



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MAYRTON DIAS DE QUEIROZ

**Otimização aplicada ao Problema de Sequenciamento de Contêiner em Terminais
Portuários**

Recife

2023

MAYRTON DIAS DE QUEIROZ

**Otimização aplicada ao Problema de Sequenciamento de Contêiner em Terminais
Portuários**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador: Prof. Dr. Ricardo Martins de Abreu Silva

Recife

2023

Catálogo na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

Q3o Queiroz, Mayrton Dias de
Otimização aplicada ao problema de sequenciamento de contêiner em terminais portuários / Mayrton Dias de Queiroz – 2023.
128 f.: il., fig., tab.

Orientador: Ricardo Martins de Abreu Silva.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2023.
Inclui referências, apêndices e anexo.

1. Inteligência computacional. 2. Deslocamento direcionado. 3. SLO+BRKGA. 4. Sublista de operações. 5. Terminais portuários. I. Silva, Ricardo Martins de Abreu (orientador). II. Título

006.31 CDD (23. ed.) UFPE - CCEN 2023 – 117

Mayrton Dias de Queiroz

“Otimização aplicada ao Problema de Sequenciamento de Contêiner em Terminais Portuários”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovado em: 26/04/2023.

Orientador: Prof. Dr. Ricardo Martins de Abreu Silva

BANCA EXAMINADORA

Prof. Dr. Kelvin Lopes Dias
Centro de Informática / UFPE

Prof. Dr. Daniel Carvalho da Cunha
Centro de Informática / UFPE

Prof. Dr. Emerson Alexandre de Oliveira Lima
Escola Politécnica de Pernambuco / UPE

Prof. Dr. Roberto Zanetti Freire
Escola Politécnica / PUC-PR

Prof. Dr. Cícero Garrozi
Departamento de Estatística e Informática / UFRPE

Dedico a meus pais.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por ter sido minha fortaleza e meu refúgio, por me ensinar que existe tempo para tudo, tempo de plantar e tempo para colher.

Meu pai Sandoval Dias pela força e apoio. Minha mãe Maria José por tudo, principalmente por compreender durante essa fase. Minha avó Santina Maria, por sua fonte de sabedoria. Minha irmã Shirley Dias pelo seu apoio. Meu cunhado Vamberto Costa por sua amizade. Meu sobrinho Zico por trazer alegria. Todos meus tios, primos e familiares.

Ao meu orientador Ricardo Martins pelas oportunidades e ensinamentos. Aos professores Roberto Zanetti, Daniel Cunha, Cícero Garrozi, Kelvin Dias e Emerson Lima que participaram da banca avaliadora. Aos professores do CIn-UFPE que ministraram as disciplinas que cursei durante essa jornada. Ao professor Ricardo Prudêncio, coordenador da pós-graduação. Aos servidores da pós-graduação, Maria Lilia, Socorro e Daniel, por toda ajuda nos processos acadêmicos.

A professora Natasha Lino por todo incentivo e dedicação. Aos professores Gustavo Motta e Claurton Siebra por me ajudar durante o processo de seleção. A professora Maria Helena pelas oportunidades oferecidas e mesmo sem saber, por seus conselhos que levo até os dias atuais, aos professores, amigos e funcionários da Escola do Legislativo.

Meus amigos Vitor Soares, Carlos Magno, Felipe Tiago, Ruan Palmeira, Augusto César, Felipe Crispim, Hozana Raquel, Joanacelle, Roberto Freire, Elaine Marques, Alisson Silva e Emanuel Nascimento. Agradeço a todos que participaram dessa caminhada, que por ventura não esteja citado nas linhas anteriores.

RESUMO

Com a utilização de contêineres em terminais portuários, surge o desafio de manipulá-los com intuito de obter um tempo menor para realização das operações. Entre os problemas encontrados nesse contexto, é possível destacar o Problema de Sequenciamento de Contêiner que busca determinar uma sequência de operações com os contêineres realizada entre o navio e o pátio através do guindaste. Diante dessa problemática, o objetivo deste trabalho consiste em propor um algoritmo capaz de encontrar uma sequência de operações com os contêineres para serem realizadas pelo guindaste e que possua um tempo total reduzido. Neste trabalho, foi reimplementado o algoritmo GRASP encontrado na literatura, em seguida foi desenvolvido o algoritmo de Deslocamento Direcionado combinado ao GRASP. Foi possível desenvolver o algoritmo de SubLista de Operações (SLO) com o intuito de combinar os contêineres da solução encontrada, logo após, o mesmo que foi combinado com o Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA) para encontrar uma sequência de exploração das pilhas da baía de entrada. Ao realizar os experimentos com o conjunto de instâncias obtido na literatura, foi possível observar que em 34% das instâncias, os resultados do algoritmo SLO+BRKGA tiveram um tempo total de realização das operações menor do que os resultados obtidos pelos algoritmos analisados nesse trabalho. Adicionalmente, foi desenvolvido um *framework* para auxiliar os especialistas na construção de soluções. Através do *framework*, foi possível visualizar e analisar as soluções geradas, o que permitiu verificar o impacto das soluções aplicadas ao problema.

Palavras-chave: deslocamento direcionado; SLO+BRKGA; sublista de operações; problema de sequenciamento de contêiner; terminais portuários.

ABSTRACT

With the use of containers in port terminals, the challenge of handling them arises in order to obtain a shorter time to carry out the operations. Among the problems encountered in this context, it is possible to highlight the Container Sequencing Problem, which seeks to determine a sequence of operations with containers performed between the ship and the yard using the crane. Faced with this problem, the objective of this work is to propose an algorithm capable of finding a sequence of operations with the containers to be performed by the crane and that has a reduced total time. In this work, the GRASP algorithm found in the literature was reimplemented, then the Directed Displacement algorithm combined with GRASP was developed. It was possible to develop the Operations SubList (SLO) algorithm in order to combine the containers of the solution found, the same one that was combined with the BRKGA algorithm to find a sequence of exploration of the stacks of the input bay. When carrying out the experiments with the set of instances obtained in the literature, it was possible to observe that in 33% of the instances, the results of the SLO+BRKGA algorithm had a total time to perform the operations smaller than the results obtained by the algorithms analyzed in this work . Additionally, a framework was developed to help specialists build solutions. Through framework, it was possible to visualize and analyze the generated solutions, which allowed verifying the impact of the solutions applied to the problem.

Keywords: container sequencing problem; directed displacement; operations subList; SLO+BRKGA; port terminals.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Exemplo de um contêiner. | 19 |
| Figura 2 – Visão geral do guindaste, das regiões disponíveis para realização das operações e a Configuração Inicial e Configuração de Partida. | 21 |
| Figura 3 – Exemplo de uma operação (VY) com um contêiner de importação. | 22 |
| Figura 4 – Exemplo de uma operação (YV) com um contêiner de exportação. | 23 |
| Figura 5 – Exemplo de uma operação (VB) com um contêiner de <i>reshuffle</i> | 24 |
| Figura 6 – Exemplo de uma operação (BV) com um contêiner de <i>reshuffle</i> | 24 |
| Figura 7 – Exemplo de uma operação (VV) com um contêiner de <i>reshuffle</i> | 25 |
| Figura 8 – Exemplo de uma Configuração Inicial e de Partida de uma instância. | 27 |
| Figura 9 – Exemplo de uma solução com 8 operações. | 28 |
| Figura 10 – Exemplo de aplicação das equações 2.22, 2.23e 2.24 durante a execução da função <i>selecionar()</i> | 36 |
| Figura 11 – Exemplo da geração de novas soluções ao deslocar a operação 4 durante a execução da função <i>mover()</i> | 38 |
| Figura 12 – Exemplo da geração de novas soluções por meio das transformações durante a função <i>transformar()</i> | 38 |
| Figura 13 – Resultado da Questão de Pesquisa 1. | 47 |
| Figura 14 – Resultado da Questão de Pesquisa 2. | 49 |
| Figura 15 – Resultado da Questão de Pesquisa 3. | 49 |
| Figura 16 – Resultado da Questão de Pesquisa 4. | 50 |
| Figura 17 – Resultado da Questão de Pesquisa 5. | 50 |
| Figura 18 – Passos realizados durante o desenvolvimento desse trabalho. | 52 |
| Figura 19 – Quantidade de artigos obtidos após a pesquisa nas Bases de Dados, o <i>Download</i> e aplicação dos Critérios de Inclusão e Exclusão. | 53 |
| Figura 20 – Exemplo de construção de sequência de contêineres de cada pilha do Algoritmo de Sub-Listas de Operações. | 57 |
| Figura 21 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VY da Pilha 1. | 62 |
| Figura 22 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação YV da Pilha 3. | 63 |

| | |
|--|----|
| Figura 23 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VV com as Pilhas 4 e 3. | 64 |
| Figura 24 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VY da Pilha 4. | 64 |
| Figura 25 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VV com as Pilhas 2 e 1. | 65 |
| Figura 26 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VV com as Pilhas 2 e 1. | 65 |
| Figura 27 – Exemplo do decode do BRKGA. | 66 |
| Figura 28 – Visão geral da comunicação do <i>framework</i> com os demais problemas e sistemas legados. | 68 |
| Figura 29 – Box plot do tempo de realização das operações do SLO com $n = 10$ | 75 |
| Figura 30 – Box plot do tempo de execução do SLO com $n = 10$ | 75 |
| Figura 31 – Box plot do tempo de realização das operações do SLO com $n = 15$ | 76 |
| Figura 32 – Box plot do tempo de execução do SLO com $n = 15$ | 76 |
| Figura 33 – Box plot do tempo de realização das operações do SLO com $n = 20$ | 77 |
| Figura 34 – Box plot do tempo de execução do SLO com $n = 20$ | 77 |
| Figura 35 – Box plot do tempo de realização das operações do SLO+BRKGA com $n = 10$ | 78 |
| Figura 36 – Box plot do tempo de execução do SLO+BRKGA com $n = 10$ | 78 |
| Figura 37 – Box plot do tempo de realização das operações do SLO+BRKGA com $n = 15$ | 79 |
| Figura 38 – Box plot do tempo de execução do SLO+BRKGA com $n = 15$ | 79 |
| Figura 39 – Box plot do tempo de realização das operações do SLO+BRKGA com $n = 20$ | 80 |
| Figura 40 – Box plot do tempo de execução do SLO+BRKGA com $n = 20$ | 80 |
| Figura 41 – Tela inicial e menu de opções do <i>framework</i> | 82 |
| Figura 42 – Tela de visualização do arquivo de configuração. | 82 |
| Figura 43 – Telas do <i>Framework</i> apresentando cada operação da solução encontrada. | 84 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Tempo necessário para realização de uma operação t em segundos. | 26 |
| Tabela 2 – Tempo necessário para movimentação do guindaste sem contêiner da operação t até a operação u em segundos. | 26 |
| Tabela 3 – Tempo gasto em cada operação da solução. | 27 |
| Tabela 4 – Bases de dados adotadas. | 45 |
| Tabela 5 – Quantidade de artigos em cada base de dados. | 46 |
| Tabela 6 – Identificação de cada alternativa das Questões de Pesquisa de cada artigo. | 51 |
| Tabela 7 – Quantidade das instâncias do conjunto obtido na literatura. | 54 |
| Tabela 8 – Resultados obtidos com a realização dos experimentos com os algoritmos. | 73 |
| Tabela 9 – Tempo gasto em cada operação da solução. | 83 |
| Tabela 10 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R00_001 e CSP_n10_I40_E70_R04_010. | 90 |
| Tabela 11 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R06_001 e CSP_n10_I40_E70_R10_010. | 91 |
| Tabela 12 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R12_001 e CSP_n10_I40_E70_R16_010. | 92 |
| Tabela 13 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R18_001 e CSP_n10_I70_E40_R00_010. | 93 |
| Tabela 14 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R02_001 e CSP_n10_I70_E40_R06_010. | 94 |
| Tabela 15 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R08_001 e CSP_n10_I70_E40_R12_010. | 95 |
| Tabela 16 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R14_001 e CSP_n10_I70_E40_R18_010. | 96 |
| Tabela 17 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R20_001 e CSP_n10_I70_E70_R02_010. | 97 |
| Tabela 18 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E70_R04_001 e CSP_n10_I70_E70_R08_010. | 98 |
| Tabela 19 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E70_R10_001 e CSP_n10_I70_E70_R14_010. | 99 |

| | |
|--|-----|
| Tabela 20 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E70_R16_001 e CSP_n10_I70_E70_R20_010. | 100 |
| Tabela 21 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R00_001 e CSP_n15_I40_E70_R04_010. | 101 |
| Tabela 22 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R06_001 e CSP_n15_I40_E70_R10_010. | 102 |
| Tabela 23 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R12_001 e CSP_n15_I40_E70_R16_010. | 103 |
| Tabela 24 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R18_001 e CSP_n15_I70_E40_R00_010. | 104 |
| Tabela 25 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R02_001 e CSP_n15_I70_E40_R06_010. | 105 |
| Tabela 26 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R08_001 e CSP_n15_I70_E40_R12_010. | 106 |
| Tabela 27 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R14_001 e CSP_n15_I70_E40_R18_010. | 107 |
| Tabela 28 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R20_001 e CSP_n15_I70_E70_R02_010. | 108 |
| Tabela 29 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E70_R04_001 e CSP_n15_I70_E70_R08_010. | 109 |
| Tabela 30 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E70_R10_001 e CSP_n15_I70_E70_R14_010. | 110 |
| Tabela 31 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E70_R16_001 e CSP_n15_I70_E70_R20_010. | 111 |
| Tabela 32 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R00_001 e CSP_n20_I40_E70_R04_010. | 112 |
| Tabela 33 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R06_001 e CSP_n20_I40_E70_R10_010. | 113 |
| Tabela 34 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R12_001 e CSP_n20_I40_E70_R16_010. | 114 |
| Tabela 35 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R18_001 e CSP_n20_I70_E40_R00_010. | 115 |

| | |
|--|-----|
| Tabela 36 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R02_001 e CSP_n20_I70_E40_R06_010. | 116 |
| Tabela 37 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R08_001 e CSP_n20_I70_E40_R12_010. | 117 |
| Tabela 38 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R14_001 e CSP_n20_I70_E40_R18_010. | 118 |
| Tabela 39 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R20_001 e CSP_n20_I70_E70_R02_010. | 119 |
| Tabela 40 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E70_R04_001 e CSP_n20_I70_E70_R08_010. | 120 |
| Tabela 41 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E70_R10_001 e CSP_n20_I70_E70_R14_010. | 121 |
| Tabela 42 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E70_R16_001 e CSP_n20_I70_E70_R20_010. | 122 |
| Tabela 43 – Resultados obtidos do SLO+BRKGA entre as instâncias CSP_n10_I40_E70_R06_003 e CSP_n15_I40_E70_R06_009. | 123 |
| Tabela 44 – Resultados obtidos do SLO+BRKGA entre as instâncias CSP_n15_I40_E70_R12_001 e CSP_n20_I40_E70_R10_010. | 124 |
| Tabela 45 – Resultados obtidos do SLO+BRKGA entre as instâncias CSP_n20_I40_E70_R16_001 e CSP_n20_I70_E70_R14_010. | 125 |

