir o nível de desempenho de um processo. Dentro do contexto de terminais de contêineres, dois principais tipos definem estes KPIs: Espaço de Armazenamento e Processo de Alocação Rekik, Elkosantini e Chabchoub (2015b).

2.3.1 Espaço de Armazenamento

O alto fluxo de contêineres em um terminal exige que o espaço seja bem aproveitado e garanta a possibilidade de alocação de todos contêineres no espaço disponível. Por este motivo, parte dos estudos focam em gerenciar o terminal para que o espaço ocupado seja o menor possível enquanto respeita restrições, como por exemplo, a data de chegada e saída do contêiner. WANG et al. (2022) propõem a minimização

do espaço 3D ocupado pelos contêineres considerando o tempo. Já Oelschlägel e Knust (2021) buscam minimizar o número de pilhas usadas e o número de itens mal alocados.

2.3.2 Processo de Alocação

Além de um espaço bem aproveitado, o terminal de contêineres exige que o desempenho operacional se mantenha adequado. É importante evitar quaisquer situações que possam se tornar custosas ao terminal, como: operações não planejadas, rotas maiores que o esperado, desperdício de energia e atraso na manipulação de contêineres.

Para garantir que isto aconteça, Gunawardhana, Perera e Thibbotuwawa (2021) e Rong et al. (2020) tem como objetivo minimizar a distância percorrida do guindaste a fim de reduzir os custos e a energia, respectivamente. Além de obter este mesmo objetivo, Li, Xiaoning e Zhengyu (2019) também buscam minimizar a quantidade de contêineres sobrepostos. Isto é, como pilhas são construídas na ordem FILO (First In, Last Out - primeiro a chegar, último a sair), evita-se que os primeiros contêineres inseridos (últimos a sair) necessitem ser retirados da pilha antes dos contêineres que estão no topo.

Uma vez que o contêiner de menor data de saída é alocado como primeiro da pilha, todos contêineres da pilha serão retirados para que este possa ser manipulado no momento da coleta. Desta forma, o problema da sobreposição está diretamente vinculado ao número de realocações necessárias ao inserir ou retirar um contêiner de uma pilha. Maglić, Gulić e Maglić (2019) buscam minimizar o número realocações de contêineres enquanto Lersteau et al. (2021) possui um objetivo parecido porém tratado de maneira diferente: minimizar o número de itens bloqueados (contêineres que devem ser retirados do terminal mas não estão diretamente acessíveis), sendo a redução de realocações uma consequência e não o objetivo principal.

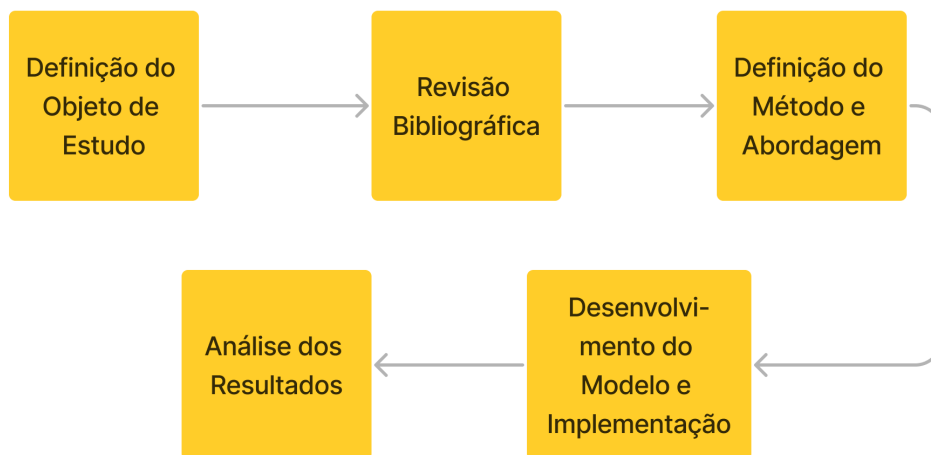
Maglić, Gulić e Maglić (2019) consideram um espaço bidimensional e estático. Já Gunawardhana, Perera e Thibbotuwawa (2021) e Lersteau et al. (2021) consideram o sistema dinâmico, isto é, como há uma maior incerteza em um terminal de contêineres de sistema em tempo real, há uma probabilidade considerável de eventos inesperados.

Como o objetivo deste trabalho é minimizar a distância percorrida pelo equipamento responsável pela alocação do contêiner, o KPI de Tempo de Manipulação do tipo Processo de Alocação será empregado como o principal indicador de desempenho nas soluções propostas.

3 MÉTODO

Para realizar este trabalho, o modelo acadêmico foi seguido e cada etapa foi subdividida em processos menores. O fluxograma representado na Figura 6 descreve as etapas seguidas para construir este trabalho.

Figura 6 – Fluxograma das etapas do estudo.



Fonte: Autor (2022).

Cada bloco representa um capítulo deste trabalho, o Capítulo 1 é representado por Definição do Objeto de Estudo, o Capítulo 2 por Revisão Bibliográfica, Definição do Método e Abordagem finaliza o Capítulo 2 e inicia o Capítulo 3, Desenvolvimento do Modelo e Implementação conclui o Capítulo 3 e, por fim, o Capítulo 4 é representado por Análise dos Resultados. Todos estes blocos são subdivididos como é apresentado na Figura 7.

Figura 7 – Subdivisão de cada etapa do fluxograma da Figura 6.



Fonte: Autor (2022)

3.1 OBJETO DE ESTUDO

Com o intuito de estudar os fatores que influenciam o Container Stacking Problem em terminais multimodais e propor uma análise que relaciona custos operacionais e heurísticas de alocação, os objetivos e as justificativas deste trabalho são apresentados na Seção 1.1 e 1.2.

3.2 REVISÃO BIBLIOGRÁFICA E DEFINIÇÃO DO MÉTODO

O Capítulo 2 foi responsável por apresentar a problemática tratada neste trabalho e conceitos que tangenciam a mesma. Com o conhecimento e entendimento destes conceitos, foi possível estudar conteúdos recentes e extrair informações úteis para serem implementados neste trabalho. Por este motivo, a seção atual combina a Revisão Bibliográfica e a Definição do Método com o intuito de relacionar tais informações com o interesse do autor para definir as regras e método a serem utilizados.

De acordo com a definição do Container Stacking Problem no Capítulo 2, o problema pode ser estático ou dinâmico. Considerá-lo estático significa que as informações são conhecidas previamente e não são alteradas durante o processo

de alocação. A abordagem dinâmica desconhece as informações previamente e permite que propriedades de blocos ou contêineres sejam alteradas em tempo real. Neste trabalho, considerar-se-á o SCSP (Static Container Stacking Problem) Rekik e Elkosantini (2019) visto que os contêineres a serem alocados já são conhecidos assim como todos os elementos que pertencem ao pátio em que os contêineres serão alocados.

Em seguida, as Regras de Empilhamento foram apresentadas e se segmentam em três partes: Bloco, Baia e Pilha. Dentre as Regras de Atribuição de Bloco apresentadas, escolheu-se a regra de Áreas Dedicadas Chen e Lu (2012) para empregar neste trabalho. Já a Regra de Atribuição de Baia não será utilizada. Por fim, a Regra de Atribuição de Pilha empregada foi desenvolvida pelo autor e será explicada no decorrer deste trabalho.

O último tópico sobre CSP apresentado descreve o desempenho da alocação dos contêineres e é dividido em dois subtópicos: Espaço de Armazenamento e Processos de Alocação Rekik, Elkosantini e Chabchoub (2015b). O primeiro trata o espaço físico ocupado pelos contêineres alocados, minimizando o espaço bidimensional ou tridimensional. Já os Processos de Alocação se referem à realocações e tempo de manipulação do contêiner.

Para o presente trabalho, o principal KPI é o tempo de manipulação do contêiner que é contabilizado pela distância percorrida pela *reach stacker* ao alocar os contêineres. Adicionalmente, os modelos que serão implementados neste trabalho também relacionam a taxa de ocupação do pátio para avaliar a qualidade do método de alocação.

Uma vez que as estratégias referentes ao CSP foram escolhidas, o Quadro 1 foi construído a fim de auxiliar no embasamento do modelo de acordo a bibliografia recente. São apresentados quais os objetivos finais dos estudos recentes sobre CSP assim como as condições que representam restrições ou premissas consideradas e os métodos empregados para alcançar tais objetivos.

Quadro 1 – Métodos e objetivos de estudos recentes sobre CSP.

Autores	Ano	Método	Condição	Objetivo
Taekwang Kim, Kwang Ryel Ryu	2022	algoritmo genético tolerante a ruído	Carga de trabalho atual da operação da embarcação	Minimizar o tempo total de movimentação de contêineres
Yuanjun Feng, Dong-Ping Song, Dong Li	2022	Mixed-Integer Programming (MIP)	Determinar os clientes/contêineres inteligentes e o número e localizações das pilhas inteligentes	Minimizar o tempo total de recuperação
Tiantian Wang, Hong Ma, Zhou Xu, Jun Xia	2021	simulated annealing-based dynamic shape adjustment and placement algorithm (SA-DSAP)	Intervalo de tempo específico	Minimizar o espaço 2D ocupado pelos contêineres
Janith A. Gunawardhana, H. Niles Perera, Amila Thibbotuwawa	2021	Rule-based systems (AI)	Tempo real e redução de custos	Minimizar a distância de movimentação do guindaste (ativo)
Charly Lersteau, Trung Thanh Nguyen, Tri Thanh Le, Ha Nam Nguyen, Weiming Shen	2021	Método iterativo com duas etapas	Redução do esforço na descarga	Minimizar o número de itens bloqueados de acordo com a ordem de chegada (e saída)
Jia Rong, Shouwen Ji, Qianru Zhao and Dong Guo	2020	algoritmo genético	Redução de consumo de energia	Minimizar a distância de movimentação do guindaste (ativo)
Tobias Oelschlägel, Sigrid Knust	2020	Mixed-Integer Programming (MIP)	Data de saída	Minimizar o número de pilhas usadas e itens mal alocados
Wang Li, Zhu Xiaoning, Xie Zhengyu	2019	simulation-based algorithm	Preferências de alocação	Minimizar a quantidade de sobreposição e a distância de movimentação do guindaste (ativo)
Livia Maglic, Marko Gulic, Lovro Maglic	2019	algoritmo genético	Bidimensional, estático e offline	Minimizar o número de realocações de contêineres dentro do pátio

Fonte: Autor (2022)

A coluna condição descreve premissas ou restrições que devem ser respeitadas ao mesmo tempo que se busca atingir o objetivo, abordam se as informações para alocação são recebidas em tempo real ou offline e quais tipos de redução devem ser respeitadas (custos, energia, esforço). Por exemplo, Feng (2022) busca minimizar o tempo total de recuperação (tempo necessário para retirar um contêiner do terminal) enquanto determina os clientes/contêineres inteligentes e o número de localizações das pilhas inteligentes.

Em relação ao objetivo, os três principais apresentados são: distância, espaço e realocações. A minimização de "itens mal alocados" ou "itens bloqueados" pode representar tanto a redução do espaço físico com os contêineres alocados como o número de realocações necessárias para alocar/retirar um contêiner, variando de acordo com o objetivo de cada autor. A fim de relacionar os objetivos diretamente com os KPIs apresentados na Seção 2.3, é apresentado o Quadro 2.

Quadro 2 – Relação entre objetivos e KPIs

Autores	Ano	Objetivo	KPIs
Taekwang Kim, Kwang Ryel Ryu	2022	Minimizar o tempo total de movimentação de contêineres	Tempo de Manipulação
Yuanjun Feng, Dong-Ping Song , Dong Li	2022	Minimizar o tempo total de recuperação	Tempo de Manipulação
Tiantian Wang, Hong Ma, Zhou Xu, Jun Xia	2021	Minimizar o espaço 2D ocupado pelos contêineres	Capacidade de Empilhamento
Janith A. Gunawardhana, H. Niles Perera, Amila Thibbotuwawa	2021	Minimizar a distância de movimentação do guindaste (ativo)	Tempo de Manipulação
Charly Lersteau, Trung Thanh Nguyen, Tri Thanh Le, Ha Nam Nguyen, Weiming Shen	2021	Minimizar o número de itens bloqueados de acordo com a ordem de chegada (e saída)	Realocações
Jia Rong, Shouwen Ji, Qianru Zhao and Dong Guo	2020	Minimizar a distância de movimentação do guindaste (ativo)	Tempo de Manipulação
Tobias Oelschlägel, Sigrid Knust	2020	Minimizar o número de pilhas usadas e itens mal alocados	Realocações e Capacidade de Empilhamento
Wang Li, Zhu Xiaoning, Xie Zhengyu	2019	Minimizar a quantidade de sobreposição e a distância de movimentação do guindaste (ativo)	Tempo de Manipulação
Livia Maglic, Marko Gulic, Lovro Maglic	2019	Minimizar o número de realocações de contêineres dentro do pátio	Realocações

Fonte: Autor (2022)

O Quadro 2 auxilia a visualizar quais os KPIs mais empregados nos estudos consultados para este trabalho e a relação com seus objetivos. É possível observar que alguns estudos possuem mais de um KPI de acordo com seus objetivos. Para quantificar o uso de cada KPI de forma mais clara, o Quadro 3 foi construído.

Quadro 3 – Quantidade de cada KPI nos estudos apresentados

KPIs	Objetivos e Quantidades
Processo de Alocação	Realocações (4) e Tempo de Manipulação (5)
Espaço de Armazenamento	Capacidade de Empilhamento (2)

Fonte: Autor (2022)

A quantidade total de KPIs presentes (11) é maior que o número dos estudos apresentados (9) visto que alguns estudos possuem mais de um KPI dentro dos seus objetivos. Nota-se que os KPIs relacionados ao Processo de Alocação possuem um emprego significativamente maior que os relacionados ao Espaço de Armazenamento. Como comentado, o Tempo de Alocação é o principal KPI deste trabalho e é justificado por o objetivo ser minimizar a distância total percorrida e este impactar diretamente no tempo de alocação.

3.3 DESENVOLVIMENTO DO MODELO E IMPLEMENTAÇÃO

A partir do estudo e análise das regras, estratégias, considerações e métodos empregados nos estudos recentes sobre CSP, tais características foram selecionadas para o modelo deste trabalho conforme visto na seção anterior. Portanto, a seção atual apresenta o desenvolvimento do modelo e implementação baseado em tais informações e segue a ordem lógica do fluxograma apresentado na Figura 7.

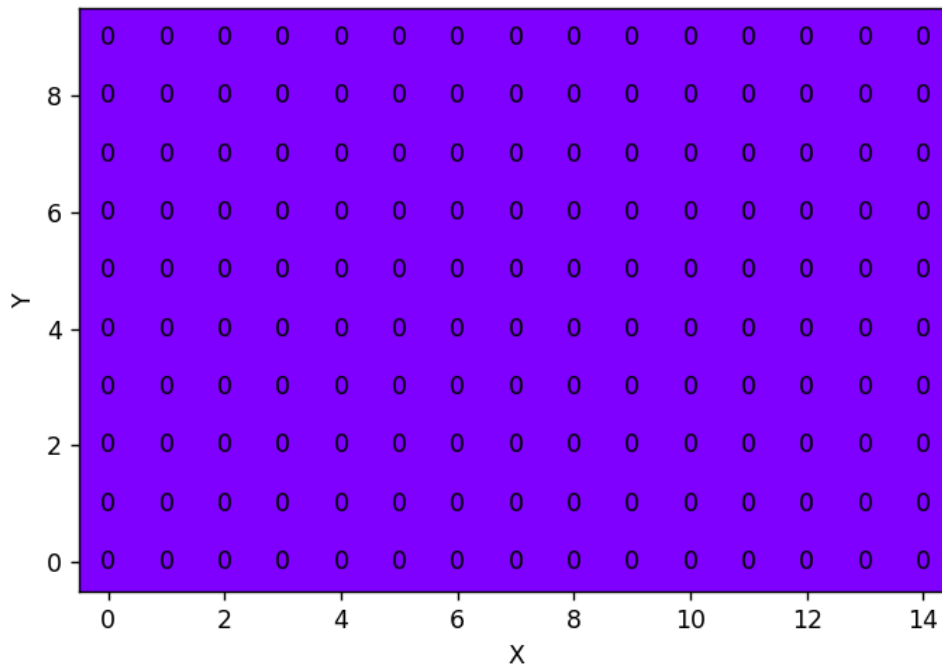
3.3.1 Divisão do terminal como ambiente de implementação

O primeiro passo para construir o modelo e implementação deste trabalho foi subdividir o terminal de contêineres em níveis a fim de compreender a hierarquia dos processos necessários para realizar a alocação de um contêiner e definir características inerentes às áreas abordadas em tais processos. Portanto, a estrutura do terminal segue a ordem: pátio, blocos, pilhas, estoque, contêineres, rua e ativos (equipamento responsável pela movimentação dos contêineres).

3.3.1.1 Pátio

Diferentemente de Ting e Wu (2017), que utiliza o sistema baia-linha-nível para definir a posição (slot) de um contêiner no pátio, utilizar-se-á o sistemas de coordenadas (x, y, z) neste trabalho. Desta forma, o pátio será representado por vistas superiores, sendo as divergentes explicitadas. Portanto, para a criação de um pátio, seus parâmetros de entrada são X e Y máximos que definem suas dimensões em unidades de comprimento (u.c). A Figura 8 apresenta a representação do pátio.

Figura 8 – Pátio com X e Y máximos iguais a 15 e 10 u.c, respectivamente.



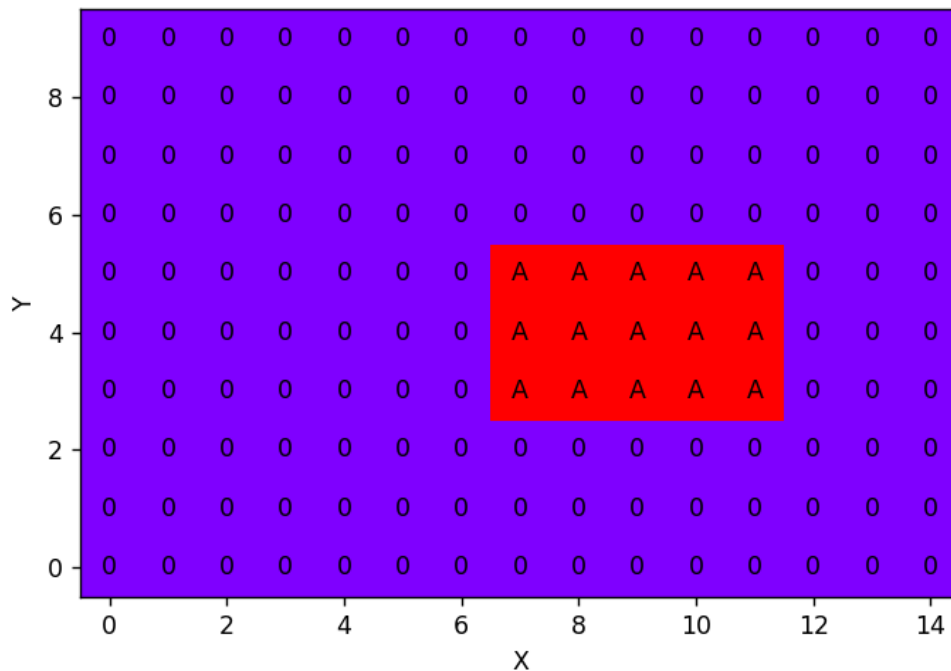
Fonte: Autor (2022)

No eixo das abcissas, que é legendado como 'X', os valores variam de 0 até 14 totalizando 15, valor de X máximo e resultando em 15 colunas. Já no eixo das ordenadas, que é legendado como 'Y', variam de 0 até 9 totalizando 10, valor de Y máximo e resultando em 10 linhas. Portanto, as posições deste pátio variam entre 0 até 14 para x e 0 até 9 para y . Vale ressaltar que inicialmente não há nenhum outro elemento inserido no pátio, os valores 0 representam tal ausência.

3.3.1.2 Bloco

O segundo nível definido pela cadeia é o bloco, o espaço designado para a alocação dos contêineres que também possui a entrada dos valores X e Y máximos para definir suas dimensões. Ademais, necessita-se de um caractere para demonstração visual e identificação, uma posição de origem com coordenadas (x, y) que define a partir de onde o bloco será criado, por exemplo: um bloco criado na posição $(1, 0)$ e com X máximo igual a 3 terá três colunas, a primeira em $(1, 0)$ e a última em $(3, 0)$. E, por fim, também exige uma lista de características que serão explicadas posteriormente nesta seção. A Figura 9 apresenta o bloco criado no pátio da Figura 8.

Figura 9 – Criação de um bloco em um pátio.



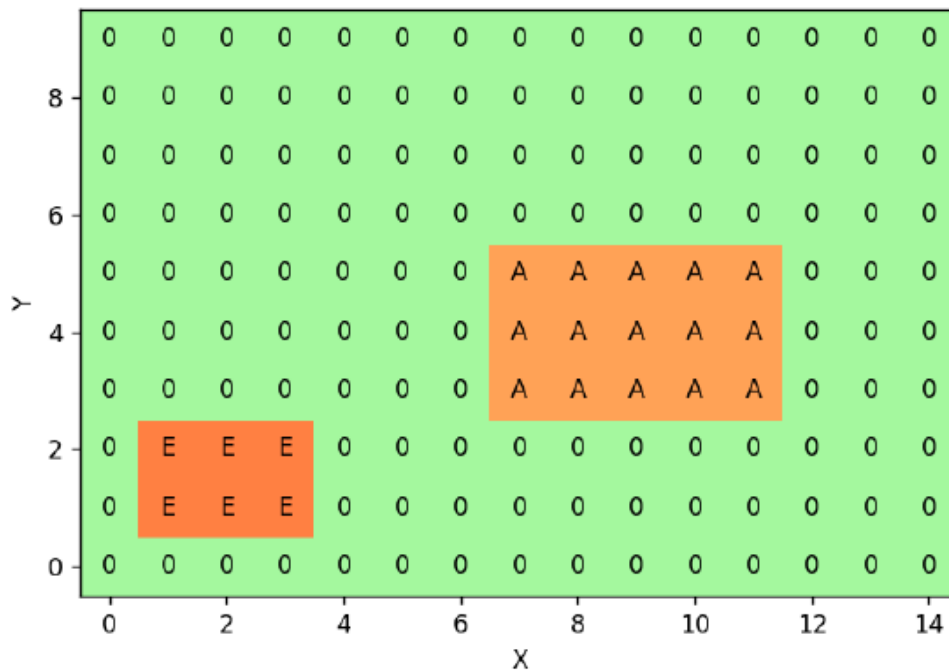
Fonte: Autor (2022)

Diferentemente da Figura 8, existem outros elementos não nulos no pátio que representam o bloco criado (atribuído com o caractere 'A' para visualização e identificação). O bloco A possui valores de X e Y máximos iguais a 5 e 3 que representam suas 5 colunas e 3 linhas, respectivamente. Os valores de origem x e y são iguais a 7 e 3, respectivamente, o início do bloco está presente na coluna 7 ($x = 7$) e encerra na coluna 11 ($x = 11$), totalizando as 5 colunas determinadas pelo X máximo. É possível adicionar blocos em quaisquer posições e quaisquer tamanhos desde que não haja sobreposição.

3.3.1.3 Estoque

Entende-se que o Bloco é uma entidade responsável por receber os contêineres que serão alocados no pátio. De maneira contrária, o Estoque representa uma área em que os contêineres se encontram inicialmente armazenados para serem alocados futuramente nos blocos. Visto que o trabalho representa terminais multimodais, o Estoque é uma área comum para que os contêineres recebidos de diferentes meios se concentrem próximos aos blocos que devem ser alocados. Também recebe os parâmetros de criação X e Y máximos e (x, y) de origem e é mostrado um exemplo na Figura 10.

Figura 10 – Criação do estoque de contêineres.



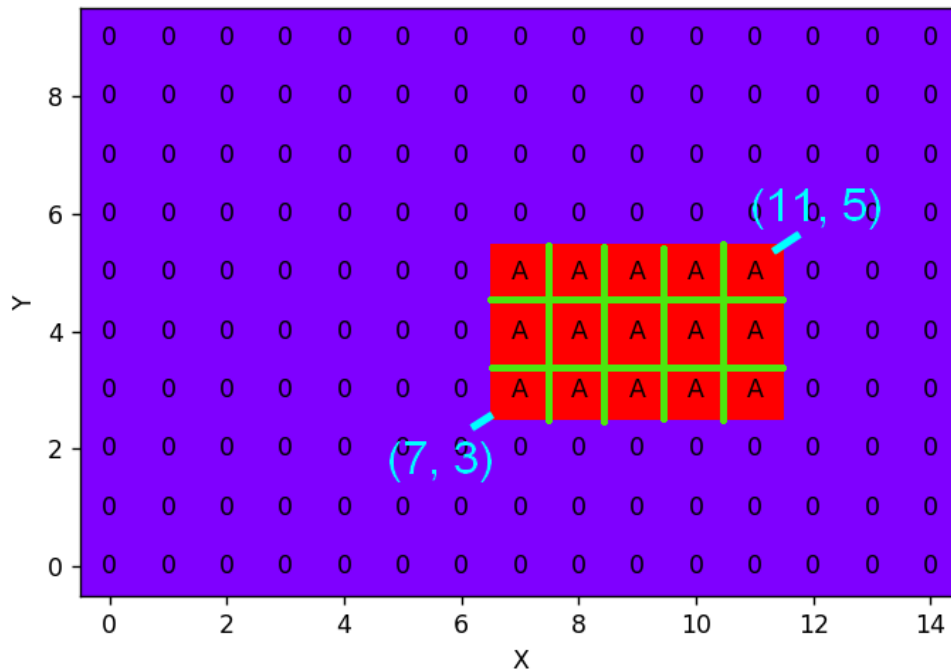
Fonte: Autor (2022).

Semelhante ao bloco, o estoque também possui um caractere de identificação e visualização, porém, é restritamente o caractere 'E' e este não ser atribuído aos blocos. Para a criação deste estoque, os valores de X e Y máximos são iguais a 3 e 2 que representam suas 3 colunas e 2 linhas, respectivamente. Os valores de origem x e y são iguais a 1 e 1, o início do bloco está presente na coluna 1 ($x = 1$) e encerra na coluna 3 ($x = 3$), totalizando as 3 colunas determinadas pelo X máximo.