SUMÁRIO

| | | |
|--------------|--|-----------|
| 1 | INTRODUÇÃO | 15 |
| 1.1 | CONTEXTUALIZAÇÃO | 15 |
| 1.2 | OBJETIVOS | 16 |
| 1.2.1 | Objetivo Geral | 16 |
| 1.2.2 | Objetivos Específicos | 16 |
| 1.3 | JUSTIFICATIVA | 17 |
| 1.4 | PUBLICAÇÃO RELACIONADA A PESQUISA | 17 |
| 1.5 | ORGANIZAÇÃO | 18 |
| 2 | REFERENCIAL TEÓRICO | 19 |
| 2.1 | INTRODUÇÃO | 19 |
| 2.2 | PROBLEMA DE SEQUENCIAMENTO DE CONTÊNER | 20 |
| 2.3 | FORMULAÇÃO MATEMÁTICA | 29 |
| 2.4 | META-HEURÍSTICA | 32 |
| 2.4.1 | GRASP | 33 |
| 2.4.1.1 | <i>GRASP aplicado ao Problema de Sequenciamento de Contêiner</i> | 33 |
| 2.4.2 | Algoritmo Genético | 39 |
| 2.4.2.1 | <i>Algoritmo Genético de Chaves Aleatórias Viciadas</i> | 41 |
| 2.5 | CONSIDERAÇÕES FINAIS | 42 |
| 3 | REVISÃO DA LITERATURA | 43 |
| 3.1 | CONTEXTUALIZAÇÃO | 43 |
| 3.2 | PLANEJAMENTO | 43 |
| 3.2.1 | Definição das Questões de Pesquisa | 43 |
| 3.2.2 | Definição da string de busca | 45 |
| 3.2.3 | Definição das bases de dados | 45 |
| 3.2.4 | Definição dos critérios de inclusão e exclusão | 45 |
| 3.3 | CONDUÇÃO | 46 |
| 3.4 | RESULTADOS | 47 |
| 3.5 | CONSIDERAÇÕES FINAIS | 51 |
| 4 | METODOLOGIA | 52 |

| | | |
|-------|---|------------|
| 4.1 | CONTEXTUALIZAÇÃO | 52 |
| 4.2 | PASSO 1 - REVISÃO DA LITERATURA | 52 |
| 4.3 | PASSO 2 - OBTENÇÃO DAS INSTÂNCIAS | 53 |
| 4.4 | PASSO 3 - DEFINIÇÃO DOS ALGORITMOS | 54 |
| 4.4.1 | Algoritmo de Deslocamento Direcionado | 55 |
| 4.4.2 | Algoritmo de Sub-Listas de Operações | 56 |
| 4.4.3 | Algoritmo de Sub-Listas de Operações + BRKGA | 66 |
| 4.5 | DEFINIÇÃO DO <i>FRAMEWORK</i> | 67 |
| 4.5.1 | Passos para realização de um experimento | 67 |
| 4.5.2 | Descrição conceitual do <i>framework</i> | 68 |
| 4.6 | ANÁLISE DOS RESULTADOS | 69 |
| 4.7 | CONSIDERAÇÕES FINAIS | 69 |
| 5 | RESULTADOS COMPUTACIONAIS | 71 |
| 5.1 | CONFIGURAÇÃO DO AMBIENTE | 71 |
| 5.2 | EXECUÇÃO DOS ALGORITMOS | 71 |
| 5.3 | EXEMPLO DE UMA SOLUÇÃO NO <i>FRAMEWORK</i> | 81 |
| 5.4 | CONSIDERAÇÕES FINAIS | 84 |
| 6 | CONCLUSÃO | 86 |
| | REFERÊNCIAS | 88 |
| | APÊNDICE A – RESULTADOS COMPUTACIONAIS | 90 |
| | APÊNDICE B – ARQUIVOS DO EXEMPLO DO <i>FRAMEWORK</i> . | 126 |
| | ANEXO A – DADOS DA INSTÂNCIA PARA CSP | 127 |

1 INTRODUÇÃO

Nesse capítulo será descrito a contextualização, os objetivos e a estrutura do trabalho.

1.1 CONTEXTUALIZAÇÃO

Conforme os dados disponibilizados no portal Comex Stat¹, em agosto de 2021 o indicador mensal de comércio exterior do Brasil teve um aumento quando comparado ao mesmo mês do ano de 2020. O crescimento das exportações foi de 68,7% do valor obtido e de 50,5% do peso líquido, com relação às importações, o aumento foi de 56,2% do valor gasto e 5,1% do peso líquido. Diante da demanda para transportar produtos e matérias-primas, se fez necessária uma alternativa para agilizar o manuseio dos mesmos durante o trajeto, sendo essa, a adoção de contêineres com o intuito de padronizar a carga, facilitando o seu transporte e evitando que seu conteúdo seja extraviado.

Com a utilização de contêineres e o aumento do fluxo de cargas, surge o desafio de realizar as operações em um tempo menor, a fim de conter gastos e diminuir o tempo de transporte das cargas. No que se refere aos problemas presentes nos terminais portuários, é possível destacar o Problema de Sequenciamento de Contêiner (do inglês, *Container Sequencing Problem - CSP*), que consiste em encontrar um conjunto de operações que o guindaste realiza com os contêineres, de forma que, se inicia de uma configuração inicial até obter a configuração de partida. Ao analisar esse problema, nota-se que ele possui diversas possibilidades de operações com os contêineres e, segundo Lee, Liu e Chu (2015), o problema pertence à classe NP-Difícil.

No trabalho de Meisel e Wichmann (2010), os autores utilizaram Programação Linear, um método exato para encontrar uma solução ótima do CSP. Conforme o aumento do tamanho da instância de entrada e a quantidade de variáveis consideradas, o tempo gasto para solucioná-lo aumenta consideravelmente. Então, uma alternativa possível nesse cenário é a adoção de heurísticas para encontrar soluções viáveis para o problema. Adicionalmente, os mesmos autores propuseram o uso da meta-heurística GRASP (do inglês, *Greedy Randomized Adaptive Search Procedure*) aplicada ao CSP, no entanto, é importante investigar outras heurísticas para aplicar ao CSP com o intuito de encontrar um conjunto de operações com os contêineres e que possua um menor tempo para serem realizadas pelo guindaste.

¹ Comex Stat. Portal para acesso gratuito às estatísticas de comércio exterior do Brasil. Disponível em: <<http://comexstat.mdic.gov.br/pt/home>>

Durante a realização deste trabalho foi possível desenvolver algoritmos e executar os mesmos em um conjunto de instâncias adotado na literatura (MEISEL; WICHMANN, 2010). Nesse trabalho, foi proposto o Algoritmo de Deslocamento de Direcionado que busca aumentar o número de operações internas no navio, evitando o tempo de inseri-lo em alguma região temporária. Também foi proposto o Algoritmo de Sub-Listas de Operações(SLO), que busca combinar as operações com os contêineres de forma que a região que o guindaste se encontra, ao finalizar uma operação, seja a mesma para iniciar a próxima operação. Adicionalmente, foi proposto um algoritmo que combina o SLO e o Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA), visto que o BRKGA foi proposto para identificar uma sequência de exploração das pilhas de contêineres, assim combinado com o SLO, isto é, o Algoritmo SLO+BRKGA, foi possível obter uma solução que reduz o tempo de realização das operações com os contêineres em 34% das instâncias analisadas.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Diante desse cenário, este trabalho tem como objetivo propor um algoritmo que seja capaz de encontrar uma sequência de operações com os contêineres a ser realizada pelo guindaste com menor tempo possível para o Problema de Sequenciamento de Contêiner.

1.2.2 Objetivos Específicos

- Obter as principais técnicas encontradas nos artigos da literatura que abordam o Problema de Sequenciamento de Contêiner;
- Identificar as operações que o guindaste realiza com o contêiner para o CSP;
- Propor um método capaz de encontrar uma sequência de operações com os contêineres que serão realizadas pelo guindaste, apresentando uma diminuição no tempo total;
- Identificar uma forma para visualizar e verificar se a sequência de operações encontrada é viável.

1.3 JUSTIFICATIVA

Ao observar a cadeia logística responsável pelo transporte de produtos e matérias-primas, é importante destacar que o modal marítimo consegue transportar uma quantidade maior de produtos e com um custo menor, quando comparado ao modal aéreo. Segundo Silva et al. (2016), “o setor portuário brasileiro é responsável por movimentar 96% do volume do comércio internacional do País”. De forma que a adoção do modal marítimo vem crescendo gradativamente (FIRMINO, 2019).

Diante deste contexto, se faz necessário encontrar soluções que diminuam o tempo de realização das operações. Sendo assim, nesse trabalho busca-se encontrar soluções para o problema de sequenciamento de contêiner, de forma que a sequência de operações com os contêineres realizadas pelo guindaste seja reduzida. Assim, espera-se obter os seguintes benefícios:

- Atender uma quantidade maior de navios;
- Diminuição no tempo total de transporte das mercadorias;
- Os operadores dos guindastes podem operar mais contêineres durante a mesma jornada de trabalho;
- Economia no gasto de energia elétrica e combustíveis, tanto com relação aos equipamentos, bem como os veículos que transitam no terminal portuário;
- Diminuição do tempo de espera do navio e dos veículos presentes no pátio, tornando o modal marítimo mais competitivo.

1.4 PUBLICAÇÃO RELACIONADA A PESQUISA

Durante a realização deste trabalho foi possível realizar a seguinte publicação:

QUEIROZ, Mayrton Dias de; SILVA, Ricardo Martins de Abreu. Framework para apoiar especialistas no Problema de Sequenciamento de Contêiner em terminais portuários. **Revista Principia - Divulgação Científica e Tecnológica do IFPB**, João Pessoa, jun. 2022. ISSN 2447-9187. Disponível em: <<https://periodicos.ifpb.edu.br/index.php/principia/article/view/6730>>. Acesso em: 27 Jul. 2023. doi:<<http://dx.doi.org/10.18265/1517-0306a2022id6730>>.

1.5 ORGANIZAÇÃO

Nessa seção será descrita a divisão deste documento:

Capítulo 1: nesse capítulo foi abordado a contextualização do trabalho, o objetivo geral e objetivos específicos.

Capítulo 2: no capítulo do Referencial Teórico, será apresentada uma introdução com alguns problemas encontrados nos terminais portuários, em seguida será descrito o Problema de Sequenciamento de Contêiner, as principais regiões utilizadas nesse problema, as principais operações com os contêineres, bem como, um exemplo com valores de entrada e uma possível solução para uma instância apresentada como exemplo. Em seguida, será feita uma descrição da meta-heurística GRASP, Algoritmo Genético e Algoritmo Genético de Chaves Aleatórias Viciadas.

Capítulo 3: neste capítulo será descrita a Revisão da Literatura alinhada ao Problema de Sequenciamento de Contêiner que foi realizada nesse trabalho, descrevendo as fases realizadas, ou seja, o Planejamento, a Condução e os Resultados.

Capítulo 4: se refere a Metodologia do trabalho, onde será descrito cada passo realizado para o desenvolvimento deste trabalho, ou seja, a revisão da literatura, a obtenção das instâncias de entrada, a definição dos algoritmos, a definição do *framework* e a comparação dos resultados obtidos.

Capítulo 5: neste capítulo dos Resultados Computacionais, são apresentadas a configuração do ambiente onde os experimentos foram realizados, a descrição da execução dos algoritmos implementados e um exemplo de uma solução no *framework*.

Capítulo 6: neste capítulo de Conclusão foram descritas as considerações finais e caminhos para realização dos trabalhos futuros.

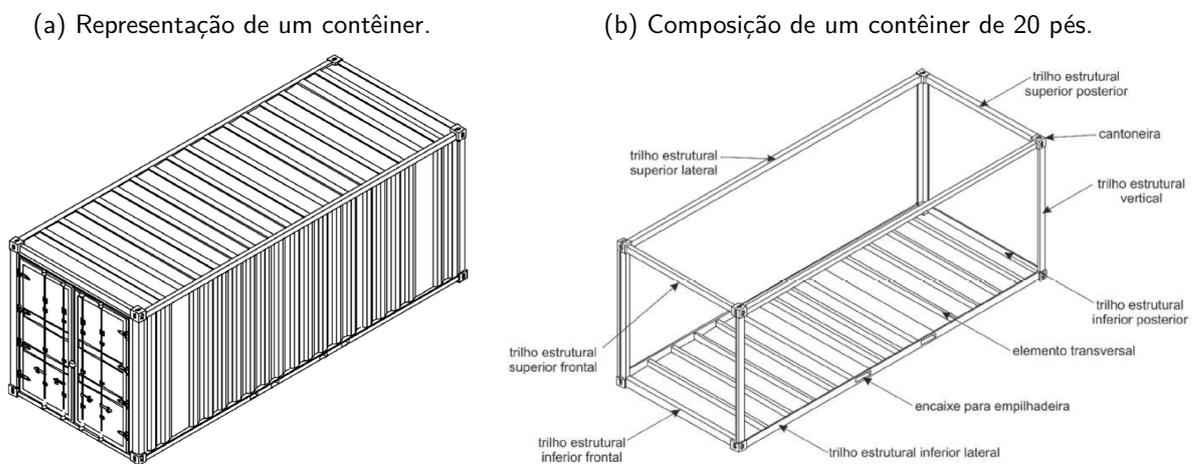
2 REFERENCIAL TEÓRICO

Neste capítulo será feita uma breve descrição dos problemas relacionados aos terminais portuários, a descrição sobre o Problema de Sequenciamento de Contêiner e uma descrição das meta-heurísticas GRASP, Algoritmos Genético e Algoritmo Genético de Chaves Aleatórias Viciadas.