3.3.1.4 Pilha

O quarto nível da hierarquia não requer uma criação própria visto que é uma consequência do nível superior. As pilhas são cada uma das células de um bloco e a necessidade dessa subdivisão é conhecer com exatidão o que há em cada posição do bloco. Além da visualização bidimensional, a pilha possui uma altura e esta é a última coordenada para definir a posição de um contêiner que será alocado no pátio. A Figura 11 destaca as pilhas para o bloco criado na Figura 10.

Figura 11 – Destaque das pilhas para o bloco da Figura 9.



Fonte: Autor (2022)

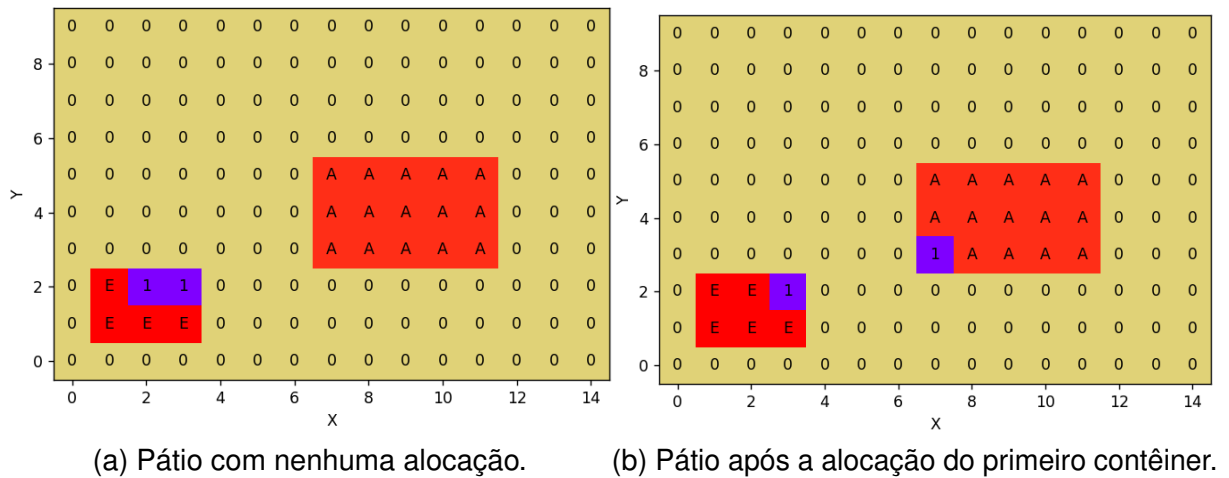
O número de pilhas de um bloco é, então, o produto entre os valores X e Y máximos desse bloco, neste caso: $5 \text{ (X máximo)} * 3 \text{ (Y máximo)} = 15$ (pilhas). A pilha é representada pelas coordenadas (x, y) , ou seja, $(7, 3)$ representa a célula mais a esquerda e mais "abaixo" do bloco A e o bloco A se estende até $(11, 5)$.

3.3.1.5 Contêiner

Por fim, o último elemento da hierarquia a ser inserido no pátio são os contêineres. Diferentemente das pilhas, definidas apenas pelas coordenadas (x, y) , os contêineres também recebem a coordenada z que representa o nível (altura) da pilha que o contêiner se encontra. Logo, a representação global da posição de um contêiner é no formato (x, y, z) .

Inicialmente, como explicado anteriormente, os contêineres permanecem no bloco de estoque que possui origem localizada na posição $(1, 1)$, X máximo igual a 3 e Y máximo igual a 2. A Figura 12a representa os contêineres no estoque que serão futuramente alocados no pátio.

Figura 12 – Demonstração de uma alocação de contêiner.



Fonte: Autor (2022).

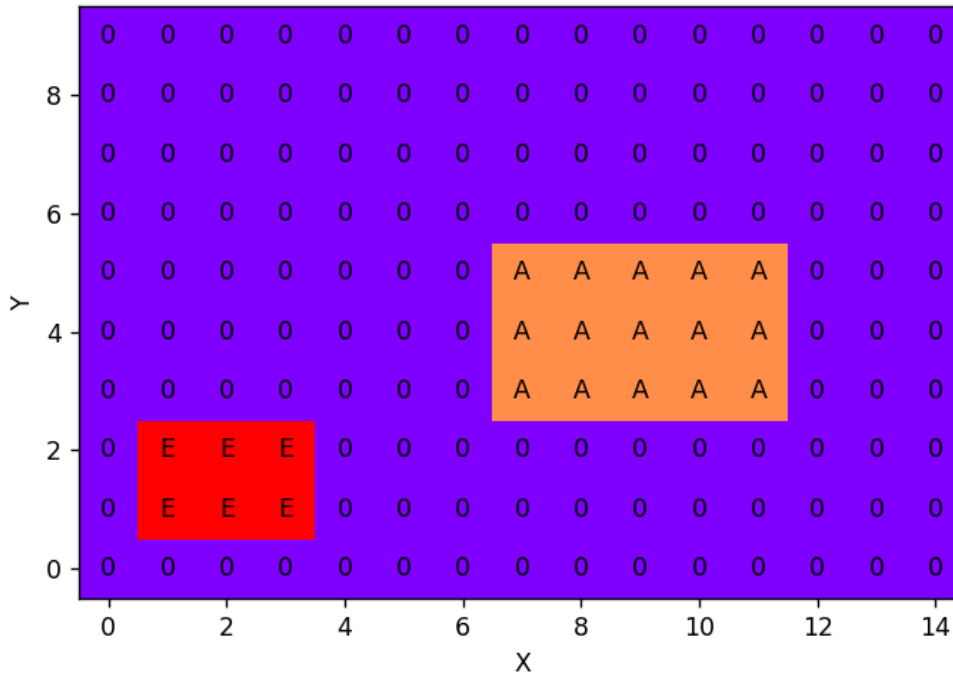
Nota-se que o estoque é um bloco que indica os contêineres a serem alocados no pátio. Seu espaço é representado pela letra 'E' e os contêineres presentes são representados por números. Por exemplo, na Figura 12a o estoque possui dois contêineres nas pilhas de posições (2, 2) e (3, 2), conseqüentemente, possuem os valores '1' nestas posições. Posteriormente, quando um contêiner for alocado a um bloco, o caractere que representa o bloco será substituído pelo número de contêineres alocados naquela pilha.

Como apresentado na Figura 12b, o contêiner que pertencia ao estoque na pilha de posição (2, 2) é alocado no pátio, então a célula de estoque que continha tal contêiner muda seu valor para 'E' e um caractere 'A' do bloco para '1', indicando a movimentação do contêiner de uma posição para outra do pátio. Para o exemplo da Figura 26, a posição inicial do contêiner era (2, 1, 0) e obtém-se a posição final (7, 3, 0), designando que 2 e 7 representam o valor no eixo x , 1 e 3 o valor no eixo y e 0 o valor no eixo z que indica um contêiner sobre o piso.

3.3.1.6 Rua

Este elemento é definido da mesma maneira que a Pilha, um elemento gerado por consequência da criação de um nível superior. Uma vez que os blocos do pátio e a área de estoque de contêineres estão definidos, o resto do pátio pode ser definido como *Rua*, representado pelas células de valor 0 e coloridos em roxo na Figura 13.

Figura 13 – Criação de um ativo e representação completa do pátio.



Fonte: Autor (2022).

Essa área representa locais trafegáveis pelo último elemento, explicado a seguir.

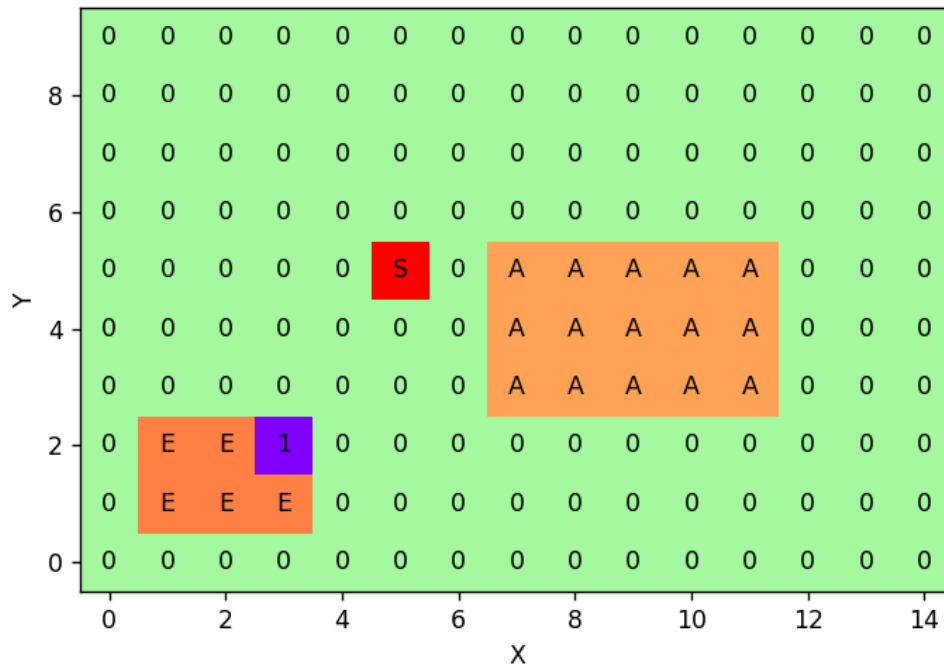
3.3.1.7 Ativo

Para que a alocação dos contêineres no pátio seja possível, um ativo (*reach stacker*) é responsável por movimentar os contêineres do estoque até a posição definida. Vale ressaltar que o ativo também pode se movimentar dentro de um bloco para efetuar a alocação, mas utiliza a rua majoritariamente para trafegar dentro do pátio.

A Figura 14 apresenta o pátio com todos os elementos necessários para representá-lo completamente.

- Bloco: representado pelo conjunto de caracteres 'A', é o espaço para alocar os contêineres.
- Estoque: exemplificado com o caractere 'E', é o local de retirada dos contêineres.
- Contêiner: simbolizado com o número 1, representa o próprio contêiner.
- Ativo: representado pelo caractere 'S', responsável por se movimentar até o estoque, coletar o contêiner e alocar em uma posição válida de um bloco.

Figura 14 – Criação de um ativo e representação completa do pátio.



Fonte: Autor (2022).

Além do conhecimento de todos elementos necessários para realizar a alocação de um contêiner, as características inerentes ao contêiner e ao bloco são indispensáveis para definir sua posição final.

3.3.2 Características dos componentes de implementação

Como segunda etapa do Desenvolvimento do Modelo e implementação conforme Figura 7, esta seção complementa a seção anterior definindo características dos elementos do pátio que são necessárias para realizar a alocação do contêiner.

3.3.2.1 Características do contêiner e bloco

No estudo do CSP é importante definir quais são as posições possíveis ou não para alocar um contêiner em um pátio de terminal e assim, alguns dados do mesmo devem ser verificados. O Quadro 4 apresenta as características de um contêiner que devem ser levadas em consideração bem como o uso de cada uma destas, seus tipos e seus possíveis valores.

Quadro 4 – Características de um contêiner.

Variável	Uso	Tipo	Valor Possível
Estado	Verifica se o contêiner está cheio ou vazio	number	1 (cheio) ou 0 (vazio)
Cliente	Nome do cliente	string	Qualquer
Tipo	Tipo do contêiner	string	"DRY"ou "REEFER"

Fonte: Autor (2022).

Tais características são relacionadas ao contêiner, porém, adicionalmente exigem-se as informações dos blocos presentes no pátio para então designar em qual bloco o contêiner poderá ser alocado. As variáveis de decisão são as mesmas do Quadro 4, entretanto, todas variáveis devem receber uma lista (array) a fim de definir quais valores aceitam, por exemplo:

Código 3.1 – Criação de um Bloco

```
BlocoA: {
  estado = [0, 1],
  cliente = ["CLIENTE1", "CLIENTE2"],
  tipo = ["DRY"]
}
```

A partir do Código 3.1, um bloco denominado *BlocoA* é criado. Como o atributo "estado" recebe os valores 0 (vazio) e 1 (cheio), o *BlocoA* pode alocar contêineres vazios ou cheios. Diferentemente do campo "tipo" que só aceita contêineres "DRY".

Para elucidar ainda mais este conceito, o Código 3.2 ilustra a composição de dois contêineres para verificar a possibilidade de alocação no *BlocoA*. Vale ressaltar que estes contêineres são fictícios e os contêineres reais já são previamente conhecidos pelo sistema.

Código 3.2 – Criação de contêineres.

```
Container1: {
  estado = 1,
  cliente = "CLIENTE1",
  tipo = "DRY"
}
```

```
Container2: {
  estado = 0,
```

```

    cliente = "CLIENTE2",
    tipo = "REEFER"
}

```

As variáveis "estado" e "cliente" diferem nos dois contêineres, porém, pelo fato de o *BlocoA* aceitar os valores do *Container1* e *Container2* para estes campos, há possibilidade de alocação por enquanto.

Entretanto, o campo "tipo" admite apenas um valor ("DRY") para o *BlocoA* e os contêineres também diferem neste campo. Consequentemente, apenas o contêiner que contém o valor igual a "DRY" para "tipo" (*Container1*) poderá ser alocado no *BlocoA*.

Desta forma, o *Container2* necessita ser alocado em outro Bloco presente no pátio ou, caso não haja nenhum bloco compatível, o programa deve retornar (com uma mensagem de erro) que não há possibilidade de alocação para tal contêiner.