2.1 INTRODUÇÃO

Por volta de 1937 surgiu a proposta de utilizar os contêineres com o intuito de padronizar o espaço para inserir os produtos e as matérias-primas a serem transportados, evitando que fossem extraviados. Os contêineres podem ter o tamanho de 20 ou 40 pés, e com relação ao formato, na Figura 1 é possível observar a representação de um contêiner comumente usado, mas que pode variar caso o conteúdo a ser inserido seja grãos ou líquidos.

Figura 1 – Exemplo de um contêiner.



Fonte: ISO 6346 (1972)

Com a adoção dos contêineres, foi possível obter um maior controle com relação às operações que podem ser realizadas em terminais portuários, visto que as dimensões são conhecidas, facilitando no planejamento do espaço que cada contêiner irá ocupar. Dessa forma, é importante destacar que existem diversos problemas presentes nos terminais portuários, desde do embarque dos contêineres de exportação recebidos pelos veículos para serem inseridos no navio, bem como o desembarque dos contêineres de importação retirados do navio até serem colocados no veículo, seja um caminhão ou trem.

No trabalho de Azevedo, Chaves e Neto (2014), os autores analisaram os trabalhos de Guan, Yang e Zhou (2013), Steenken, Voß e Stahlbock (2004) e Stahlbock e Voß (2008), de forma que os autores identificaram que as operações em um terminal de contêiner podem ser divididas em cinco problemas principais:

1. Alocação de Berços
2. Plano de Estiva
3. Atribuição e *Scheduling* de Guindastes Portuários
4. Transporte no Cais
5. Transporte no Pátio

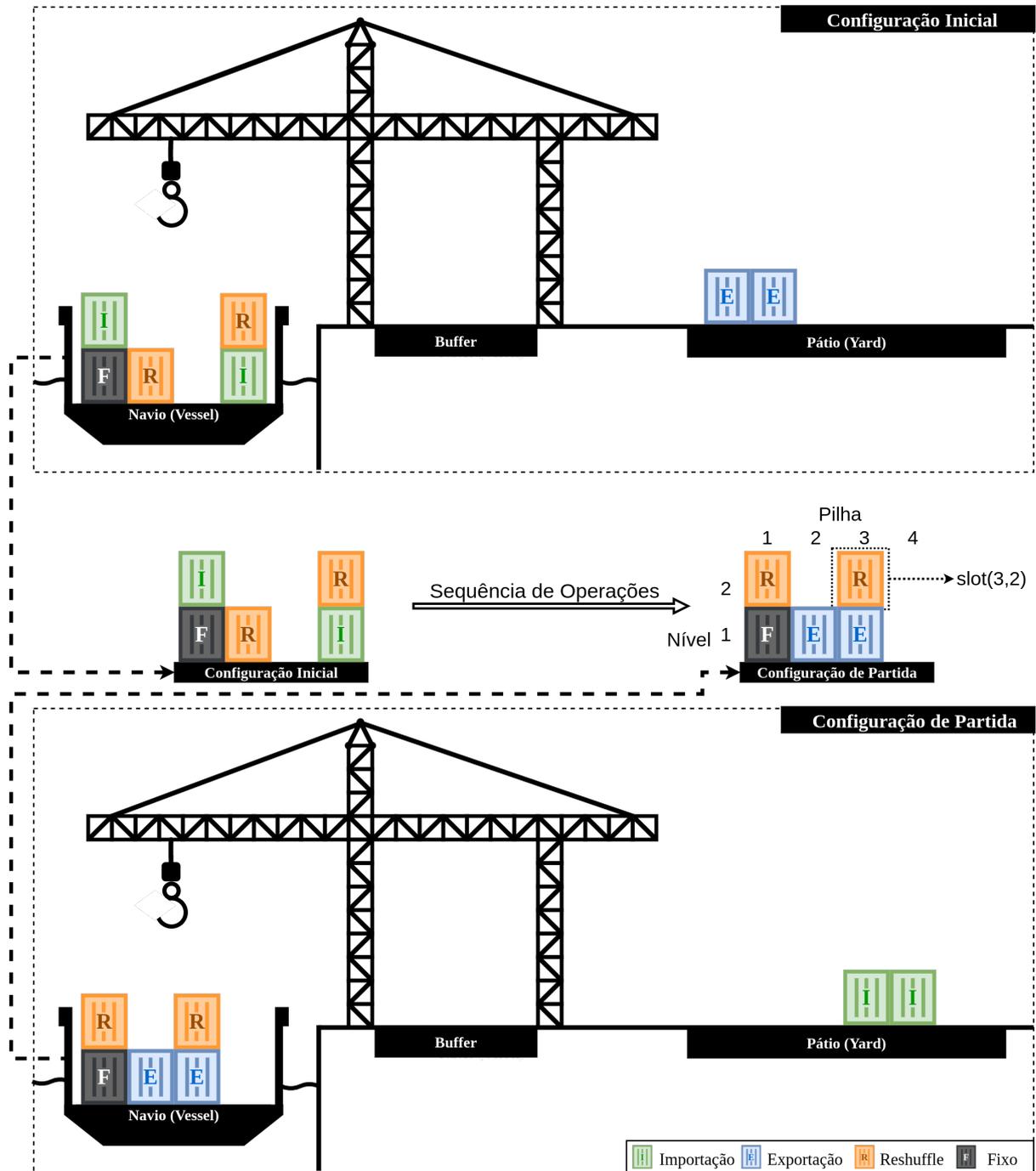
Dentre esses cinco problemas citados, será destacado o problema de Plano de Estiva (do inglês, *Stowage Planning*), visto que, esse problema está alinhado a essa pesquisa. O problema de Plano de Estiva contempla as operações realizadas com os contêineres entre o navio e o berço do porto utilizando os guindastes (AVRIEL et al., 1998) e (BARBOSA; CARDOSO, 2018). Diante desse cenário é possível encontrar diversas variações desse problema, como a adoção de múltiplos guindastes, múltiplos navios, equilíbrio do navio, múltiplas baias, guindastes com duplo ciclo, entre outras. Dentre eles, esse trabalho abordará o Problema de Sequenciamento de Contêiner, que será descrito na Subseção 2.2.

2.2 PROBLEMA DE SEQUENCIAMENTO DE CONTÊINER

O Problema de Sequenciamento de Contêiner (do inglês, *Container Sequencing Problem* - CSP), consiste em encontrar uma sequência de operações com os contêineres partindo de uma configuração inicial (do inglês, *Arrival Configuration* - AC) até obter a configuração de partida (do inglês, *Departure Configuration* - DC). Na Figura 2, é possível ter uma visão geral das principais áreas disponíveis para a realização das operações com o guindaste. O navio (do inglês, *vessel*), possui diversos contêineres organizados por baias (do inglês, *bay*). Com relação ao pátio (do inglês, *yard*) é importante destacar que nesta região transitam os veículos, auxiliando no recebimento e disponibilização dos contêineres, conforme a operação que os guindastes estão realizando. A região do *buffer* é utilizada para realizar operações com

o contêiner de forma temporária, ou seja, é possível colocar um contêiner nessa região, realizar outras operações e depois recuperar o contêiner colocando-o no local desejado.

Figura 2 – Visão geral do guindaste, das regiões disponíveis para realização das operações e a Configuração Inicial e Configuração de Partida.



Fonte: Elaborada pelo autor (2023)

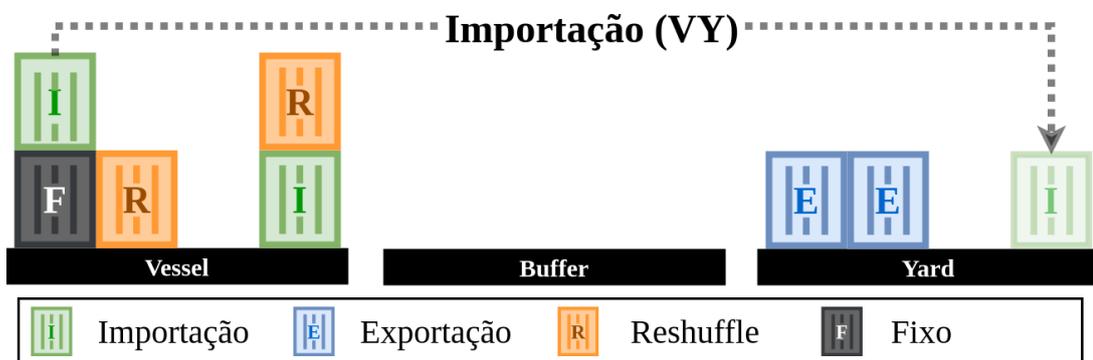
As operações realizadas pelo guindaste consideram uma baía composta por m pilhas e n níveis. A configuração inicial e de partida, representam o estado desejado para a baía antes e depois das operações, respectivamente. Na Figura 2, é possível observar um exemplo de

configuração inicial e de partida, nesse exemplo, nota-se que uma configuração é formada por 4 pilhas e 2 níveis. Cada contêiner possui uma posição, que é formada pelo par ordenado $slot(i, j)$, sendo $1 < i \leq m$ e $1 < j \leq n$.

Com relação às operações que o guindaste pode realizar com cada tipo de contêiner é possível destacar:

Importação: O contêiner de importação contém o conteúdo que será desembarcado, sendo assim, a operação que será realizada com esse contêiner irá retirá-lo do *Vessel* e inseri-lo no *Yard*. Na Figura 3, é possível observar a operação realizada com o contêiner de importação, representado pela letra I. Para realizar essa operação o contêiner precisa estar localizado no topo da pilha e não possuir nenhum outro contêiner com o nível superior na mesma pilha. No exemplo apresentado na figura, nota-se esse cenário ocorrendo com o contêiner localizado no $slot(1, 2)$. Já o contêiner localizado no $slot(4, 1)$ só estará pronto para realizar a operação após a remoção do contêiner do $slot(4, 2)$.

Figura 3 – Exemplo de uma operação (VY) com um contêiner de importação.



Fonte: Elaborada pelo autor (2022)

Exportação: O contêiner de exportação possui o conteúdo que será embarcado, de forma que esse contêiner será recuperado do *Yard* e transferido pelo guindaste para o *Vessel* no $slot$ desejado, conforme a configuração de partida. Na Figura 4, é possível observar um exemplo de movimentação com o contêiner de exportação, representado pela letra E. Para a operação de exportação ser realizada é importante que os contêineres que pertencem à mesma pilha e com o valor de nível inferior do $slot$ a ser inserido, já tenham realizado as operações necessárias, ou seja, todos os contêineres de importação já tenham sido retirados e todos os contêineres de nível inferior estejam conforme a configuração de partida. De posse do $slot$ livre do *Vessel* é possível realizar a operação

do contêiner, no exemplo da figura nota-se que o $slot(3, 1)$ está livre, então o contêiner de exportação do *Yard* pode ser movimentado.

Figura 4 – Exemplo de uma operação (YV) com um contêiner de exportação.



Fonte: Elaborada pelo autor (2022)

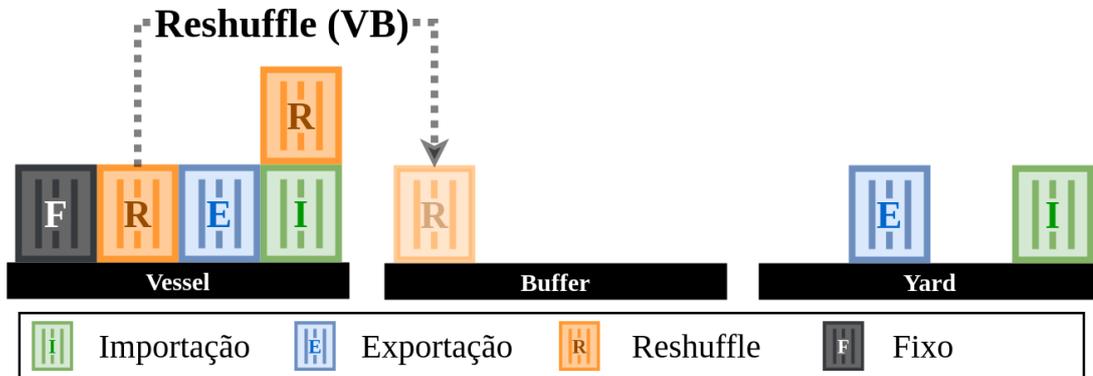
Reshuffle: O contêiner de *reshuffle* possui o conteúdo que será mantido no *Vessel*, ou seja, esse contêiner estará presente tanto na configuração inicial como na configuração de partida. É possível destacar três operações que podem ser realizadas com esse tipo de contêiner:

Inserir no *Buffer*: O *Buffer* é uma região que pode ser utilizada para inserir os contêineres de *reshuffle* enquanto outras operações são realizadas, por exemplo, a operação de importação com os contêineres que estão na mesma pilha com níveis inferiores. Na Figura 5, é possível observar um exemplo de operação com o contêiner de *reshuffle*, representado pela letra R. Nesse exemplo, é possível notar o contêiner que se encontra no $slot(2, 1)$ no *Vessel* será removido para o *buffer*, pois dessa forma o *slot* ficará livre para inserção de outro contêiner. Já o contêiner do $slot(4, 2)$ precisa realizar uma operação, pois ele se encontra acima de um contêiner de importação, que só poderá ser transferido para o *Yard* após todos os *slots* de nível superior estiverem livres.