3.3.2.2 Características do ativo para alocação

Como foi abordado na Seção 3.3.1.7, o ativo será responsável por efetuar a alocação do contêiner no pátio. O único ativo empregado neste trabalho para realizar as alocações será uma *reach stacker*.

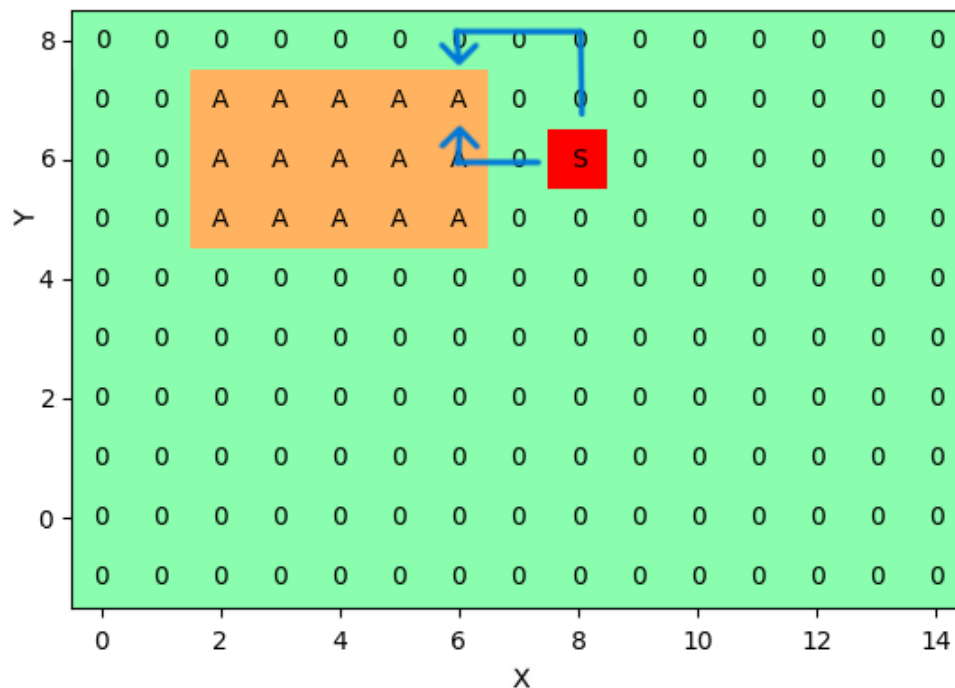
Assim como os contêineres e blocos possuem suas características, o mesmo ocorre para o ativo. Sua movimentação ocorre apenas na direção dos eixos x e y do pátio, ou seja, ocorre apenas horizontal e verticalmente, não há deslocamento diagonal.

Após verificar se um contêiner pode ser alocado em uma posição de um bloco, é necessário analisar se o ativo é capaz de atingir tal posição. As duas características que definem a viabilidade de atingir uma posição para o ativo são: direção e alcance.

Como o próprio nome sugere, a direção limita os contêineres de serem alocados em uma única orientação. Tal característica é originada pelo fato de os contêineres serem retangulares e não quadrados, limitando a manipulação dos mesmos. Uma vez que a posição desejada é atribuída ao contêiner, o ativo irá, obrigatoriamente, posicionar-se em outra célula para efetuar a alocação. Neste trabalho, o contêiner só é permitido ser alocado caso as posições acima ($y + 1$) ou abaixo ($y - 1$) da posição desejada estiverem livres para que o ativo consiga alcançá-la.

Com 4 possibilidades possíveis, a única que apresenta uma não possibilidade de alocação é onde ambas posições (acima e abaixo em relação à posição desejada) estão ocupadas. A Figura 15 ilustra duas trajetórias possíveis do ativo para alocar uma contêiner na posição (1, 6) do pátio.

Figura 15 – Caminhos possíveis para alocar na posição desejada.

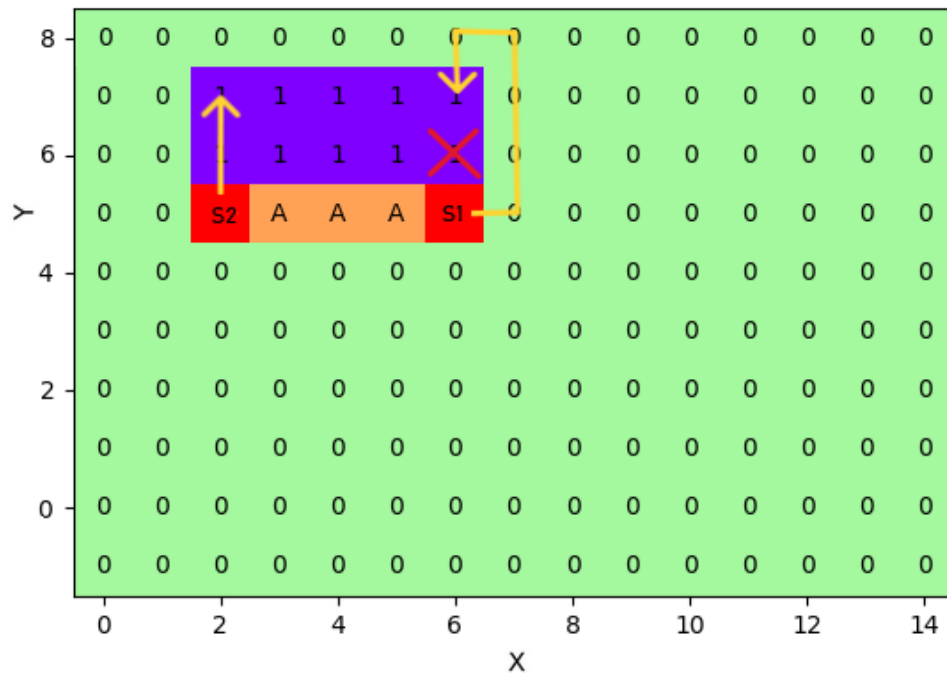


Fonte: Autor (2022).

Apesar de apenas duas trajetórias serem ilustradas, outras também são possíveis contanto que terminem nas posições (0, 6) ou (2, 6), acima e abaixo da posição desejada, respectivamente. Salienta-se que o ativo não poderia alocar o contêiner na posição desejada (1,6) pelas coordenadas (1, 5) e (1, 7), sendo os lados esquerdo e direito, respectivamente.

O alcance define a quantas células de distância em relação à posição desejada o ativo é capaz de efetuar a alocação. Ou seja, um alcance igual a 2 permite que, mesmo com uma linha de contêineres (linha de células com o valor 1 que representam a presença de um contêiner) bloqueando a chegada direta do ativo na posição desejada (duas células de distância), seja possível alcançar o objetivo. A Figura 16 demonstra dois ativos, representados por S1 e S2, com diferentes alcances e as respectivas possibilidades de alocação.

Figura 16 – Caminhos para ativos de diferentes alcances.

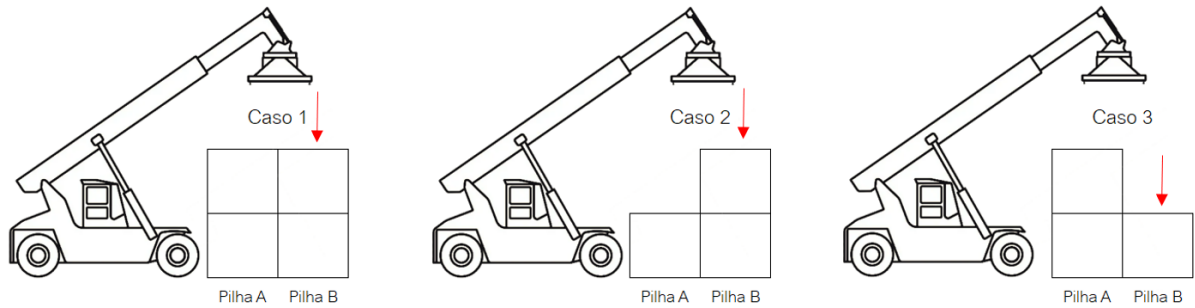


Fonte: Autor (2022).

Considera-se que o ativo S2 tenha alcance igual a 2 e o S1 igual a 1. Como explicado no parágrafo anterior, o ativo S2 mesmo posicionado em (2, 3) supera a linha de bloqueio (contornada em amarelo) e é capaz de realizar a alocação na posição (2, 1) com 2 células de distância. O ativo S1, pelo contrário, não possui alcance suficiente para ultrapassar o bloqueio da posição que se encontra (6, 3) até a posição desejada (6, 1). Apenas permite alocações nas posições que é capaz de chegar diretamente na posição desejada evitando obstáculos e com distância igual a 1, como demonstrado com a trajetória de linha amarela.

Nota-se também que, com o alcance igual ou superior a 2, seria necessário verificar a altura das pilhas frente à posição desejada no algoritmo de decisão. Uma vez que, além da limitação do alcance, a própria altura limitaria fisicamente o ativo de alocar um contêiner. Para elucidar, a Figura 17 mostra o que seriam estes casos observados de uma vista lateral. A seta vermelha indica a posição desejada para alocar o contêiner.

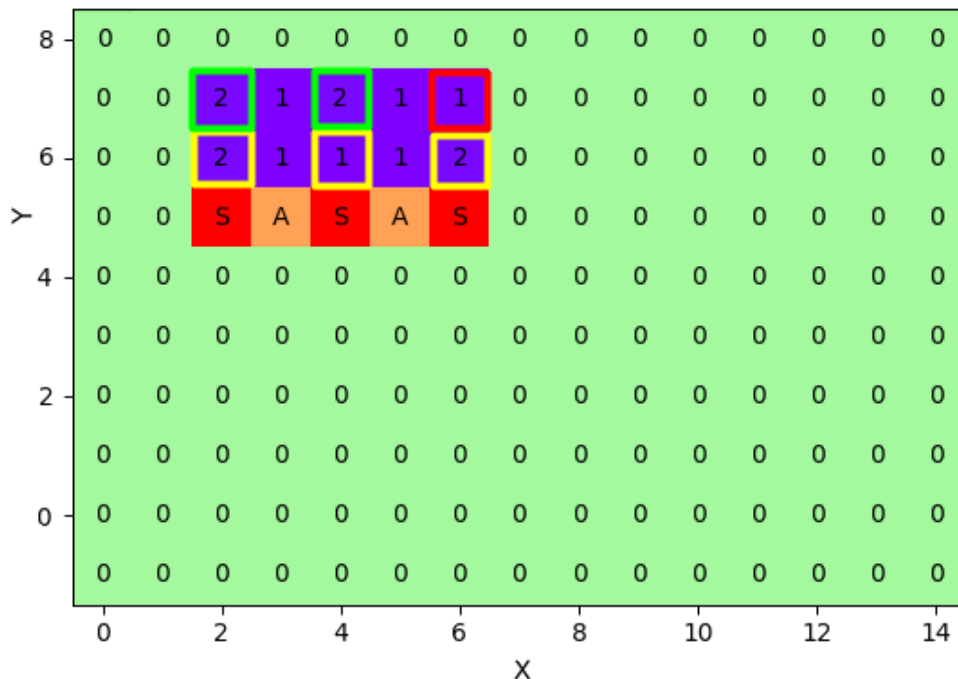
Figura 17 – Vista lateral para os casos da Figura 18.



Fonte: Autor (2022).

Para o Caso 1, as pilhas A e B possuem a mesma altura (mesmo número de contêineres empilhados), assim, o ativo tem alcance suficiente para alocar o contêiner na posição desejada da pilha B. Já para o Caso 2, a pilha B tem uma altura maior e não é um empecilho para que a alocação ocorra. No Caso 3, a altura da pilha A é superior a altura da pilha B e obstrui a manipulação necessária para que o contêiner seja alocado na posição desejada (indicado pela seta vermelha) na pilha B. Portanto, se a altura da pilha da posição desejada for igual ou superior a pilha a sua frente, o contêiner pode ser alocado, caso contrário, não é possível aloca-lo. A Figura 18 descreve o pátio simulado neste trabalho para os Casos 1, 2 e 3 da esquerda para a direita, respectivamente.

Figura 18 – Ativos de alcances iguais a 2 frente à pilhas de diferentes alturas.



Fonte: Autor (2022).

Os pontos de alocação do contêiner destinados para cada ativo são, da esquerda para direita: (1, 2), (1, 4) e (1, 6). Como os três ativos (representados pela letra 'S' e que indicam as *reach stackers* da Figura 18) possuem alcance igual a

2, deveriam ser capazes de alocar o contêiner com a distância considerada. Porém, deve-se considerar a altura da pilha que se deseja alocar o contêiner, assim como a altura da pilha em frente a esta.

As pilhas contornadas em amarelo são as pilhas em frente às pilhas da posição desejada para alocação. Já as pilhas contornadas em verde e vermelho simbolizam a pilha de destino para cada um dos ativos daquela mesma coluna, sendo verde uma alocação possível e vermelho uma alocação inviável. Neste trabalho, considera-se que as reach stackers tem alcance igual a 1.

3.3.3 Definição do modelo e implementação

3.3.3.1 Variáveis de tomada de decisão

Esta seção aprofunda alguns conceitos vistos nas seções 3.3.2.1 e 3.3.2.2 e explora novas premissas que baseiam o algoritmo da tomada de decisão a ser implementado neste trabalho.

1. Endereço não pode estar bloqueado

Está diretamente relacionado ao alcance do ativo. O endereço sugerido (posição desejada) para alocar o contêiner não pode estar bloqueado para movimentação. É possível bloquear endereço para movimentação caso haja manutenção da rua, outra operação em andamento ou até mesmo acidente.

2. Máximo do empilhamento suportado

Seja por capacidade operacional da reach stacker ou pela capacidade do piso, existe o máximo de 4 contêineres que podem ser empilhados quando estão cheios e 5 para contêineres vazios.

3. Se o contêiner é do tipo REEFER ou se há necessidade de colocar na tomada

Na maioria das vezes, os contêineres refrigerados precisam ser colocados em tomadas, então, se o contêiner é REEFER e requer tomada, o mesmo só poderá ser colocado em blocos que têm tomada. Para isso, também deverá ser levado em consideração se há tomada disponível ou não. Os blocos designados para os contêineres do tipo REEFER são indicados pelo caractere 'R', obrigatoriamente.

4. Estado do contêiner

Existe uma separação de contêineres cheios dos vazios, porém, não é obrigatória. Esta é definida conforme configuração, ou seja, pode ser alterada de acordo com a necessidade do usuário. Neste trabalho, haverá a opção de separação entre contêineres cheios e vazios.

5. Cliente do contêiner

Alguns pátios de contêiner não permitem a mistura de clientes para facilitar a organização, mas não necessariamente é um item obrigatório. É definida conforme configuração, ou seja, pode ser alterada de acordo com a necessidade do usuário. Neste trabalho, haverá a opção de separação entre clientes.

6. Tipo do contêiner

O tipo do contêiner também pode ou não permitir a mistura de diferentes tipos, conforme configuração. Apesar de REEFER ser um tipo, o item 3 se refere estritamente aos contêineres do tipo REEFER que estão cheios, enquanto os que estão vazios e são do tipo REEFER entram neste quesito. Neste trabalho, haverá a opção de separação entre tipos.

7. Deadline do contêiner

Uma variável no formato data/hora que define o tempo máximo para o contêiner ser retirado do pátio. Uma deadline maior significa mais tempo de estadia no pátio em relação a outra deadline, então a deadline 01/01/2023 é maior que 31/12/2022. Pode ser considerada a variável de maior complexidade e importância, pois um contêiner com deadline maior não pode ser colocado em cima de um contêiner com deadline menor. Isto ocorre a fim de evitar remoções quando for retirar o contêiner, ou seja, realocações e custo adicional.

3.3.3.2 Cálculo do custo

O intuito deste trabalho é minimizar os custos de operação para o processo de alocação de contêineres em um terminal multimodal, por este motivo, é de suma importância definir quais variáveis influenciam a função de custo.

Como citado na Seção 3.2, o tempo de alocação será o indicador de performance (KPI) nos métodos de alocação e está ligado diretamente a distância viajada pela reach stacker a cada alocação. Portanto, esta será a variável que define o cálculo do custo de alocação.

A modelagem matemática do custo leva em consideração o ambiente descrito na Seção 3.3.1 composto por pátio, bloco, contêineres, rua, ativo e o sistema de coordenadas (x, y, z) .

E : conjunto de contêineres no estoque para serem alocados,

X : valor máximo do pátio na horizontal,

Y : valor máximo do pátio na vertical,

Z : altura máxima de empilhamento permitido,

x : coordenada horizontal da posição de um contêiner, $1 \leq x \leq X$,

y : coordenada vertical da posição de um contêiner, $1 \leq y \leq Y$,

z : altura da posição de um contêiner, $1 \leq z \leq Z$,

(x, y, z) : coordenadas no valor horizontal x , valor vertical y e altura z do pátio,

N_E : quantidade de contêineres no estoque E ,

N_P : quantidade de posições possíveis nas coordenadas (x, y, z) para a alocação do contêiner,

N_K : quantidade de ativos disponíveis no pátio,

i : sequência de alocação do contêiner, $1 \leq i \leq N_E$,

j : índice da posição possível nas coordenadas (x, y, z) em que o contêiner pode ser alocado, $1 \leq j \leq N_P$,

k : índice do ativo utilizado para realizar a alocação do contêiner, $1 \leq k \leq N_K$

$P_{i,j}$: 1 se a alocação do contêiner i na posição j for possível, 0 caso contrário,

$PA_{i,j}$: 1 se a alocação do contêiner i na posição j for realizada, 0 caso contrário,

$D_{i,j,k}$: distância percorrida pelo ativo k para alocar o contêiner i na posição j .

A partir das variáveis apresentadas, a Equação 1 descreve o cálculo do custo operacional empregado neste trabalho para o CSP.

$$C = \sum_{i=1}^{N_E} \sum_{j=1}^{N_P} \sum_{k=1}^{N_K} D_{i,j,k} \times P_{i,j} \times PA_{i,j} \quad (1)$$

A distância é calculada de acordo com o número de células do pátio que o ativo percorre para realizar a alocação, sendo o valor de cada movimentação entre células adjacentes igual a 1. Com a equação capaz de representar o custo, a última etapa antes de demonstrar o modelo é descrita a seguir.

3.3.3.3 Considerações e Restrições

Apesar de vários detalhes terem sido descritos ao decorrer deste trabalho, faz-se necessário listar as considerações já citadas assim como novas restrições a fim de elucidar o entendimento do modelo todo.

- Ambiente de implementação é descrito no sistemas de coordenadas (x, y, z) e possui as entidades: pátio, bloco, estoque, contêiner, pilha, rua e ativo.
 - Pátio: espaço que contém todas entidades;
 - Bloco: espaço para alocar os contêineres até sua saída do pátio;
 - Estoque: espaço que recebe os contêineres temporariamente antes de alocá-los no bloco;
 - Contêiner: o próprio contêiner;
 - Pilha: uma pilha dentro de um bloco;
 - Rua: espaço que não é bloco nem estoque, trafegável;
 - Ativo: equipamento responsável por alocar os contêineres.
- A configuração do pátio é conhecida previamente à alocação:

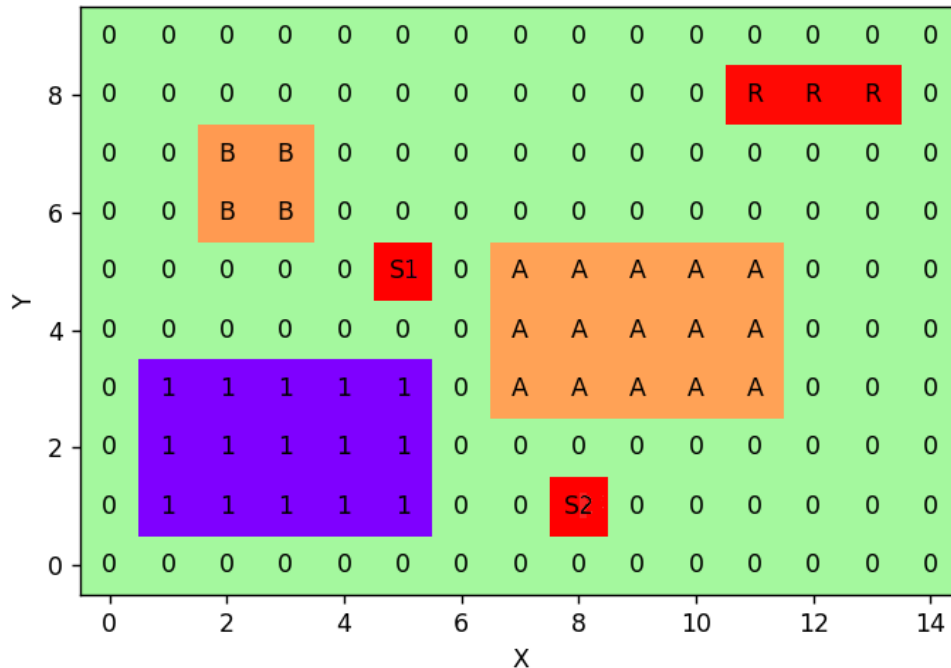
- Características dos contêineres (alocados e para serem alocados, referentes ao estoque) incluindo sua posição;
- Características dos blocos (origem, dimensões e contêineres aceitáveis);
- Características dos ativos (posição, alcance).
- Regras de Atribuição empregadas:
 - Blocos seguem a regra das Áreas Dedicadas;
 - Baias não são utilizadas;
 - Pilhas seguem as regras desenvolvidas pelo autor.
- A possibilidade da alocação de um contêiner é ditada pelas variáveis de tomada de decisão em relação a um bloco;
- Ativos não se movimentam diagonalmente;
- Apenas uma alocação é feita por vez;
- O Estoque é preenchido com uma camada de contêineres apenas.

3.3.3.4 *Demonstração do modelo e implementação*

Esta seção serve para exemplificar a seção anterior e demonstrar todos os procedimentos necessários para realizar a alocação dos contêineres desde a criação dos componentes do ambiente até como calcular o custo da alocação de cada contêiner. Primeiramente, é necessário criar o ambiente de implementação e o Código 5 apresenta como este processo é feito.

Para este código, três blocos são criados com os caracteres A, B e R, o primeiro recebe contêineres do tipo DRY e REEFER, o segundo apenas DRY e o último é especial para contêineres do tipo REEFER. Em relação ao estado dos contêineres, o bloco A e R recebem os dois tipos (cheio e vazio) enquanto o bloco B recebe apenas vazios. Já para os clientes, todos clientes são aceitos em todos os blocos. Apenas 2 ativos são definidos e os contêineres são distribuídos de 15 em 15. O ambiente de implementação resultante é exibido na Figura 19.

Figura 19 – Ambiente de implementação conforme Código 5.



Fonte: Autor (2022).

Nota-se que há um bloco completamente preenchido por valores iguais a 1, este é o estoque com os 15 contêineres gerados. De acordo com uma das restrições citadas anteriormente, o estoque possui apenas um nível de contêineres, isto é, não há empilhamento de contêineres sobre os outros. Logo, se 16 contêineres fossem alocados para este pátio, o estoque possuiria tamanho insuficiente visto que ocupa exatamente 15 contêineres, exigindo a mudança na dimensão do estoque. Além disso, tem-se os 3 blocos e os 2 ativos gerados como esperado.

Dentro do ambiente de implementação, um conjunto de contêineres foi gerado no código demonstrado anteriormente porém sem apresentar suas características específicas. A Tabela apresenta os contêineres e todas as características que carregam.

Tabela 1 – Informações dos contêineres do Código 5.

Contêiner	Armador	Número	Estado	Tipo	Cliente	Deadline
0	BRD	323	VAZIO	DRY	BETA	2022-11-14 10:39:00
1	MSK	625	CHEIO	REEFER	BETA	2022-01-23 18:59:00
2	MSK	134	VAZIO	DRY	GAMA	2022-04-24 18:30:00
3	MSK	451	CHEIO	DRY	BETA	2022-05-20 20:16:00
4	BRD	871	VAZIO	DRY	ALFA	2022-08-24 14:25:00
5	ONE	139	VAZIO	DRY	XIS	2022-04-14 20:13:00
6	BRD	750	VAZIO	DRY	GAMA	2022-08-15 20:34:00
7	ONE	619	VAZIO	DRY	ALFA	2022-08-11 10:35:00
8	MSK	273	VAZIO	DRY	XIS	2022-06-12 22:32:00
9	MSK	914	VAZIO	DRY	ALFA	2022-02-11 22:51:00
10	BRD	618	VAZIO	DRY	XIS	2022-09-19 16:22:00
11	BRD	334	CHEIO	DRY	GAMA	2022-12-15 10:53:00
12	MSK	735	VAZIO	DRY	ALFA	2022-07-22 20:41:00
13	MSK	741	VAZIO	DRY	GAMA	2022-05-22 14:47:00
14	ONE	216	CHEIO	DRY	BETA	2022-12-14 20:52:00

Fonte: Autor (2022).

Os contêineres são gerados de maneira aleatória para que os modelos de alocação não se limitem a apenas um conjunto de contêineres quando forem analisados. Para a coluna Armador os valores variam entre: BRD, MSK e ONE, representando siglas de empresas logísticas atuais. A coluna Número é composta por 3 algarismos aleatórios de 0 até 9. O Estado, como já mencionado, assume os valores "VAZIO" e "CHEIO". Cliente recebe 4 valores fictícios: ALFA, BETA, GAME, XIS. Por fim, Deadline é a data de retirada do contêiner do pátio no formato ANO-MÊS-DIA HORA-MINUTO-SEGUNDO.

As colunas Armador e Número são utilizadas para identificar os contêineres de forma similar aos terminais reais. Já as colunas Estado até Deadline se referem as variáveis de tomada de decisão da Seção 3.3.3.1 dos números 3 à 7.

Posterior à criação completa do ambiente de implementação, o processo de alocação se encontra disponível para execução. A continuidade da Figura 19 é a alocação do primeiro contêiner. Para ilustrar a alocação na Figura 26, o contêiner a ser alocado será o contêiner posicionado na origem do estoque, o ativo que carregará o contêiner será o mais próximo deste contêiner e a posição desejada será a mais próxima disponível de acordo com as características do contêiner e dos blocos presentes no pátio.

Para definir o caminho mínimo entre a posição de estoque do contêiner a ser alocado e de um ativo, assim como o caminho mínimo entre a posição de estoque do contêiner a ser alocado e a posição desejada de alocação, o algoritmo Breadth-First Search (BFS) GeeksForGeeks (2022) foi implementado e é explicado a seguir.