Retirar do *Buffer*: Outra operação possível com o contêiner de *reshuffle*, é a remoção de um contêiner do *Buffer* para o *Vessel*, conforme pode ser visto na Figura 6. Após a realização de todas as operações com os contêineres de nível inferior, é possível movimentar o contêiner de *reshuffle* de volta ao *Vessel*. No exemplo da figura nota-se que o contêiner será retirado do *Buffer* e será inserido no $slot(3, 2)$ que se encontra livre e corresponde à configuração de partida desejada. É importante

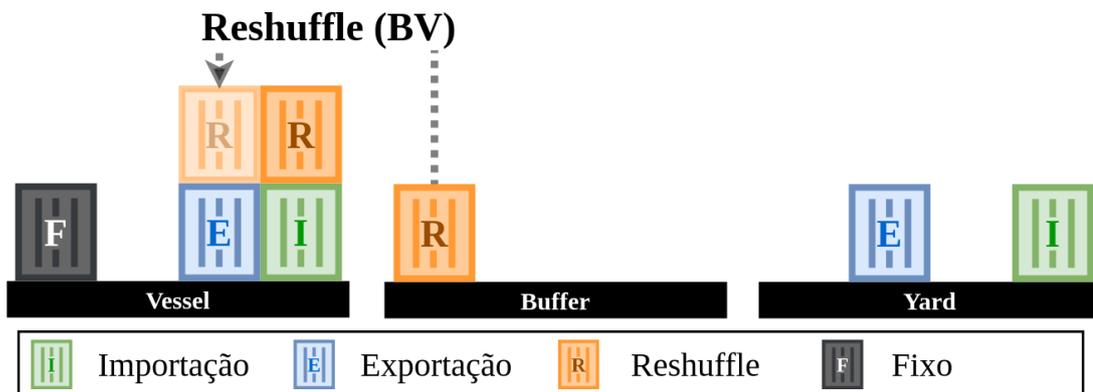
destacar que antes da primeira operação o *Buffer* estava vazio, e após todas as operações, ou seja, chegando à configuração de partida, nenhum contêiner de *reshuffle* deve permanecer no *Buffer*.

Figura 5 – Exemplo de uma operação (VB) com um contêiner de *reshuffle*.



Fonte: Elaborada pelo autor (2022)

Figura 6 – Exemplo de uma operação (BV) com um contêiner de *reshuffle*.

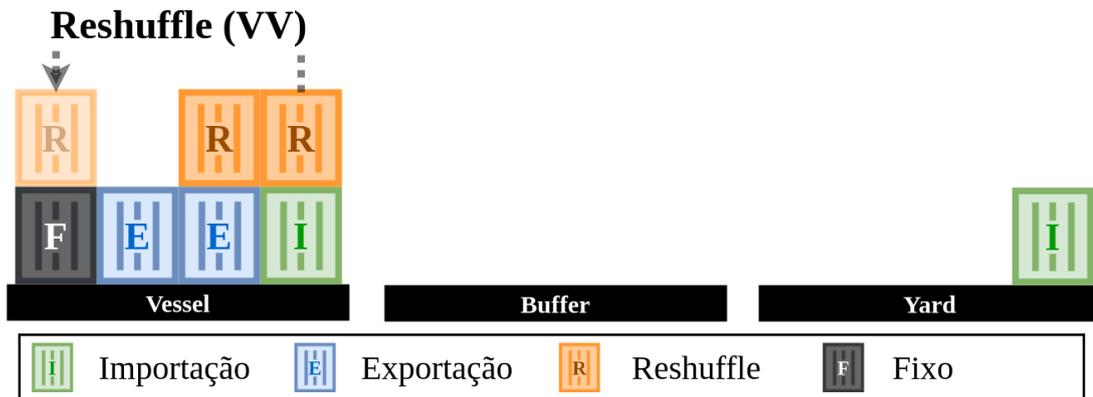


Fonte: Elaborada pelo autor (2022)

Movimentação interna: A movimentação interna envolve o contêiner de *reshuffle* no *Vessel*, ou seja, um contêiner que está em um *slot* será movimentado para outro *slot* que já esteja pronto para receber esse contêiner e esteja conforme a configuração de partida. Na Figura 7, é possível notar um exemplo de um contêiner de *reshuffle* que está sendo retirado do *slot*(4, 2) para ser inserido no *slot*(1, 2).

Fixo: O contêiner Fixo, representado pela letra F, não está envolvido nas operações de movimentação, de forma que ele permanece no mesmo *slot* tanto na configuração de chegada como na configuração de partida e não é movimentado pelas operações.

Figura 7 – Exemplo de uma operação (VV) com um contêiner de *reshuffle*.



Fonte: Elaborada pelo autor (2022)

Esses são os tipos de contêineres e as principais operações realizadas pelo guindaste no Problema de Sequenciamento de Contêiner. Com relação ao *buffer*, é importante destacar que uma solução de uma instância do problema precisa garantir que todos os contêineres inseridos no *buffer* sejam retirados, ou seja, o *buffer* inicia vazio e após todas as operações ele permanece vazio. Já o *Yard* inicia com os contêineres de exportação e após todas as operações ficarão apenas os contêineres de importação retirados do *vessel*. Vale ressaltar que a configuração e sequência de recuperação dos contêineres fazem parte de outro problema de otimização em terminais portuários, como pode ser visto no trabalho de (FIRMINO, 2019). Dessa forma, para não sair do escopo do trabalho, o Problema de Sequenciamento de Contêiner considera a sequência fornecida pelo Problema de Recuperação de Contêineres, pois quando as soluções forem integradas cada uma irá se responsabilizar com sua parte do problema.

Conforme foi visto, é possível identificar 5 operações:

- **VY:** Contêiner de importação do *Vessel* para o *Yard*;
- **YV:** Contêiner de exportação do *Yard* para o *Vessel*;
- **VB:** Contêiner de *reshuffle* do *Vessel* para o *Buffer*;
- **BV:** Contêiner de *reshuffle* do *Buffer* para o *Vessel*;
- **VV:** Contêiner de *reshuffle* do *Vessel* para o *Vessel*.

Cada operação realiza uma movimentação com um contêiner retirando-o de uma determinada posição e inserindo-o em outra, no entanto, dependendo da região que o contêiner for remanejado o tempo para realização dessa operação pode variar, sendo assim, considerando

que cada operação t gasta um tempo igual d^t segundos, na Tabela 1 é possível identificar um exemplo de uma instância conforme apresentado em Meisel e Wichmann (2010).

Tabela 1 – Tempo necessário para realização de uma operação t em segundos.

| t | d^t |
|-----|-------|
| VY | 100 |
| YV | 100 |
| VB | 100 |
| BV | 100 |
| VV | 90 |

Fonte: Meisel e Wichmann (2010)

Ao concluir uma operação o guindaste precisa se posicionar para iniciar a próxima operação, no entanto, esse tempo de transição do guindaste sem contêiner também pode variar. Na Tabela 2, é possível observar o exemplo apresentado por Meisel e Wichmann (2010), onde é possível observar o tempo d^{tu} em segundos, onde t é a operação que está sendo concluída e u a próxima operação que será iniciada, dessa forma d^{tu} representa o tempo necessário para o guindaste se movimentar sem contêiner para iniciar a próxima operação da solução.

Tabela 2 – Tempo necessário para movimentação do guindaste sem contêiner da operação t até a operação u em segundos.

| $t u$ | VY | YV | VB | BV | VV |
|-------|----|----|----|----|----|
| VY | 20 | 10 | 20 | 10 | 20 |
| YV | 10 | 20 | 10 | 20 | 10 |
| VB | 20 | 10 | 20 | 10 | 20 |
| BV | 10 | 20 | 10 | 20 | 10 |
| VV | 10 | 20 | 10 | 20 | 10 |

Fonte: Meisel e Wichmann (2010)

Após a descrição das informações principais sobre o CSP, será apresentado um exemplo de solução para uma instância, conforme mostra a Figura 8. No exemplo, é possível observar que essas configurações possuem 4 pilhas e 2 níveis; dessa forma, $m = 4$ e $n = 2$. Uma solução possível para esse problema pode ser observada na Tabela 3. Essa tabela mostra os detalhes das 8 operações que compõem uma solução para a instância apresentada. Assim, é possível obter o tempo gasto pelo guindaste para executar essa solução, que nesse caso será o somatório de todos os d^{tu} e os d^t , resultando no valor de 930 segundos. Já na Figura 9, é possível obter uma visão geral das modificações realizadas por cada operação.

Figura 8 – Exemplo de uma Configuração Inicial e de Partida de uma instância.

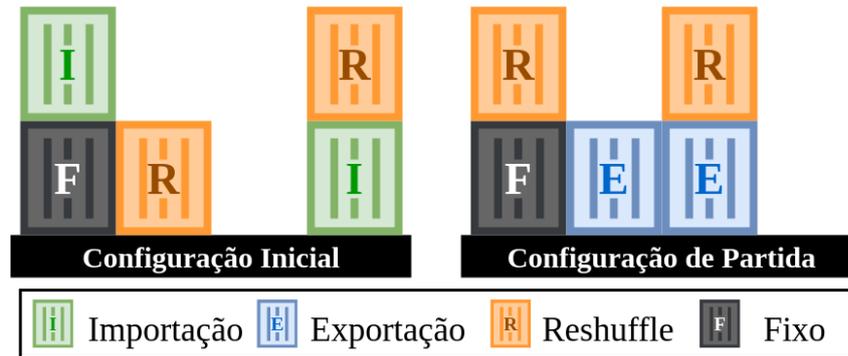


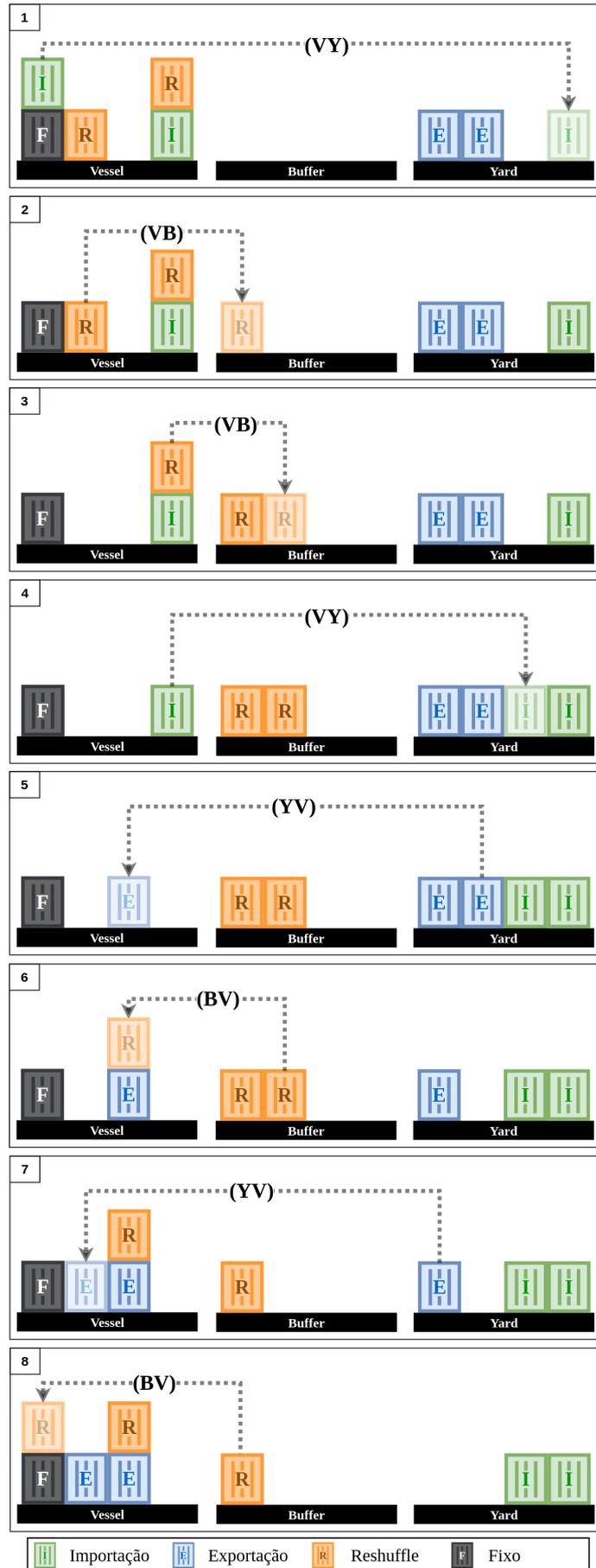
Tabela 3 – Tempo gasto em cada operação da solução.

| N | d^{tu} | Origem | Destino | Operação | d^t |
|--------------|----------|--------|---------|----------|------------|
| 1 | - | (1,2) | Y | VY | 100 |
| 2 | 20 | (2,1) | B | VB | 100 |
| 3 | 20 | (4,2) | B | VB | 100 |
| 4 | 20 | (4,1) | Y | VY | 100 |
| 5 | 10 | Y | (3,1) | YV | 100 |
| 6 | 20 | B | (3,2) | BV | 100 |
| 7 | 20 | Y | (2,1) | YV | 100 |
| 8 | 20 | B | (1,2) | BV | 100 |
| Total | | | | | 930 |

Fonte: Elaborada pelo autor (2022)

Após a descrição do exemplo de uma solução, é importante destacar que essa é uma das soluções viáveis para a instância apresentada desse problema. Na instância da Figura 8, nota-se que existe mais de uma possibilidade para realizar a primeira operação, que pode ser: VY com o contêiner de importação do $slot(1,2)$; VB com o contêiner de *reshuffle* do $slot(2,1)$; YV com o contêiner de exportação para ser inserido no $slot(3,1)$; e VB com o contêiner de *reshuffle* no $slot(4,2)$. É válido ressaltar que o guindaste possui diversas possibilidades para realizar a próxima operação. Dessa forma, surge o desafio de investigar alternativas que consigam encontrar soluções eficientes, que realizem as operações no menor tempo possível.