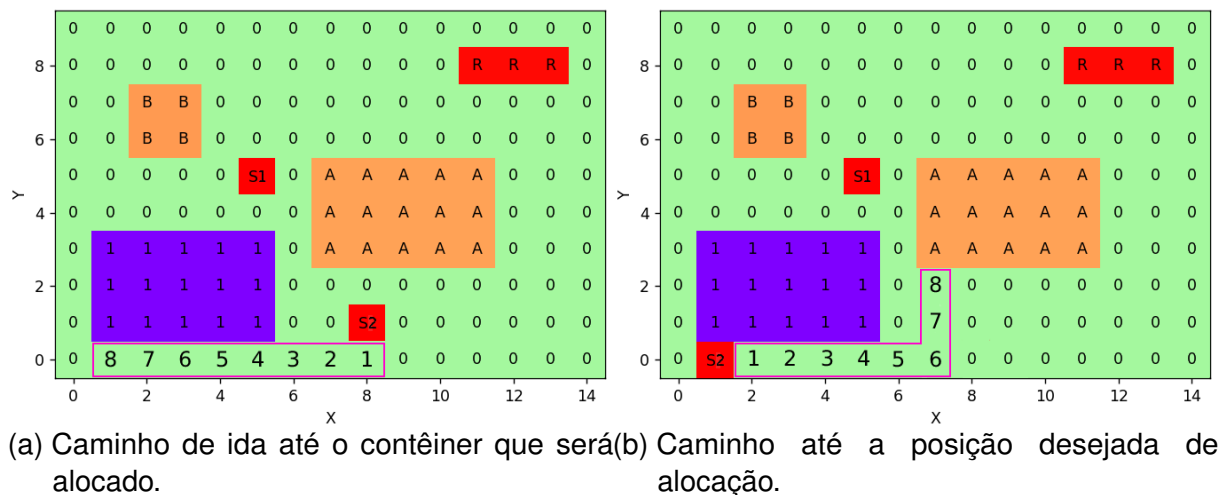
Partindo da posição inicial como o nó raiz, o método de pesquisa em largura (BFS) busca uma solução expandindo o nó raiz para os próximos níveis, todos os

movimentos possíveis, não quebrando as condições e resultando em novas condições sendo desenvolvidas ao máximo possível. A busca termina quando não houver mais nós ou novas condições que possam ser desenvolvidas. Todos os nós que são a posição de destino são soluções. O movimento permitido de objetos é Sihotang (2020):

- Mover pra direita ($x + 1$).
- Mover pra esquerda ($x - 1$).
- Mover pra cima ($y + 1$).
- Mover pra baixo ($y - 1$).
- Todos movimentos acima são permitidos se estiver em conformidade com as seguintes proposições:
 - Não visitar uma posição que é um obstáculo (pilha de contêineres de qualquer tamanho).
 - Não visitar um ponto que já foi visitado.

Com o entendimento do algoritmo, a alocação é apresentada a seguir. As Figuras 20a e 20b ilustram o caminho de ida até o contêiner do estoque e o caminho até a posição desejada para alocação, respectivamente.

Figura 20 – Caminho do ativo para uma alocação de contêiner.

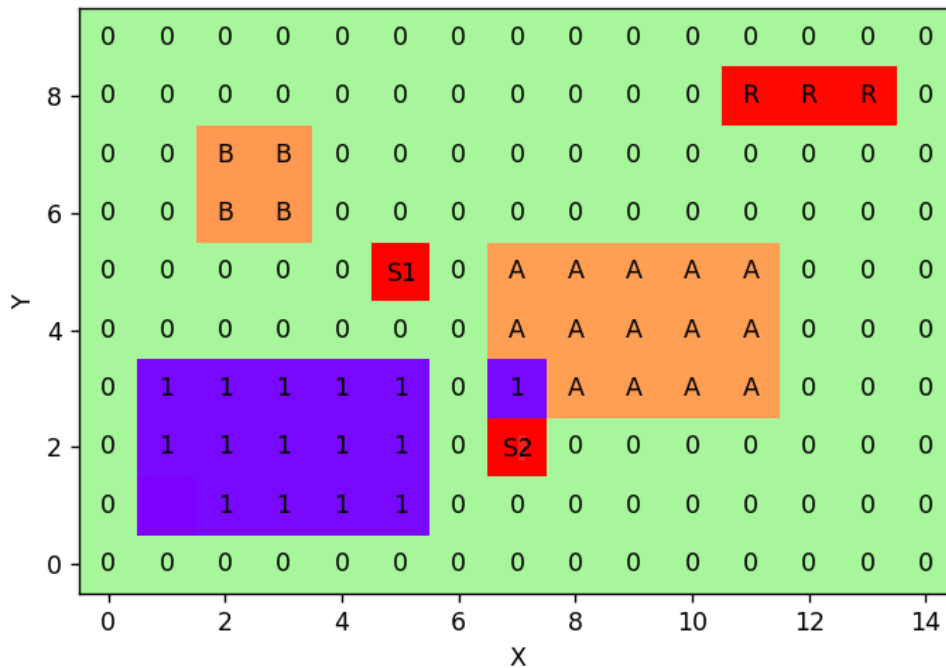


Fonte: Autor (2022).

Ambos caminhos estão destacados na cor rosa, e em cada célula destes caminhos há um valor que se refere à distância da posição do ativo S2 em relação àquela célula, de acordo com o caminho mínimo encontrado pelo algoritmo BFS.

Por exemplo, em ambas figuras o valor da célula final do caminho é igual a 8 o que indica uma distância de 8 células para chegar até estas duas posições, tornando o custo (distância total percorrida) desta alocação igual a 16 (8 para ida ao estoque + 8 para ida à posição desejada). Vale ressaltar que cada visita a cada célula corresponde a uma unidade de comprimento (u.c). Após essa movimentação, o pátio assume a configuração abaixo.

Figura 21 – Configuração do pátio após alocação da Figura 20.



Fonte: Autor (2022).

Como uma das principais funções da implementação, a cada alocação de um contêiner uma figura que representa o pátio é apresentada e o programa gera uma saída no formato: *ATIVO* → *CONTEINER* → *BLOCO* → *POSICAO* → *CUSTOTOTAL* = *X*. Para o caso da Figura 21 a saída retornada é: $S_2 \rightarrow BRD323 \rightarrow A \rightarrow (7, 3, 0) \rightarrow CUSTOTOTAL = 16$. Vale ressaltar que a posição retornada é dada nas coordenadas (x, y, z) globais (referentes ao pátio), o retorno da identificação do bloco é informativa e não indica que seja a posição (x, y, z) do bloco.

É possível observar que o ativo S_2 assume uma nova posição $(7, 2)$ e o contêiner se encontra na posição gerada pela saída do programa $(7, 3, 0)$ que é indicado pelo número 1 no lugar do caractere de identificação 'A' do *BlocoA*. O estoque adquire uma célula vazia e não igual a '0' para indicar que um contêiner foi retirado daquela posição e ainda é uma célula de estoque e não uma célula de rua.

Para completar a alocação de todos os contêineres, bastaria repetir o processo de acordo com o método escolhido. Ou seja, definir um critério para selecionar qual contêiner do estoque será escolhido e, de acordo com as características deste contêiner, em que posição do pátio este pode ser inserido e como esta posição será escolhida.

Portanto, o método é dividido em duas etapas: escolha do contêiner no estoque e escolha da posição dentro das possíveis, estas etapas são definidas a seguir.

3.4 DEFINIÇÃO DE HEURÍSTICAS

Como o intuito deste estudo é definir um algoritmo de tomada de decisão para alocação dos contêineres que minimize o custo total de operação e identificar quais os parâmetros que mais impactam nesta tomada de decisão, quatro heurísticas foram implementadas para a primeira etapa do processo de alocação (escolha do contêiner no estoque), sendo estas:

1. Retirar os contêineres do estoque em ordem sequencial das coordenadas (x, y) (da esquerda para direita e de baixo para cima).
2. Retirar o contêiner do estoque que seja alcançável e esteja mais próximo de um ativo.
3. Retirar o contêiner do estoque que seja alcançável e possua maior deadline.
4. Dentre i contêineres mais próximos a u ativo que sejam alcançáveis, retirar o que possui maior deadline.

A primeira heurística foi criada com a finalidade de comparar futuros métodos com um modelo básico e direto, desta forma, é possível concluir se métodos mais complexos são realmente capazes de superar o desempenho de uma abordagem "simples".

Uma vez que a distância total percorrida pelos ativos é diretamente proporcional ao custo total, é o principal valor a ser minimizado. A segunda heurística surgiu com o princípio de um algoritmo guloso, isto é, procura o ótimo local a cada iteração (contêiner mais próximo do ativo naquele momento) e não um ótimo global ao final de todas iterações.

Já a terceira heurística foi montada baseada em uma das principais restrições deste trabalho. Dado que os contêineres de deadline maior não podem ser alocados sobre contêineres de deadline menor, o objetivo é, também a partir de uma abordagem gulosa, escolher o contêiner de maior deadline disponível.

Por fim, a quarta heurística procura mesclar a segunda e a terceira com o propósito de evitar o comportamento guloso ao mesmo tempo que abrange contêineres próximos e com deadline maior. Além disso, possui um parâmetro variável dentro da própria heurística que garante flexibilidade e uma análise ainda mais completa.

Já para a segunda etapa (escolha da posição de alocação), três heurísticas análogas as da primeira etapa foram implementadas, sendo estas:

1. Alocar em ordem sequencial das coordenadas (x, y) , se possível.
2. Alocar na posição possível mais próxima.
3. Alocar sobre o contêiner de menor deadline.

Os motivos de emprego destas heurísticas são os mesmos para as heurísticas da primeira etapa visto que são análogos. A principal diferença é o terceiro item da lista que busca contêineres de menor deadline possível. O intuito desta heurística é

evitar que contêineres de deadline muito baixa sejam alocados sobre contêineres de deadline muito alta, impedindo que outros contêineres de deadline média pudessem ser alocados sobre o contêiner de maior deadline.

Visto que a tomada de decisão para alocar um contêiner é feita em duas etapas e cada uma dessas etapas possui 4 heurísticas possíveis, há 12 combinações possíveis para definir o método de alocação de um contêiner.

Além das heurísticas, um pátio vazio (sem contêineres alocados) receberá um número de contêineres para serem alocados e o custo total da alocação de contêineres será calculado. A partir do pátio com nova configuração, o estoque de contêineres é reabastecido por outros contêineres para serem alocados. Este processo será repetido até que não haja mais espaço disponível para alocação a fim de analisar como as heurísticas se comportam com poucos e muitos contêineres presentes no pátio e qual quantidade máxima de contêineres alocados para cada heurística.

4 ANÁLISE DE RESULTADOS

A partir da definição dos métodos de alocação para as duas etapas, é necessário definir o ambiente de implementação para realizar o processo de alocação completo e comparar os resultados para cada combinação de heurística possível.

4.1 RELAÇÃO ENTRE CUSTOS E HEURÍSTICAS

A configuração do pátio utilizado para realizar a comparação entre as heurísticas é descrita definido no Código 5.

Observa-se que apesar de existir a possibilidade de separação entre estados, clientes e tipos neste trabalho, os blocos A, B e C do código apresentado aceitam todos os valores para as 3 variáveis. Apenas o bloco R difere no estado (permite apenas contêineres cheios) e tipo (apenas do tipo REEFER).

Para analisar os resultados, todas heurísticas foram combinadas e são separadas em 4 tabelas, mantendo a heurística da primeira etapa para cada tabela e variando a heurística da segunda etapa. As tabelas apresentam os valores de distância percorrida pelo ativo após a alocação de um número de contêineres.

Tabela 2 – Distância total percorrida para as heurísticas 1.1, 1.2 e 1.3.

Heurística 1ª etapa como 1			
Qtd. Cont.	1.1	1.2	1.3
15	214	184	226
30	468	402	544
45	744	628	873
60	1064	864	1246
75	1383	1139	X
90	1695	1427	X
105	2013	1747	X
120	2287	2141	X

Fonte: Autor (2022).

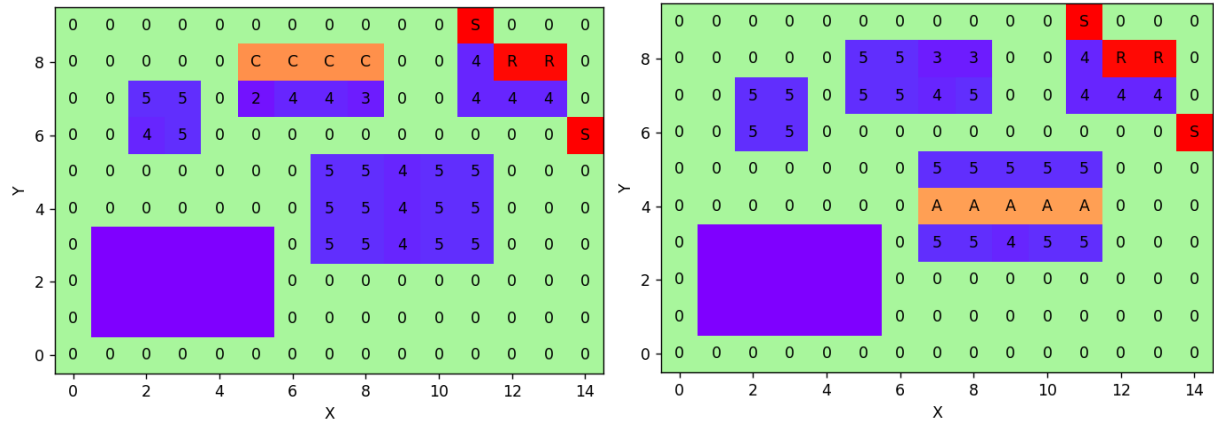
A Tabela 2 apresenta a heurística da primeira etapa como 1 (retirar os contêineres do estoque em ordem sequencial das coordenadas (x, y)) e relaciona o número de contêineres com as três heurísticas da segunda etapa.

A heurística da segunda etapa proposta por 3 não é capaz de alocar os 120 contêineres desejados e tem o pior valor de distância percorrida, 44% a mais que a heurística 2 e 17,1% a mais que a 1 quando se compara aos 60 contêineres alocados.

Já as heurísticas 1 e 2 da segunda etapa são ambas capazes de alocar todos contêineres e possuem valores similares para a distância percorrida, com uma pequena

melhor performance de 6,4% a menos no custo total para 2. Outro critério avaliado para definir o desempenho das heurísticas é o aproveitamento da área ocupada e por este motivo, as Figuras 22a e 22b são apresentadas abaixo.

Figura 22 – Comparação entre pátios após diferentes heurísticas aplicadas com 120 contêineres.



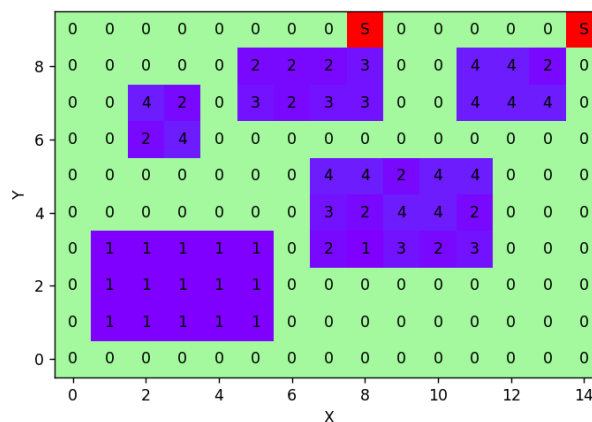
(a) Pátio após alocação com a heurística 1.1. (b) Pátio após alocação com a heurística 1.2.

Fonte: Autor (2022).

Como a heurística 1.2 escolheu as posições mais próximas, a linha que se estende de (7,5) até (11, 5) se tornou inutilizável pelo bloqueio das pilhas dos contêineres vizinhos. Portanto, por mais que a distância tenha sido minimizada e este é o principal objetivo do trabalho, o usuário deve estar atento às possíveis consequências que a busca pelo objetivo pode trazer.

Para fins comparativos, a Figura 23 representa o pátio após a aplicação da heurística 1.3.

Figura 23 – Pátio após a aplicação da heurística 1.3.



Fonte: Autor (2022).

O estoque de contêiner permanece cheio pois nem todos os contêineres foram alocados e grande parte do pátio não atinge uma altura superior ou igual a 4 (valor

máximo para pilhas de contêineres vazios).

A Tabela 5 apresenta a heurística da primeira etapa como 2 (retirar o contêiner do estoque que seja alcançável e esteja mais próximo de um ativo) e relaciona o número de contêineres com as três heurísticas da segunda etapa.

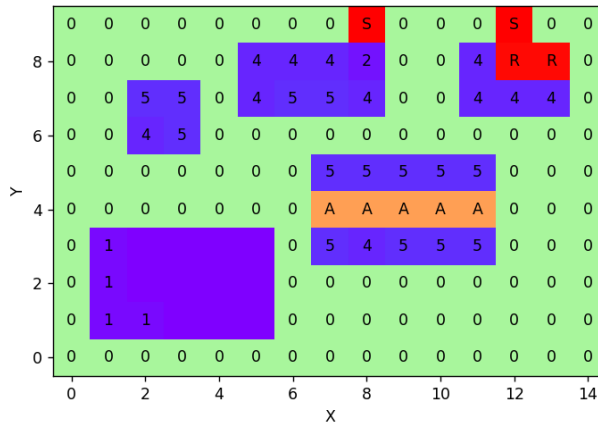
Tabela 3 – Distância total percorrida para as heurísticas 2.1, 2.2 e 2.3.

Heurística 1º etapa como 2			
Qtd. Cont.	2.1	2.2	2.3
15	196	140	220
30	441	325	497
45	683	508	784
60	964	705	1005
75	1264	921	1245
90	1452	1170	X
105	1661	1446	X
120	1896	X	X

Fonte: Autor (2022).

Para esta combinação, apenas a heurística 2.1 foi capaz de alocar os 120 contêineres. Uma explicação plausível para o não êxito da heurística 2.2 é o baixo aproveitamento da área ocupada mostrada na Figura 22b e a Figura 24 corrobora esta ideia.

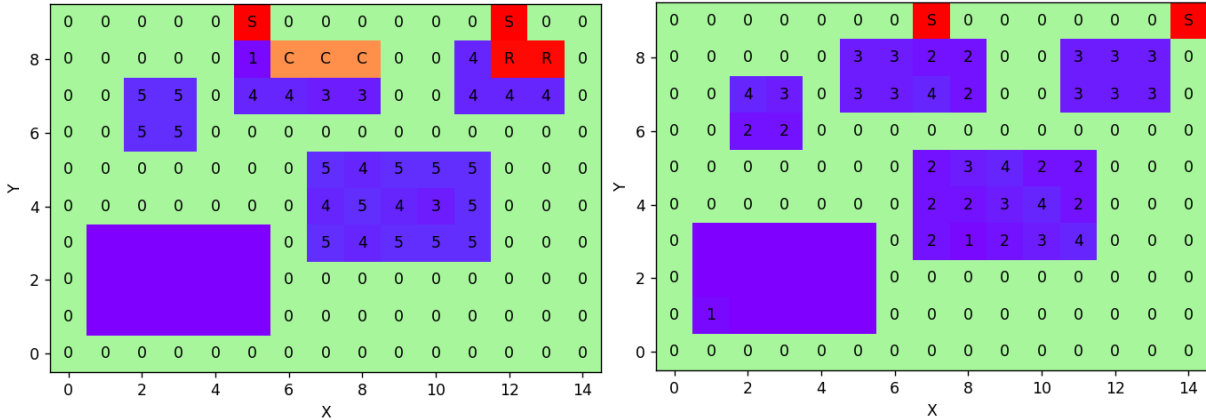
Figura 24 – Pátio após a aplicação da heurística 2.2.



Fonte: Autor (2022).

Os pátios para a alocação das outras duas heurísticas são representadas nas Figuras 26a e 26b.

Figura 25 – Comparação entre pátios após diferentes heurísticas aplicadas.



(a) Pátio após alocação com a heurística 2.1. (b) Pátio após alocação com a heurística 2.3.

Fonte: Autor (2022).

De forma semelhante as heurísticas da primeira etapa como 1, a primeira heurística da segunda etapa demonstra melhor aproveitamento enquanto 2 e 3 não possuem espaço suficiente para o preenchimento de 120 contêineres.

Por fim, as tabelas parece as heurísticas 3.x e 4.x são apresentadas.

Tabela 4 – Distância total percorrida para as heurísticas 3.1, 3.2 e 3.3.

Heurística 1º etapa como 3			
Qtd. Cont.	3.1	3.2	3.3
15	192	160	212
30	459	365	496
45	711	579	792
60	1024	794	1118
75	1324	1054	1474
90	1612	1323	1756
105	1885	1628	X
120	2131	2002	X

Fonte: Autor (2022).

Novamente, o comportamento se repete: a segunda etapa para a heurística 2 é a de menor valor total de custo enquanto a heurística 1 é 6,4% maior e a heurística 3 não completa a alocação de todos contêineres desejados.

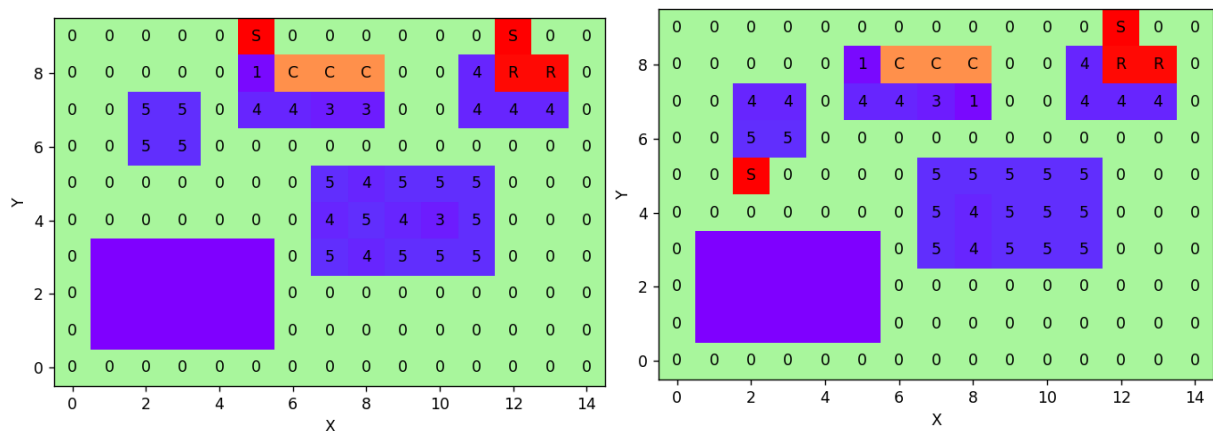
Tabela 5 – Distância total percorrida para as heurísticas 4.1, 4.2 e 4.3.

Heurística 1º etapa como 4			
Qtd. Cont.	4.1	4.2	4.3
15	193	148	215
30	436	329	488
45	703	517	789
60	983	722	1077
75	1274	959	1313
90	1526	1224	1607
105	1721	1508	X
120	1951	X	X

Fonte: Autor (2022).

Similar a primeira etapa com a heurística 2, apenas a segunda etapa com a heurística 1 é capaz de alocar todos os contêineres. Com um custo igual à 1951, é apenas 3% maior que o menor de todos os custos para 120 contêineres dentro das 12 heurísticas. Portanto, torna-se interessante comparar a configuração do pátio para as heurísticas 2.1 e 4.1.

Figura 26 – Comparação entre os pátios de menores custos.



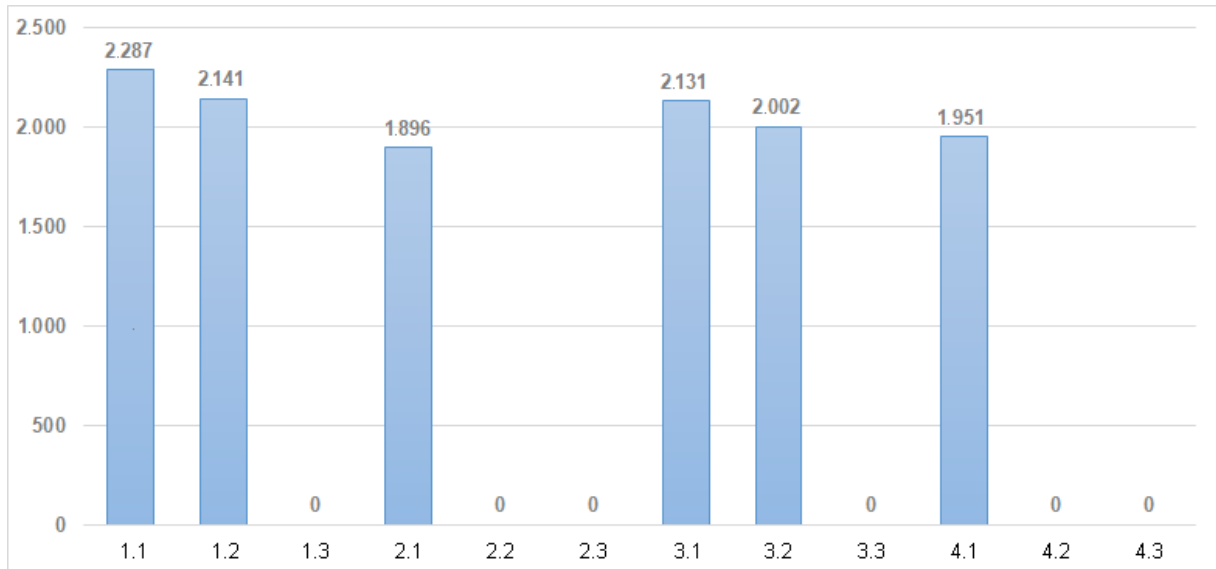
(a) Pátio após alocação com a heurística 2.1. (b) Pátio após alocação com a heurística 4.1.

Fonte: Autor (2022).

Apesar do alto grau de similaridade, ainda existem diferenças, porém, não são suficientes para desconsiderar a heurística 2.1 como a mais apropriada, visto que possui o menor custo e 100% de sucesso para alocar os 120 contêineres desejados.

Como última análise, a Figura 27 apresenta das distâncias totais percorridas (no eixo y) para cada heurística (no eixo x) com 120 contêineres para alocação.

Figura 27 – Distâncias totais de todas heurísticas para 120 contêineres.



Fonte: Autor (2022).