Figura 9 – Exemplo de uma solução com 8 operações.



Fonte: Elaborada pelo autor (2022)

2.3 FORMULAÇÃO MATEMÁTICA

De posse da descrição inicial do Problema de Sequenciamento de Contêiner, nesta seção será apresentada a formulação matemática. Ao analisar os trabalhos que abordaram o CSP, é possível destacar o trabalho de (MEISEL; WICHMANN, 2010), visto que, esse foi o primeiro a introduzir o contêiner de *reshuffle* ao problema, os trabalhos anteriores adotaram os contêineres de importação e exportação. Sendo assim, como a pesquisa atual faz uso do contêiner de *reshuffle*, então será utilizada a formulação matemática de Meisel e Wichmann (2010), descrita a seguir:

Notação dos dados de entrada com as configurações para uma instância do problema:

m número de pilhas da baia;

n número de níveis da baia;

A^I conjunto de *slots* com contêineres de importação na configuração inicial;

A^R conjunto de *slots* com contêineres de *reshuffle* na configuração inicial;

A conjunto de *slots* com contêineres a serem removidos, ou seja, *slots* que contêm um contêiner de importação ou de *reshuffle* na configuração inicial, dessa forma, tem-se que: $A = A^I \cup A^R$;

D^E conjunto de *slots* com contêineres de exportação na configuração de partida;

D^R conjunto de *slots* com contêineres de *reshuffle* na configuração de partida;

D conjunto de *slots* com contêineres a serem inseridos, ou seja, os *slots* que contêm um contêiner de importação ou de *reshuffle* na configuração inicial, dessa forma, tem-se que: $D = D^E \cup D^R$.

Descrição dos dados de entrada referente as operações:

T conjunto de tipos de operações com os contêineres, $T = \{VY, YV, VB, BV, VV\}$;

d^t tempo gasto para mover um contêiner durante uma operação $t \in T$;

d^{tu} tempo necessário para o guindaste, sem contêiner, sair da região na qual a operação t foi finalizada e se reposicionar na região que irá iniciar a operação u , sendo $t, u \in T$;

l limite superior do número de movimentações necessárias para processar uma baia;

K, \bar{K} conjunto com os índices das movimentações, $K = \{1, 2, \dots, l\}$, $\bar{K} = K \setminus \{l\}$.

É importante destacar que, a quantidade de operações que serão realizadas em uma solução pode variar, visto que essa versão do CSP possui contêineres de *reshuffle*. Dessa forma, esses contêineres podem ser manipulados pelo guindaste internamente no navio ou utilizando a região do *buffer*, sendo que, essa última realiza duas operações e a interna, uma operação. Sendo assim, o número de movimentações não é conhecido a priori, dessa forma o valor de l é um valor máximo possível para uma dada instância, que é definido, como:

$$l = |A^I| + |A^R| + |D^R| + |D^E| = |A| + |D|$$

Variáveis de decisão:

x_{ijk}^V binária, 1 se o contêiner for retirado da baia no *slot* (i, j) para a realização da operação $k \in K$;

x_k^Y binária, 1 se o contêiner for retirado do pátio para a realização da operação $k \in K$;

x_k^B binária, 1 se o contêiner for retirado do *buffer* para a realização da operação $k \in K$;

y_{ijk}^V binária, 1 se o contêiner for inserido na baia no *slot* (i, j) para a realização da operação $k \in K$;

y_k^Y binária, 1 se o contêiner for inserido no pátio para a realização da operação $k \in K$;

y_k^B binária, 1 se o contêiner for inserido no *buffer* para a realização da operação $k \in K$;

τ_k^t binária, e se o contêiner k realiza o tipo da operação $t \in T$;

e_k^{tu} binária, 1 se o contêiner k realiza o tipo da operação $t \in T$ e o contêiner $k + 1$ realiza o tipo da operação $u \in K$, ou seja, $\tau_k^t = \tau_{k+1}^u = 1$, indicando que o guindaste precisará ser reposicionado, sem contêiner, entre a região do final da operação k e a região do início da operação $k + 1$;

b_k valor inteiro com o número de contêineres *reshuffle* presentes no *buffer* após a operação k .

Modelo de otimização:

$$\text{minimizar} = \sum_{k \in K} \sum_{t \in T} d^t \cdot \tau_k^t + \sum_{k \in \bar{K}} \sum_{t, u \in T} d^{tu} \cdot e_k^{tu} \quad (2.1)$$

$$\sum_{k \in K} x_{ijk}^V = 1 \quad \forall (i, j) \in A \quad (2.2)$$

$$\sum_{k \in K} y_{ijk}^V = 1 \quad \forall (i, j) \in D \quad (2.3)$$

$$\sum_{k \in K} x_{i,j+1,k}^V \cdot k \leq \sum_{k \in K} x_{ijk}^V \cdot k \quad \forall (i, j), (i, j+1) \in A \quad (2.4)$$

$$\sum_{k \in K} y_{ijk}^V \cdot k \leq \sum_{k \in K} y_{i,j+1,k}^V \cdot k \quad \forall (i, j), (i, j+1) \in D \quad (2.5)$$

$$\sum_{k \in K} x_{ijk}^V \cdot k \leq \sum_{k \in K} y_{ijk}^V \cdot k \quad \forall (i, j) \in A \cap D \quad (2.6)$$

$$\sum_{t \in T} \tau_k^t \leq 1 \quad \forall k \in K \quad (2.7)$$

$$\tau_k^{VY} = \sum_{(i,j) \in A^I} x_{ijk}^V \quad \forall k \in K \quad (2.8)$$

$$\tau_k^{VY} = y_k^Y \quad \forall k \in K \quad (2.9)$$

$$\tau_k^{YV} = x_k^Y \quad \forall k \in K \quad (2.10)$$

$$\tau_k^{YV} = \sum_{(i,j) \in D^E} y_{ijk}^V \quad \forall k \in K \quad (2.11)$$

$$\tau_k^{VV} + \tau_k^{VB} = \sum_{(i,j) \in A^R} x_{ijk}^V \quad \forall k \in K \quad (2.12)$$

$$\tau_k^{VB} = y_k^B \quad \forall k \in K \quad (2.13)$$

$$\tau_k^{BV} = x_k^B \quad \forall k \in K \quad (2.14)$$

$$\tau_k^{VV} + \tau_k^{BV} = \sum_{(i,j) \in D^R} y_{ijk}^V \quad \forall k \in K \quad (2.15)$$

$$\sum_{t \in T} \tau_k^t \geq \sum_{t \in T} \tau_{k+1}^t \quad \forall k \in K \quad (2.16)$$

$$e_k^{tu} \geq \tau_k^t + \tau_{k+1}^u - 1 \quad \forall k \in \bar{K}, t, u \in T \quad (2.17)$$

$$b_0 = 0 \quad (2.18)$$

$$b_k = b_{k-1} - x_k^B + y_k^B \quad \forall k \in K \quad (2.19)$$

$$b_k \geq 0, \text{inteiro} \quad \forall k \in K \quad (2.20)$$

$$x_{ijk}^V, x_k^Y, x_k^B, y_{i'j'k}^V, y_k^Y, y_k^B, \tau_k^t, e_k^{tu} \in \{0; 1\} \quad \forall (i, j) \in A, (i', j') \in D, k \in K, t, u \in T \quad (2.21)$$

2.4 META-HEURÍSTICA

Conforme citado anteriormente, uma alternativa para encontrar soluções para o Problema de Sequenciamento de Contêiner é a adoção de meta-heurísticas. "Uma heurística é uma técnica computacional aproximativa que visa alcançar uma solução avaliada como aceitável para um dado problema que pode ser representado em um computador" (GOLDBARG; GOLDBARG; LUNA, 2017, p. 72). Originária do termo *heuriskein* que significa descobrir, uma heurística é uma forma de encontrar uma solução em um espaço de soluções, de forma que o tempo computacional para realizar esse procedimento seja o menor possível, sendo assim, não há garantias que a solução encontrada será a solução ótima. Desta forma, nesta seção será realizada uma descrição breve das meta-heurísticas GRASP, Algoritmo Genético e Algoritmo Genético de Chaves Aleatórias Viciadas.

2.4.1 GRASP

Proposto por Feo e Resende (1995), a meta-heurística GRASP (do inglês, *Greedy Randomized Adaptive Search Procedure*), de maneira geral, inicia construindo uma solução aleatória e gulosa, e depois realiza uma busca local, conforme apresentado no Código Fonte 1. A função `grasp` inicia fazendo a leitura da instância do problema (linha 2), em seguida, o laço `for` irá verificar se o critério de parada definido não foi satisfeito (linha 3), assim as três funções presentes em seu bloco de instruções serão executadas. Ao chamar a função `ConstructGreedyRandomizedSolution`, a mesma criará uma solução aleatória e gulosa (linha 4), de posse dessa solução viável, será iniciada a função `LocalSearch` (linha 5), por fim a função `UpdateSolution` atualiza a solução atual (linha 6). Após o critério de parada satisfeito, a função `grasp` retorna a melhor solução encontrada (linha 8).

Código Fonte 1 – Pseudocódigo do GRASP.

```
1 procedure grasp()
   InputInstance();
3   for GRASP stopping criterion not satisfied
       ConstructGreedyRandomizedSolution(Solution);
5       LocalSearch(Solution);
       UpdateSolution(Solution, BestSolutionFound);
7   of;
   return(BestSolutionFound);
9 end grasp;
```

Fonte: Feo e Resende (1995)

2.4.1.1 GRASP aplicado ao Problema de Sequenciamento de Contêiner

No trabalho de Meisel e Wichmann (2010), os autores adotaram o GRASP como técnica de otimização para encontrar soluções para o CSP, esse algoritmo possui uma boa aceitação na literatura, sendo utilizado para realizar comparações com novos algoritmos de otimização. Dessa forma, nessa subseção será apresentada uma descrição desse algoritmo. No Algoritmo 1, é possível notar o pseudocódigo do GRASP alinhado ao CSP.

Algoritmo 1: GRASP

Entrada: iterações, AC, DC

```

1 begin
2   melhorSolução = Empty
3   for it de 0 até iterações do
4     solução = Construção(AC, DC)
5     solução = BuscaLocal(solução)
6     if tempo(solução) < tempo(melhorSolução) then
7       melhorSolução = solução
8     end
9   end
10  return melhorSolução
11 end

```

Fonte: Elaborado pelo autor (2022)

Em linhas gerais, o algoritmo possui a fase de construção e a fase da busca local. Na fase de construção, o algoritmo identifica quais possibilidades podem ser realizadas para as próximas operações. No exemplo apresentado na Figura 8, nota-se a configuração inicial e de partida de uma instância, onde foi visto que existiam 4 possibilidades para a realização da próxima operação: VY com o contêiner de importação do *slot*(1, 2); VB com o contêiner de *reshuffle* do *slot*(2, 1); YV com o contêiner de exportação para ser inserido no *slot*(3, 1) e VB com o contêiner de *reshuffle* no *slot*(4, 2). No Algoritmo 2, é possível notar os passos realizados na função Construção(). Essa função recebe como entrada a configuração inicial e de partida, em seguida o estado atual da baía, representado pela variável *atual*, é iniciado com AC. Assim, a função *construir()* (linha 4) é responsável por identificar quais são as possibilidades de operações e inseri-las em uma lista de candidatas.

Algoritmo 2: CONSTRUÇÃO

Entrada: AC, DC

```

1 begin
2   atual = AC
3   while atual <> DC do
4     construir()
5     selecionar()
6     atualizar()
7   end
8   return solução
9 end

```

Fonte: Elaborado pelo autor (2022)

De posse da lista de candidatas, a função `selecionar()` (linha 5) irá escolher uma operação dentre as possibilidades presentes na lista de candidatas. Inicialmente, a função obtém o valor de w que se refere ao tempo gasto de uma operação somado ao tempo gasto com as operações presentes na solução atual. O intuito dessa função é escolher uma operação aleatoriamente da lista de candidatas, de forma que as operações que possuam um tempo menor tenham maior probabilidade de serem selecionadas. Dessa forma, na Equação 2.22 é possível obter o valor S que corresponde o inverso de w . Em seguida, a função aplica a Equação 2.23, obtendo o valor total de todos os elementos da lista. De posse do valor total, a função irá calcular a probabilidade de cada elemento da lista com base na Equação 2.24. Ou seja, será feita a divisão do inverso de w pelo valor total de todos elementos.

$$S_i = \frac{1}{w_i} \quad (2.22)$$

$$S^T = \sum_1^N \frac{1}{w_i} \quad (2.23)$$

$$P_i = \frac{S}{S^T} \quad (2.24)$$

Onde:

w tempo gasto da operação atual adicionado ao somatório do tempo gasto da sequência de operações;

S inverso do valor do somatório do tempo para o guindaste realizar uma sequência de operações, incluindo o tempo para realização de uma das operações da lista de candidatas;

S^T somatório do tempo total de cada operação após ser incluída na sequência de operações, ou seja, o valor de S para cada operação que se encontra na lista de candidatas;

N tamanho da lista de soluções candidatas;

P_i valor ponderado de cada operação da lista de candidatas.