É possível observar que ao utilizar a terceira heurística para a segunda etapa, não foi possível alocar os 120 contêineres independente da heurística usada na primeira etapa. Em contrapartida, a primeira heurística para a segunda etapa obteve o resultado contrário, garantindo o empilhamento dos 120 contêineres para todas as combinações com a primeira etapa.

Apesar de as heurísticas 2.1 e 4.1 serem as únicas abaixo de 2.000 u.c, a heurística 3.2 se aproxima desta marca apesar do mal aproveitamento de espaço apresentado pela segunda heurística empregada nas segundas etapas apresentadas nas Figuras 22b e 24. Estas ideias reforçam a eficiência da primeira heurística para a segunda etapa e confirmam a heurística 2.1 como a de melhor desempenho.

5 CONCLUSÕES

O propósito deste trabalho atribuído ao objetivo geral de um desenvolvimento de um algoritmo capaz de definir a posição de alocação de um contêiner descarregado em um terminal multimodal com a finalidade de minimizar a distância total percorrida pelo ativo e os custos operacionais foi cumprido.

Um ambiente de implementação satisfatório foi criado a partir da research question: quais considerações influenciam na escolha de decisão da posição do contêiner no pátio e, a partir da comparação de diferentes métodos, qual algoritmo mais minimiza o custo operacional total para tal tomada de decisão? Tal ambiente foi extremamente necessário para compreender as hierarquias de um terminal de contêineres e poder atribuir características reais a um sistema virtual bidimensional. A análise em duas etapas foi de suma importância para definir quais as considerações influenciam na escolha da tomada de decisão.

Com 12 heurísticas propostas, foi definido que se deve retirar o contêiner do estoque que seja alcançável e esteja mais próximo de um ativo ao mesmo tempo que aloca os contêineres em uma ordem geográfica sequencial, nomeada a heurística 2.1 neste trabalho.

Como continuidade deste trabalho, a inclusão de mais características de cenários reais são desejadas, como: variar o alcance da reach stacker, permitir que contêineres que sejam realocados e mais de um ativo realizando alocação em determinado instante de tempo. Ademais, o emprego de um algoritmo mais sofisticado, como MIP, é recomendado para auxiliar na tomada de decisão com mais variedade de parâmetros.

Por fim, conclui-se que o estudo definiu os parâmetros mais influentes (distância dos ativos ao selecionar contêineres no estoque e sequência de posições alocadas) na tomada de decisão para a alocação de contêineres, sendo um ótimo ponto de partida para expansão de um estudo próspero.

REFERÊNCIAS

- ANTAQ. **Estatístico Aquaviário - Movimentação de Contêineres**. 2022. Disponível em: <http://ea.antaq.gov.br/>.
- ASPEREN, E. van; BORGMAN, B.; DEKKER, R. Evaluating impact of truck announcements on container stacking efficiency. **Flexible Services and Manufacturing Journal**, v. 25, p. 543–556, 2013.
- AZIZ, A. A.; ZAKARIA, A. Container stacking and retrieval prototype simulation using genetic algorithms. **Jurnal Teknologi**, Universiti Teknologi Malaysia, v. 11, n. 3, p. 61—74, jun. 2003.
- CHEN, L.; LU, Z. The storage location assignment problem for outbound containers in a maritime terminal. **International Journal of Production Economics**, Elsevier BV, v. 135, n. 1, p. 73–80, jan. 2012. Disponível em: <https://doi.org/10.1016/j.ijpe.2010.09.019>.
- CNT. **PIB do transporte tem crescimento expressivo de 11,4% em 2021**. 2022. Disponível em: <https://setcesp.org.br/noticias/pib-do-transporte-tem-crescimento-expressivo-de-114-em-2021/>.
- DUINKERKEN, M. B.; EVERS, J. J.; OTTJES, J. A. A simulation model for integrating quay transport and stacking policies on automated container terminals. In: SAN DIEGO, CA, USA. **Proceedings of the 15th european simulation multiconference**. [S.l.], 2001. p. 909–916.
- FENG, B.; YE, Q. Operations management of smart logistics: A literature review and future research. **Frontiers of Engineering Management**, Springer Science and Business Media LLC, v. 8, n. 3, p. 344–355, abr. 2021. Disponível em: <https://doi.org/10.1007/s42524-021-0156-2>.
- GEEKSFORGEEEKS. **Breadth First Search or BFS for a Graph**. 2022. Disponível em: <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>.
- GOVBR. **Balança Comercial - Valores dos Totais**. 2022. Disponível em: https://balanca.economia.gov.br/balanca/pg_principal_bc/principais_resultados.html.
- GUNAWARDHANA, J. A.; PERERA, H. N.; THIBBOTUWAWA, A. Rule-based dynamic container stacking to optimize yard operations at port terminals. **Maritime Transport Research**, Elsevier BV, v. 2, p. 100034, 2021. Disponível em: <https://doi.org/10.1016/j.martra.2021.100034>.
- HUYNH, N. Analysis of container dwell time on marine terminal throughput and rehandling productivity. **Journal of International Logistics and Trade**, Emerald, v. 6, n. 2, p. 69–89, dez. 2008. Disponível em: <https://doi.org/10.24006/jilt.2008.6.2.69>.
- LERSTEAU, C. et al. Solving the problem of stacking goods: Mathematical model, heuristics and a case study in container stacking in ports. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 25330–25343, 2021. Disponível em: <https://doi.org/10.1109/access.2021.3052945>.

LI, W.; XIAONING, Z.; ZHENGYU, X. Efficient container stacking approach to improve handling: efficiency in chinese rail-truck transshipment terminals.

SIMULATION, SAGE Publications, v. 96, n. 1, p. 3–15, abr. 2019. Disponível em: <https://doi.org/10.1177/0037549719843347>.

MA, Y.; KIM, K.-H. A comparative analysis: Various storage rules in container yards and their performances. **Industrial Engineering and Management Systems**, Korean Institute of Industrial Engineers, v. 11, n. 3, p. 276–287, set. 2012. Disponível em: <https://doi.org/10.7232/iems.2012.11.3.276>.

MAGLIĆ, L.; GULIĆ, M.; MAGLIĆ, L. OPTIMIZATION OF CONTAINER RELOCATION OPERATIONS IN PORT CONTAINER TERMINALS. **Transport**, Vilnius Gediminas Technical University, v. 35, n. 1, p. 37–47, dez. 2019. Disponível em: <https://doi.org/10.3846/transport.2019.11628>.

OELSCHLÄGEL, T.; KNUST, S. Solution approaches for storage loading problems with stacking constraints. **Computers & Operations Research**, Elsevier BV, v. 127, p. 105142, mar. 2021. Disponível em: <https://doi.org/10.1016/j.cor.2020.105142>.

REKIK, I.; ELKOSANTINI, S. A multi agent system for the online container stacking in seaport terminals. **Journal of Computational Science**, Elsevier BV, v. 35, p. 12–24, jul. 2019. Disponível em: <https://doi.org/10.1016/j.jocs.2019.06.003>.

REKIK, I.; ELKOSANTINI, S.; CHABCHOUB, H. Container stacking problem: a survey. **CIE45 Proceedings**, p. 10, 2015. Cited By :1. Disponível em: www.scopus.com.

REKIK, I.; ELKOSANTINI, S.; CHABCHOUB, H. Container stacking problem: a survey. In: . [S.l.: s.n.], 2015.

RIES, J.; GONZÁLEZ-RAMÍREZ, R. G.; MIRANDA, P. A fuzzy logic model for the container stacking problem at container terminals. In: **Lecture Notes in Computer Science**. Springer International Publishing, 2014. p. 93–111. Disponível em: https://doi.org/10.1007/978-3-319-11421-7_7.

RONG, J. et al. Stacking position allocation model based on automatic rail crane. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 790, n. 1, p. 012164, mar 2020. Disponível em: <https://dx.doi.org/10.1088/1757-899X/790/1/012164>.

SIHOTANG, J. Analysis of shortest path determination by utilizing breadth first search algorithm. **Jurnal Info Sains : Informatika dan Sains**, Cattleya Darmaya Fortuna, v. 10, n. 2, p. 1–5, set. 2020. Disponível em: <https://doi.org/10.54209/infosains.v10i2.30>.

TING, C.-J.; WU, K.-C. Optimizing container relocation operations at container yards with beam search. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier BV, v. 103, p. 17–31, jul. 2017. Disponível em: <https://doi.org/10.1016/j.tre.2017.04.010>.

UNCTAD. **Review of maritime transport 2019**. 2019. Disponível em: https://unctad.org/en/PublicationsLibrary/rmt2019_en.pdf.

UNCTAD. **Global trade hits record high of \$28.5 trillion in 2021, but likely to be subdued in 2022**. 2022. Disponível em: <https://unctad.org/news/global-trade-hits-record-high-285-trillion-2021-likely-be-subdued-2022>.

WANG, T. **Yard Allocation Problem with Time Dimension at Seaport Terminal**. Tese (Doutorado) — The Hong Kong Polytechnic University, 2021.

WANG, T. et al. A new dynamic shape adjustment and placement algorithm for 3d yard allocation problem with time dimension. **Computers & Operations Research**, Elsevier BV, v. 138, p. 105585, fev. 2022. Disponível em: <https://doi.org/10.1016/j.cor.2021.105585>.

WOO, Y. J.; KIM, K. H. Estimating the space requirement for outbound container inventories in port container terminals. **International Journal of Production Economics**, Elsevier BV, v. 133, n. 1, p. 293–301, set. 2011. Disponível em: <https://doi.org/10.1016/j.ijpe.2010.04.032>.

APÊNDICE A - CRIAÇÃO DE UM AMBIENTE DE IMPLEMENTAÇÃO.

```

patio = Patio(15, 10)
#patio criado a partir da classe Patio
#recebe Xmax igual a 15 e Ymax igual a 10

caracteristicasA = [[0, 1], ["ALFA", "BETA", "GAMA", "XIS"], ["DRY,
                        REEFER"]]
caracteristicasB = [[0], ["ALFA", "BETA", "GAMA", "XIS"], ["DRY"]]
caracteristicasR = [[0, 1], ["ALFA", "BETA", "GAMA", "XIS"], ["REEFER"]]
#array composto pelos arrays referentes ao
#estado, cliente e tipo, respectivamente

blocoA = Bloco(patio, 5, 3, 7, 3, 'A', caracteristicasA)
blocoB = Bloco(patio, 2, 2, 2, 6, 'B', caracteristicasB)
blocoR = Bloco(patio, 3, 1, 11, 8, 'R', caracteristicasR)
#criacao dos blocos a partir da classe Bloco
#recebe o patio que sera inserido, Xmax, Ymax
#x e y de origem, caractere de ID e caracteristicas de aceite

blocos = [blocoA, blocoB, blocoR]
#conjunto de blocos

ativo1 = Ativo(patio, 5, 5, 1)
ativo2 = Ativo(patio, 8, 1, 2)
#tambem recebe o patio assim como as coordenadas em x e y deste patio
#o ultimo parametro eh um numero de identificacao

ativos = [ativo1, ativo2]
#conjunto de ativos

containeres = []
n_containeres = 15
containeres = ClasseAuxiliar.containeres_aleatorios(n_containeres)

#containeres gerados aleatoriamente para teste
#um quadro sera construido para apresentar a estrutura completa do
                        container

estoque = Estoque(patio, 5, 3, 1, 1, containeres)
#criacao do estoque a partir da classe Estoque
#assim como Bloco, recebe o patio que sera inserido, Xmax, Ymax, x e y
                        de origem
#diferentemente, recebe os containeres que serao futuramente alocados

```

```
algoritmo = Algoritmo(patio, blocos, containeres, ativos, estoque)
#criacao do algoritmo a partir da classe Algoritmo
#recebe todas as outras variaveis necessarias para realizar a alocao
#dos containeres

algoritmo.solve(1, 2)
#funcao da classe Algoritmo responsavel por retornar
#a configuracao do patio apos a alocao de containeres
#de acordo com o metodo de alocao numero '1' para a primeira etapa
#e o metodo de alocao numero '2' para a segunda etapa
#os metodos de alocao serao explicados no proximo Capitulo
```

APÊNDICE B - CRIAÇÃO DO AMBIENTE PARA ANÁLISE DE RESULTADOS.

```
patio = Patio(15,10)

caracteristicasA = [[0, 1], ['ALFA', 'BETA', 'GAMA', 'XIS'], ['DRY', 'REEFER']]
caracteristicasB = [[0, 1], ['ALFA', 'BETA', 'GAMA', 'XIS'], ['DRY', 'REEFER']]
caracteristicasC = [[0, 1], ['ALFA', 'BETA', 'GAMA', 'XIS'], ['DRY', 'REEFER']]
caracteristicasR = [[1], ['ALFA', 'BETA', 'GAMA', 'XIS'], ['REEFER']]

blocos = [
blocoA = Bloco(patio, 5, 3, 7, 3, 'A', caracteristicasA)
blocoB = Bloco(patio, 2, 2, 2, 6, 'B', caracteristicasB)
blocoC = Bloco(patio, 4, 2, 5, 7, 'C', caracteristicasC)
blocoR = Bloco(patio, 3, 2, 11, 7, 'R', caracteristicasD)
]

ativos = [
ativo1 = Ativo(patio, 5, 5, 1)
ativo2 = Ativo(patio, 8, 1, 2)
]

containeres = Classe_Auxiliar.carrega_containeres()

estoque = Estoque(patio, 5, 3, 1, 1, containeres)

algoritmo = Algoritmo (patio, blocos, containeres, ativos, estoque)
```