No exemplo citado, para realizar a primeira operação, a função `construir()` identificou 4 possíveis operações, então a função `selecionar()` irá aplicar as 3 equações conforme mostra a Figura 10. Nesse exemplo, a lista de candidatas possui 4 operações, de forma que foram aplicadas as 3 equações obtendo o intervalo $[0, 1]$. Após a obtenção desse intervalo, é possível

selecionar uma das soluções candidatas. Então, ao gerar um número aleatório, será verificado a qual região do intervalo ele pertence e qual é a operação correspondente. Suponha que o número aleatório obtido foi 0.47: então ele pertence ao intervalo $[0.25, 0.50)$; dessa forma, o item 2 da lista de candidatas foi selecionado, ou seja, a Operação 2 que corresponde a VB com o contêiner de *reshuffle* do *slot*(2, 1). É importante destacar que essa função possibilita que todas as movimentações possíveis na configuração atual dos contêineres tenham chance de serem selecionadas, mas priorizando as operações que gastam menos tempo para serem realizadas.

Figura 10 – Exemplo de aplicação das equações 2.22, 2.23e 2.24 durante a execução da função *selecionar*().

| | | | | | |
|-----------|-------|-------|-------|-------|---|
| Operações | VY | VB | VY | VB | |
| <i>w</i> | 100 | 100 | 100 | 100 | |
| Equação 1 | 0.01 | 0.01 | 0.01 | 0.01 | |
| Equação 2 | 0.04 | | | | |
| Equação 3 | 0.25 | 0.25 | 0.25 | 0.25 | |
| Intervalo | 0 | 0.25 | 0.50 | 0.75 | 1 |
| | └───┘ | └───┘ | └───┘ | └───┘ | |
| | Op1 | Op2 | Op3 | Op4 | |

Fonte: Elaborada pelo autor (2022)

Em seguida, é executada a função *atualizar*() (na linha 6), que após a seleção da operação, será inserida na solução atual e retirada da lista de candidatas. Após a atualização, será verificado se o estado atual representado pela variável *atual* é idêntica à configuração de partida. Caso seja verdadeiro, o *while* finaliza e segue para a próxima função; caso seja diferente, as funções *construir*(), *selecionar*() e *atualizar*() serão executadas novamente até obter a configuração de partida, pois a condição de serem identificadas indica que a solução com as operações obtidas é uma solução viável para o problema.

Com a finalização do *while* na linha 3 do Algoritmo 2, o algoritmo possuirá uma solução válida para o problema, assim a função *Construção*() retorna a solução encontrada nessa fase para a função *GRASP* do Algoritmo 1, que por sua vez, faz a chamada da função *BuscaLocal*() (linha 5), que nesse algoritmo realiza dois passos, conforme pode ser visto no Algoritmo 3. O primeiro passo é realizado pela função *mover*() (linha 2), que realiza alterações na solução atual de forma que uma operação é deslocada para posições anteriores ou posteriores, dentro

de um intervalo seguro de movimentações. Esse passo é realizado com cada operação da solução atual; assim que inicia uma determinada operação, o algoritmo identifica um intervalo que essa operação pode percorrer. Para determinar o intervalo que a operação irá percorrer, deve-se observar a pilha dos *slots* da operação atual.

Algoritmo 3: BUSCALOCAL

Entrada: solução

```

1 begin
2   | solução = mover(solução)
3   | solução = transformar(solução)
4   | return solução
5 end

```

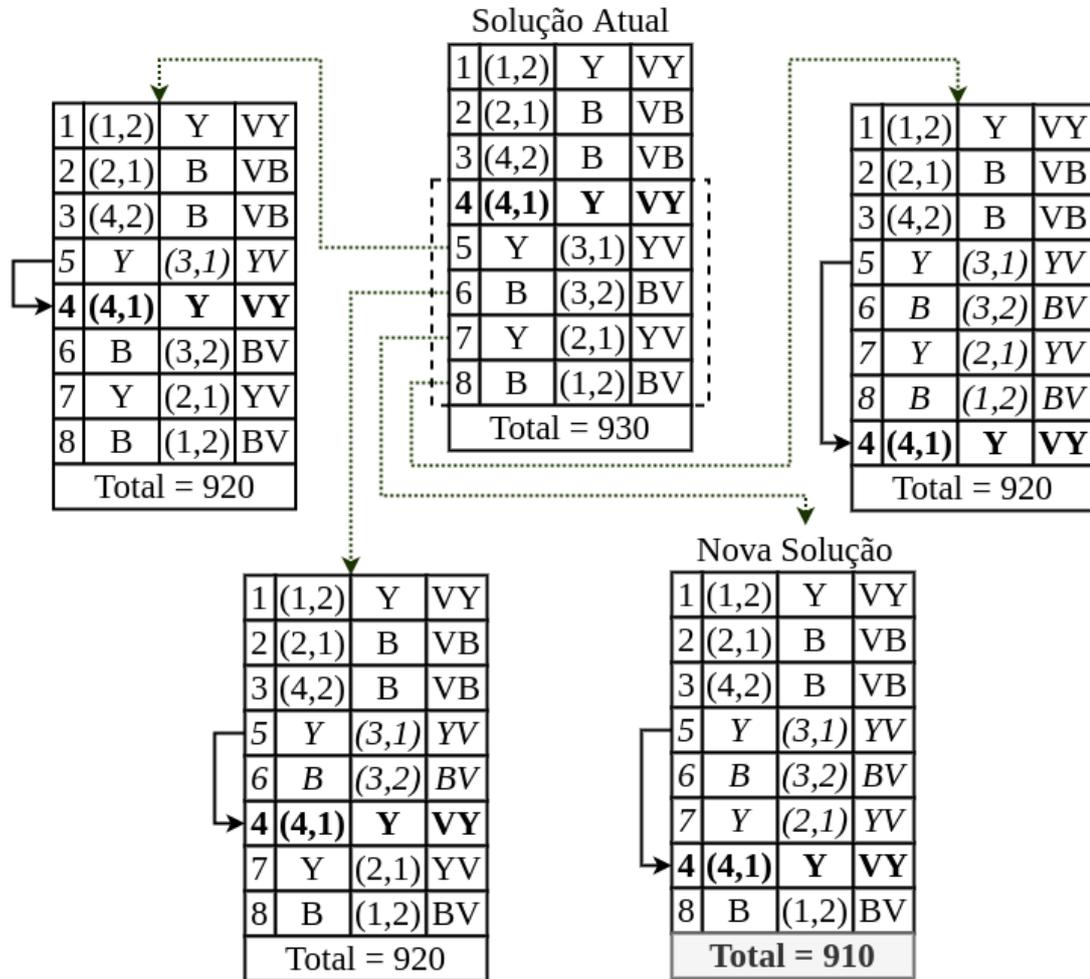
Fonte: Elaborado pelo autor (2022)

Na Figura 11, é possível notar um exemplo do funcionamento desse passo durante a execução da operação 4. Ao analisar a operação 4, é possível identificar que o intervalo será da operação 5 até a operação 8. Esse intervalo foi definido, pois, nesse exemplo, a operação 3 fez uma ação na pilha 4 que é a mesma pilha da operação 4; assim, as operações anteriores à operação 4 não foram consideradas, pois geram uma solução infactível; no entanto, as operações após a operação 4 foram todas consideradas, pois não afetam a pilha 4.

Visto que o intervalo varia entre 5 e 8, então a função `mover()` irá gerar 4 soluções nesta iteração, sendo que em cada solução a posição da operação 4 será deslocada em uma unidade, conforme mostra a Figura 11. A solução atual do exemplo possui um tempo total de 930 segundos. Ao realizar os deslocamentos foram encontradas 3 soluções com 920 segundos e 1 solução com 910 segundos. Dessa forma, a função armazenará essa nova solução onde a operação 4 foi deslocada para a posição da operação 7, pois essa possui o menor tempo entre as soluções geradas nessa iteração.

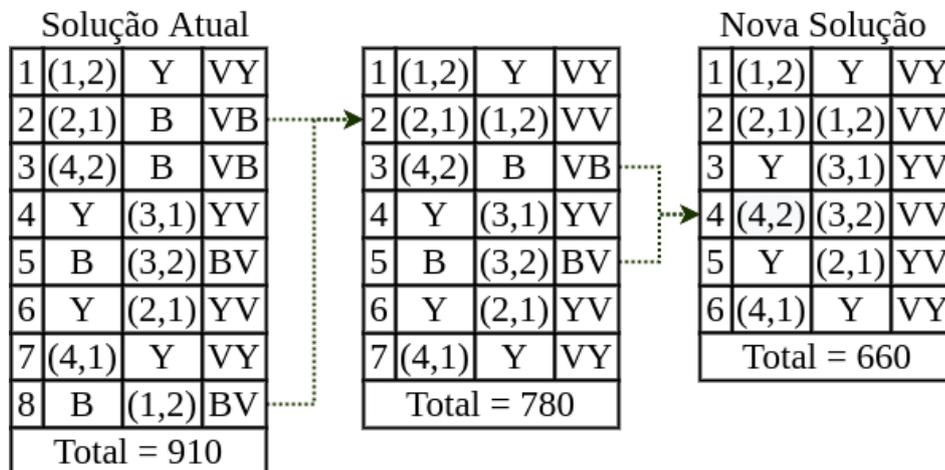
Em seguida a função `transformar()` (linha 3) inicia o segundo passo que consiste em transformar duas operações, sendo uma VB e outra BV, em uma nova operação VV que será a combinação das duas primeiras. Essa transformação gera uma solução com o tempo total menor e com um número menor de operações. Na Figura 12, é possível notar um exemplo da aplicação de duas transformações. Inicialmente, o algoritmo identificou que poderia transformar as operações 2 e 8 e inseriu a nova operação na posição 2, pois o `slot(1, 2)` não sofre mais alterações após a operação 1, já o `slot(2, 1)` sofre ação da operação 6; portanto, a nova operação VV não poderia ficar na posição da operação 8, pois geraria uma solução infactível.

Figura 11 – Exemplo da geração de novas soluções ao deslocar a operação 4 durante a execução da função mover().



Fonte: Elaborada pelo autor (2022)

Figura 12 – Exemplo da geração de novas soluções por meio das transformações durante a função transformar().



Fonte: Elaborada pelo autor (2022)

Em seguida, foi aplicada a transformação com as operações 3 e 5 da solução intermediária, gerando a operação 4 da nova solução. Nesse caso, a nova operação deve ficar na posição que estava a operação 5 da solução intermediária, visto que a operação 4 possui o $slot(3, 1)$ e a nova operação VV possui o $slot(3, 2)$; então, a nova operação deve ficar após a operação 4 da solução intermediária. Atualizando os identificadores das operações na nova solução, nota-se que a mesma possui 6 operações, onde a operação 2 foi gerada das operações 2 e 8 e a operação 4 foi gerada das operações 3 e 5. Com relação ao tempo total, a solução atual que possui 930 segundos foi atualizada para a nova solução que possui 660 segundos.

Ao executar esses dois passos, a função `BuscaLocal()` retorna a solução encontrada que é armazenada na variável `solução` do Algoritmo 1. Em seguida, será executado o `if` que verifica se a solução armazenada possui um tempo total menor que a solução atual armazenada em `melhorSolução`. Caso seja menor, o conteúdo de `melhorSolução` é atualizado, assim o algoritmo volta para a condição do `for` da linha 3 que irá repetir todas as funções até que o valor da variável `iterações` seja atingido.

2.4.2 Algoritmo Genético

O Algoritmo Genético – AG (do inglês, *Genetic Algorithm*), foi criado por John Holland. O AG tinham como objetivo inicial estudar formalmente o fenômeno de adaptação natural, tal como ocorre na natureza e para desenvolver maneiras pelas quais os mecanismos de adaptação natural podem ser importados para sistemas computacionais. No entanto, pesquisadores identificaram que o AG poderia ser adotado para solucionar problemas que possuíam um grau de combinação elevado, caracterizando assim o AG como uma técnica de otimização.

O AG pode ser definido como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural (LINDEN, 2012). O AG é uma técnica de otimização global, possuindo a característica de explorar o espaço de busca através dos operadores. Para solucionar um problema com AG, tem-se a necessidade de analisar os passos básicos do algoritmo a fim de identificar quais são as adaptações necessárias para que o mesmo atenda de maneira efetiva ao problema.

No que se refere a representação cromossomial, é importante destacar que esta é fundamental para o algoritmo genético, sendo a maneira básica de traduzir as informações de um determinado problema para que essas possuam um tratamento viável no computador (LINDEN, 2012). Em outras palavras, a representação cromossomial é a forma de conectar as proprie-

dades relevantes do mundo real ao algoritmo genético, com a finalidade de se obter o melhor resultado possível.

Existem vários tipos de representação cromossomial, como a binária, a de números reais, permutação de elementos para AG baseado em ordem, lista de regras, ou qualquer estrutura de dados necessária (LINDEN, 2012). De posse da representação cromossomial, pode-se observar, no Algoritmo 4, a sequência de etapas de um Algoritmo Genético Básico, conforme os itens descritos a seguir:

Algoritmo 4: ALGORITMOGENETICO

Entrada: tamanho, gerações, taxa

```
1 inicio
2   população = criarPopulação(tamanho)
3   for cont de 0 até gerações do
4     avaliar(população)
5     for i de 0 até tamanho do
6       pai1 = selecionar()
7       pai2 = selecionar()
8       filho = reproduzir( pai1, pai2)
9       novoFilho = mutar( filho, taxa)
10      novaPopulação[i] = novoFilho
11    end
12    moduloPopulação( população, novaPopulação)
13    melhorFilho = selecionarMelhor( população)
14  end
15  melhorFilho
16 fin
```

Fonte: Elaborado pelo autor (2023)

1. Inicialização da população de cromossomos;
2. Avaliação individual de cada cromossomo da população gerada;
3. Seleção dos pais da população;
4. Aplicação dos operadores de *crossover* e de mutação nos pais selecionados para obtenção de novos cromossomos da próxima geração;
5. Exclusão dos pais da população atual;
6. Avaliação dos filhos e inserção na nova população;

7. Verificação se já atende ao número de gerações, ou se o melhor indivíduo já satisfaz os requerimentos de desempenho. Caso positivo, será retornado o melhor indivíduo, caso contrário deve-se seguir para o passo 3.

2.4.2.1 Algoritmo Genético de Chaves Aleatórias Viciadas

O Algoritmo Genético de Chaves Aleatórias Viciadas (do inglês, *Biased Random Key Genetic Algorithm* - BRKGA) é uma metaheurística baseada nos Algoritmos Genéticos, que nessa versão possui um vetor de números reais na sua representação cromossomial. Na etapa da inicialização da população, o algoritmo necessita criar indivíduos alinhados a representação cromossomial, no caso do BRKGA, a representação é formada por um vetor de números reais aleatórios no intervalo $[0, 1)$, que por sua vez são chamadas de chaves aleatórias.

De posse do vetor de chaves aleatórias, é possível fazer o mapeamento para uma solução de um problema combinatório, como o problema de sequenciamento de contêiner. O algoritmo do BRKGA, conforme mostra o Algoritmo 5, inicia gerando os indivíduos compostos pelas chaves aleatórias na função `criarPopulação()`. Em seguida, os vetores aleatórios são decodificados conforme o problema desejado, assim é possível avaliar cada indivíduo da população. Logo após, inicia-se o laço de repetição, que ficará executando enquanto não atingir o critério de parada.

Algoritmo 5: BRKGA

```

Entrada: tamanho, gerações
1 inicio
2   população = criarPopulação(tamanho)
3   decodificarVetoresAleatórios(população)
4   for condição de parada do
5     ordenarAvaliação(população)
6     [elite, nãoElite] = particionarElite(população)
7     novaPopulação.add(elite)
8     novaPopulação.add(mutar(população))
9     crossover(elite, nãoElite)
10    decodificarVetoresAleatórios(novaPopulação)
11  end
12 fin

```

Fonte: Elaborado pelo autor (2023)

Na etapa seguinte, realizada pela função `ordenarAvaliação`, os indivíduos das soluções serão ordenados conforme o valor da função de avaliação. De posse dessa ordenação, a população será particionada entre as soluções elites e não elites, ou seja, as soluções elites são os

indivíduos que obtiveram melhores avaliações, de forma que essas são copiadas para a próxima geração, isto é, para a nova população, conforme mostra a linha 7 do Algoritmo 5.

Na etapa seguinte, inicia-se o operador de mutação que irá perturbar a solução, em busca de uma nova solução para explorar outra região do espaço de busca. Na função crossover, serão selecionados dois pais, sendo eles, um do conjunto de elite e outro do conjunto não elite. Na linha 10, o processo de decodificação dos vetores com as chaves aleatórias é realizado com o intuito de avaliar as soluções, visto que ao voltar o laço de repetição todo processo será repetido até que a condição de parada seja satisfeita.

2.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi possível obter uma visão geral sobre os principais problemas encontrados nos terminais portuários. Diante desse cenário foi destacado o Problema de Sequenciamento de Contêiner, identificando as principais variáveis, regiões utilizadas, os tipos de contêineres e as operações relacionadas a cada tipo. Adicionalmente, foi apresentada uma solução viável para uma instância que foi dada como exemplo, exibindo as modificações geradas após a aplicação de cada operação nas regiões dos terminais portuários.

Em seguida foi apresentada uma breve descrição da meta-heurística GRASP, bem como a versão aplicada ao CSP, em seguida a descrição do Algoritmo Genético e do Algoritmo Genético de Chaves Aleatórias Viciadas para encontrar soluções com tempo menor de realização das operações, conforme será visto nos Capítulos 3 e 4.

3 REVISÃO DA LITERATURA

Neste capítulo serão descritas as etapas realizadas durante a Revisão da Literatura do trabalho.

3.1 CONTEXTUALIZAÇÃO

Diante do objetivo do trabalho, foi necessário a realização de uma revisão sistemática com o intuito de identificar os principais trabalhos relacionados com essa pesquisa. Esta fase da pesquisa foi organizada em três etapas: Planejamento, Condução e Resultados que serão descritas nas próximas seções.

3.2 PLANEJAMENTO

Segundo Kitchenham e Charters (2007), o Planejamento é a etapa de definição de cada passo do protocolo, que consiste em: definição das Questões de Pesquisa; definição da *string* de busca; definição das bases de dados e definição os critérios de inclusão e exclusão. Sendo assim, nas próximas seções, serão descritos os passos desse protocolo.

3.2.1 Definição das Questões de Pesquisa

Inicialmente, são definidas as Questões de Pesquisa (QPs) que serão respondidas durante a fase de condução. Nesse trabalho foram definidas 5 QPs, que abordam: o método de otimização; o algoritmo de otimização utilizado; se foi destacado alguma restrição adicional; os tipos de contêineres adotados e a movimentação dos guindastes. As QPs definidas foram:

1. Qual o método de otimização encontrado no trabalho?

- Exato
- Uma heurística
- Uma simulação
- Outro
- Não se aplica

2. Qual o algoritmo adotado?

- Programação Dinâmica
- Programação Linear
- GRASP
- Algoritmo Híbrido
- Outro
- Não se aplica

3. Quais são as restrições ou características destacadas pelo trabalho analisado?

- Quantidade de ciclos
- Quantidade operações
- Outro
- Não se aplica

4. Usa quais tipos de contêiner?

- Importação
- Exportação
- Reshuffle*
- Fixo
- Outro
- Não se aplica

5. Qual a movimentação do guindaste utilizada no trabalho?

- Ciclo simples
- Ciclo duplo
- Não se aplica

3.2.2 Definição da string de busca

Após a definição das QPs, definiu-se uma *string* de busca composta com as principais palavras-chave sobre o tema, a qual pode ser observada a seguir:

(Container Sequencing Problem) AND ((Quay Crane) OR (Double Cycling))

3.2.3 Definição das bases de dados

Em seguida, foram definidas as principais bases de dados, na área de Ciência da Computação, disponíveis no Capes Periódicos¹. Foi possível identificar 7 bases de dados, como apresentado na Tabela 4. Para cada base selecionada, a *string* de busca foi configurada a fim de que a mesma atendesse às especificações de cada base.

Tabela 4 – Bases de dados adotadas.

| Base de Dados | Endereço Web |
|---------------------------|---|
| Arxiv | < https://arxiv.org/ > |
| EBSCOhost | < https://www.ebsco.com/ > |
| Google Scholar | < https://scholar.google.com/ > |
| IEEE | < https://ieeexplore.ieee.org/Xplore/home.jsp > |
| Science Direct (Elsevier) | < https://www.sciencedirect.com/ > |
| Scopus | < https://www.scopus.com/ > |
| Springer | < https://link.springer.com/ > |

Fonte: Elaborada pelo autor (2022)

3.2.4 Definição dos critérios de inclusão e exclusão

Para concluir essa etapa do planejamento, foram definidos os critérios de inclusão e exclusão que todos os artigos deverão satisfazer. Os critérios definidos são:

Critérios de inclusão:

- Estudos completos publicados em revistas ou conferências sobre o Problema de Sequenciamento de Contêiner;

¹ Disponível em: <<http://www.periodicos.capes.gov.br/>>

- Estudos teóricos ou experimentais com o objetivo de apresentar conceitos para o entendimento da área;
- Acessível eletronicamente e ter sido publicado no período de 2009 a 2023.

Critérios de exclusão:

- Estudos que não estejam relacionados ao Problema de Sequenciamento de Contêiner;
- Estudos que não respondem a nenhuma das questões de pesquisa;
- Artigos duplicados, ou seja, aqueles encontrados em mais de um engenho de busca;
- Artigos convidados, tutoriais, relatórios técnicos que não passam pelo critério de avaliação das conferências ou revistas;
- Estudos não disponíveis para *download* eletronicamente.

3.3 CONDUÇÃO

Na fase de condução foi colocado em prática o planejamento definido na seção de Planejamento, dessa forma a *string* de busca foi inserida em cada base de dados, respeitando as especificações de cada base. De posse dos resultados da busca, foi realizado o *download* dos estudos disponíveis. Em seguida, os trabalhos passaram pelos critérios de inclusão e exclusão. Na Tabela 5 é possível identificar a quantidade de trabalhos em cada base durante a busca, a realização do *download* e a aplicação dos critérios de inclusão e exclusão.

Tabela 5 – Quantidade de artigos em cada base de dados.

| Base de Dados | Resultados | <i>Download</i> | Inclusão/Exclusão |
|---------------------------|------------|-----------------|-------------------|
| Arxiv | 2 | 2 | 0 |
| EBSCOhost | 3 | 3 | 1 |
| Google Scholar | 13 | 12 | 2 |
| IEEE | 8 | 8 | 1 |
| Science Direct (Elsevier) | 4 | 4 | 2 |
| Scopus | 1 | 1 | 0 |
| Springer | 3 | 3 | 2 |
| Total | 34 | 33 | 8 |

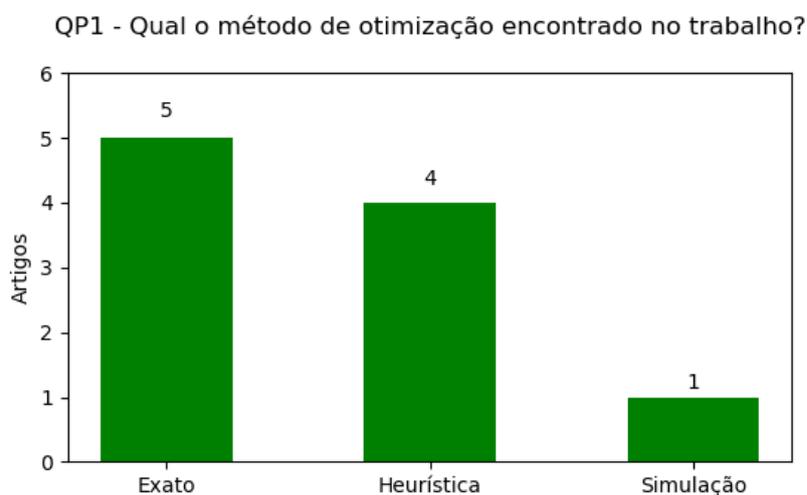
Fonte: Elaborada pelo autor (2022)

Após os critérios de inclusão e exclusão, foram identificados 8 artigos. De posse desses 8 artigos, iniciou-se a leitura de cada um com o intuito de responder às cinco questões de pesquisa.

3.4 RESULTADOS

Nessa última fase, serão apresentados os resultados obtidos das respostas das cinco questões de pesquisa. Na Figura 13, é possível observar um gráfico de barras correspondente a QP1, que se refere ao método de otimização adotado nos trabalhos analisados. Ao observar o gráfico é possível notar que 5 trabalhos usaram métodos exatos entre eles: o trabalho de Zhang e Kim (2009), que os autores utilizaram Programação Linear para realizar comparações com uma heurística que usa Busca Local; já o trabalho de Meisel e Wichmann (2010) os autores também utilizaram Programação Linear, no entanto, realizaram comparações com a meta-heurística GRASP; no trabalho de Lee, Liu e Chu (2015) os autores fizeram uso do algoritmo de Sidney, um algoritmo exato que busca minimizar o tempo de operações dos contêineres explorando o duplo ciclo dos guindastes; em Liu et al. (2015), os autores propuseram uma abordagem híbrida com uso de Programação Inteira com a heurística *Internal-Reshuffle-Dense*, esse algoritmo tem como objetivo maximizar o número de operações com o contêiner de *reshuffle*; e no trabalho de Zheng et al. (2020), os autores utilizaram Programação Dinâmica.

Figura 13 – Resultado da Questão de Pesquisa 1.



Fonte: Elaborada pelo autor (2022)

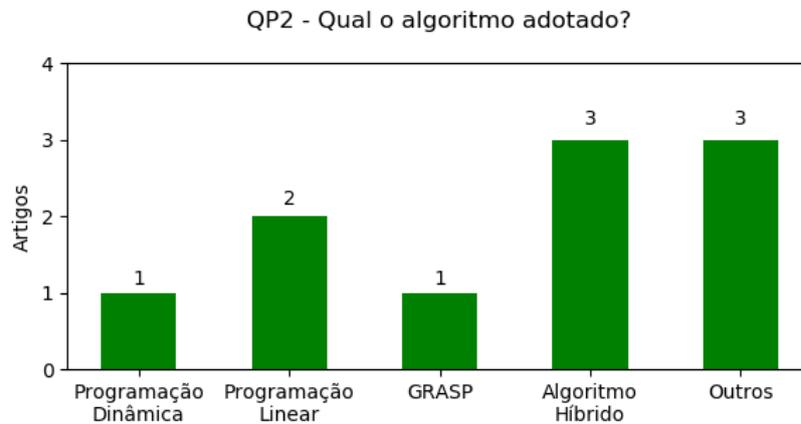
A outra alternativa da QP1, foi para identificar quais trabalhos adotaram uma heurística, conforme mostra o gráfico pode-se observar que 4 trabalhos adotaram alguma heurística,

sendo eles: o trabalho de Meisel e Wichmann (2010), como citado anteriormente, usou o algoritmo GRASP; No trabalho He, Wang e Zheng (2011) foi utilizada uma abordagem híbrida onde os autores definiram o algoritmo em duas partes, a primeira utiliza a *Johnson Rule's* e a segunda parte utiliza o *Particle Swarm Optimization*; no trabalho de Liu et al. (2015) foi utilizado um algoritmo híbrido que utilizou Programação Inteira e a heurística *Internal-Reshuffle-Dense*, conforme foi citado anteriormente; e no trabalho de Ding et al. (2017), os autores adotaram um algoritmo híbrido que possui duas etapas, inicialmente é aplicado o algoritmo *neighborhood search*, em seguida o algoritmo *tabu search*. Na alternativa simulação da QP1 foi possível observar que o trabalho de Ahmed et al. (2021) apresentou um modelo para simular as abordagens de ciclo simples e ciclo duplo dos guindastes.

A QP2 foi elaborada para identificar os algoritmos adotados pelos oito trabalhos. No gráfico da Figura 14, é possível observar os principais algoritmos encontrados. O trabalho que utilizou Programação Dinâmica foi Zheng et al. (2020). Já os trabalhos de Zhang e Kim (2009) e Meisel e Wichmann (2010) utilizaram Programação Linear para comparar os resultados com suas heurísticas. A heurística desenvolvida por Meisel e Wichmann (2010), foi o GRASP. Com relação aos Algoritmos Híbridos, foram identificados em três trabalhos: o primeiro trabalho foi de He, Wang e Zheng (2011), que utilizou o algoritmo *Johnson Rule's* com o algoritmo *Particle Swarm Optimization*; o segundo trabalho foi de Liu et al. (2015) que fez uso do algoritmo de Programação Inteira e o algoritmo chamado *Internal-Reshuffle-Dense* e o terceiro trabalho foi de Ding et al. (2017), que os autores apresentaram o algoritmo *Neighborhood Search* combinado com o algoritmo *Tabu Search*. Na alternativa outros foi possível observar três trabalhos: o trabalho de Zhang e Kim (2009) adotou o algoritmo de Busca Local para comparar com o algoritmo de Programação Linear; o trabalho de Lee, Liu e Chu (2015) fez uso do *Sidney's algorithm* e o trabalho de Ahmed et al. (2021) que realizou simulações, de forma que foi utilizado fluxogramas com as etapas das simulações.

A QP3 foi elaborada para identificar as restrições ou características dos trabalhos. Foi possível destacar duas características principais que os trabalhos consideraram. A primeira foi a quantidade de ciclos que os guindastes realizam na solução encontrada nos experimentos, dessa forma, os trabalhos de Zhang e Kim (2009) e Zheng et al. (2020) buscaram soluções visando a quantidade de ciclos. A segunda característica identificada foi a quantidade de operações realizadas pelas soluções, essa característica foi investigada pelos outros seis artigos restantes.

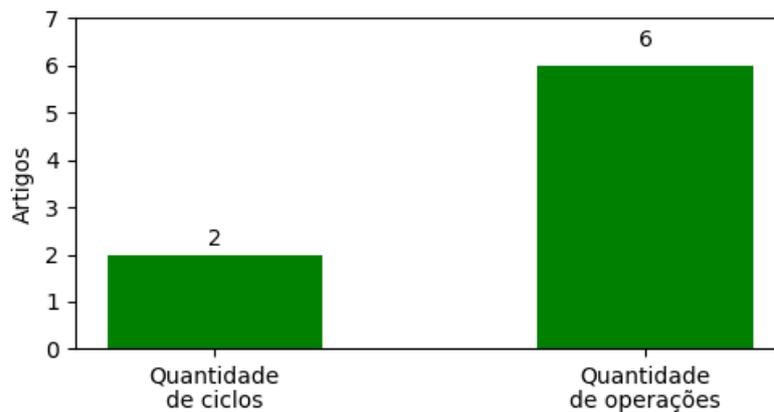
Figura 14 – Resultado da Questão de Pesquisa 2.



Fonte: Elaborada pelo autor (2022)

Figura 15 – Resultado da Questão de Pesquisa 3.

QP3 - Quais são as restrições ou características destacadas pelo trabalho analisado?



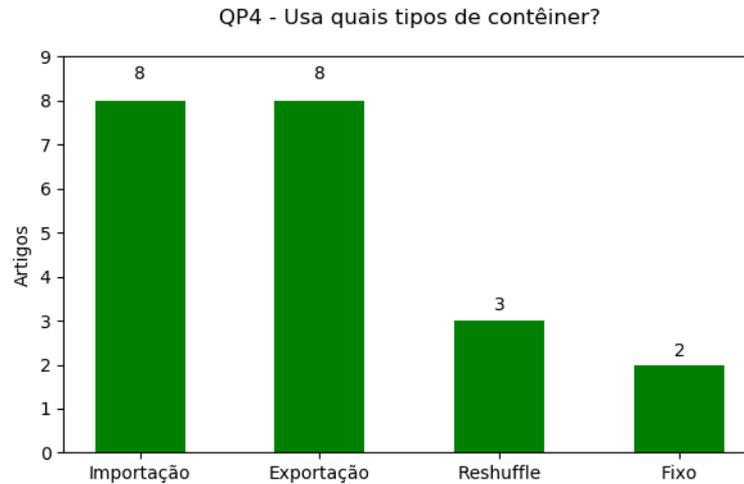
Fonte: Elaborada pelo autor (2022)

Na QP4, foi verificado o tipo de contêiner utilizado pelos trabalhos. Como já era esperado, os oito trabalhos utilizaram os contêineres de Importação e Exportação, conforme mostra a Figura 16. Já os trabalhos de Meisel e Wichmann (2010) e Liu et al. (2015) utilizaram também os contêineres de *reshuffle* e o fixo, enquanto no trabalho de Ding et al. (2017) os autores descreveram que consideraram o contêiner de *reshuffle*, mas não descreveram se fizeram uso de contêiner fixo.

A QP5 foi responsável por identificar quais as movimentações realizadas pelos guindastes que foram consideradas pelos trabalhos. Nessa questão, conforme mostra a Figura 17, foi possível observar que o trabalho He, Wang e Zheng (2011) utilizou uma abordagem com ciclo simples, enquanto o trabalho de Ahmed et al. (2021) que realizou simulações fez uso do ciclo simples e ciclo duplo e nos resultados fez a comparação de produtividade para cada tipo de

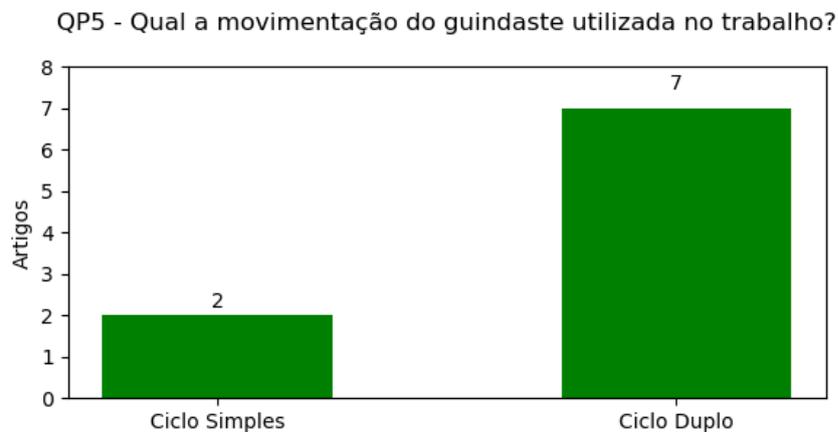
movimentação. Com relação ao ciclo duplo foi possível identificar esse tipo de movimentação nos demais trabalhos.

Figura 16 – Resultado da Questão de Pesquisa 4.



Fonte: Elaborada pelo autor (2022)

Figura 17 – Resultado da Questão de Pesquisa 5.



Fonte: Elaborada pelo autor (2022)

Dessa forma, foi possível descrever os resultados obtidos em cada Questão de Pesquisa. Adicionalmente, foram organizadas as informações sobre cada Questão de Pesquisa em cada um dos oito artigos identificados, conforme mostra a Tabela 6.

Tabela 6 – Identificação de cada alternativa das Questões de Pesquisa de cada artigo.

| Artigo | QP1 | | | QP2 | | | QP3 | | QP4 | | | QP5 | | | | |
|--------------------------|-------|------------|-----------|----------------------|--------------------|-------|-------------------|--------|-----------|--------|------------|------------|-----------|------|---------|-------|
| | Exato | Heurística | Simulação | Programação Dinâmica | Programação Linear | GRASP | Algoritmo Híbrido | Outros | Operações | Ciclos | Importação | Exportação | Reshuffle | Fixo | Simples | Duplo |
| Zhang e Kim (2009) | x | | | | x | | x | | x | | x | x | | | | x |
| Meisel e Wichmann (2010) | x | x | | | x | x | | | x | | x | x | x | x | | x |
| Lee, Liu e Chu (2015) | x | | | | | | | x | x | | x | x | | | | x |
| He, Wang e Zheng (2011) | | x | | | | | x | | x | | x | x | | | x | |
| Ahmed et al. (2021) | | | x | | | | | x | x | | x | x | | | x | x |
| Liu et al. (2015) | x | x | | | | | x | | x | | x | x | x | x | | x |
| Ding et al. (2017) | | x | | | | | x | | x | | x | x | | | | x |
| Zheng et al. (2020) | x | | | x | | | | | | x | x | x | x | | | x |

Fonte: Elaborada pelo autor (2022)

3.5 CONSIDERAÇÕES FINAIS

A realização dessa revisão foi relevante para identificar os principais trabalhos que tratam sobre o Problema de Sequenciamento de Contêiner com o guindaste em terminais portuários. Com as Questões de Pesquisa, foi possível observar como os estudos foram realizados, ou seja, quais os métodos de otimização, os algoritmos utilizados, as restrições ou as características principais, os tipos de contêineres e as movimentações realizadas pelos guindastes.

Ao mesmo tempo que se observa como os estudos foram realizados, também é possível notar possíveis lacunas, surgindo assim novas pesquisas. Com relação a esse trabalho, essa revisão ajudou a nortear tanto em relação à abordagem adotada pelos trabalhos, como também apresentando a ausência de algumas técnicas de otimização que não foram adotadas nos estudos, por exemplo, *Path Relinking*, Algoritmo Genético de Chaves Aleatórias Viciadas, entre outras.

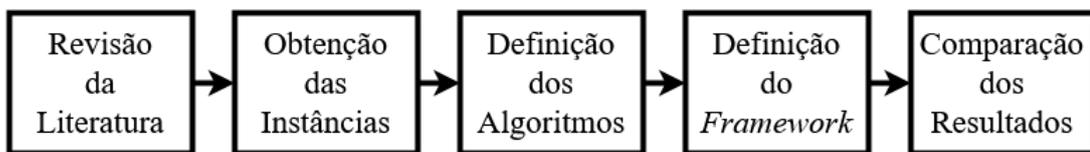
4 METODOLOGIA

Neste capítulo serão descritos os passos realizados durante o desenvolvimento da metodologia adotada neste trabalho, como: a revisão sistemática da literatura; a obtenção das instâncias do problema; os algoritmos desenvolvidos; a descrição do *framework*; e os resultados obtidos.

4.1 CONTEXTUALIZAÇÃO

Diante do objetivo do trabalho que consiste em identificar uma estratégia para encontrar soluções para o Problema de Sequenciamento de Contêiner com uso de guindaste em terminais portuários, assim, foi possível organizar a metodologia desse trabalho em 5 passos, conforme mostra a Figura 18. Nas subseções seguintes serão descritos os passos realizados durante esta pesquisa.

Figura 18 – Passos realizados durante o desenvolvimento desse trabalho.



Fonte: Elaborada pelo autor (2022)

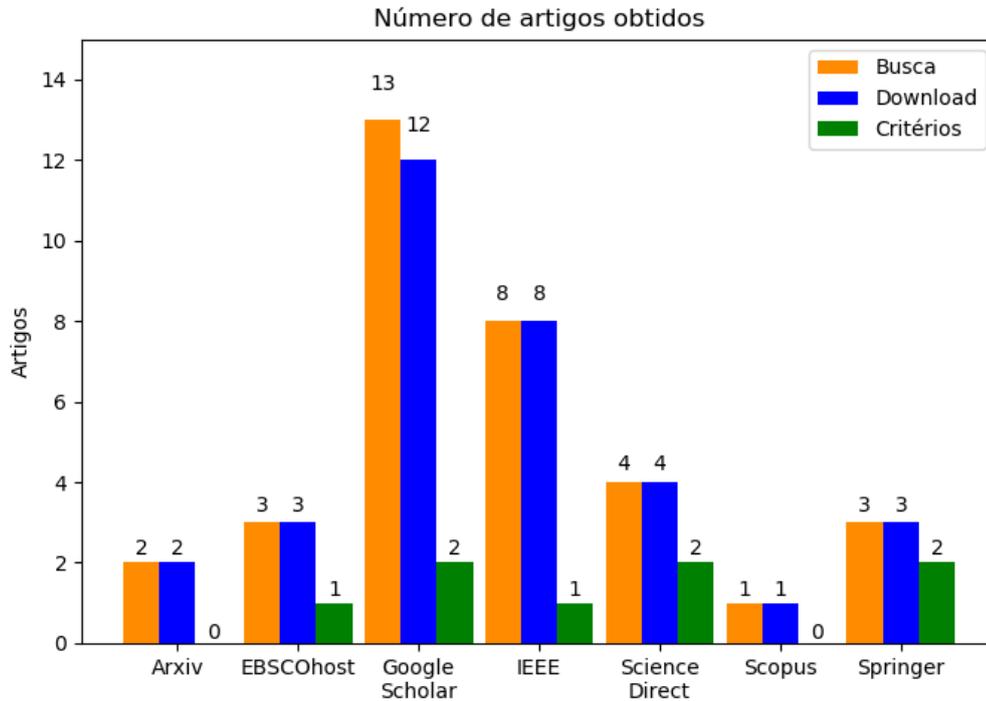
4.2 PASSO 1 - REVISÃO DA LITERATURA

O primeiro passo realizado no desenvolvimento da metodologia deste trabalho foi a Revisão da Literatura com o intuito de identificar os principais trabalhos que abordam o Problema de Sequenciamento de Contêiner. A descrição das três etapas realizadas durante a revisão pode ser observada no Capítulo 3.

Inicialmente, ao realizar a busca dos artigos nas bases de dados foram encontrados 34 trabalhos, logo após, foi realizado o *download* desses artigos, em seguida os mesmos passaram pelos critérios de inclusão e exclusão restando 8 trabalhos primários, conforme mostra a Figura 19. Entre os artigos resultantes foi possível notar os métodos de otimização utilizados, os principais algoritmos adotados, os tipos de contêineres, as características relevantes destacadas

pelos autores e o tipo de movimentação do guindaste. Desta forma, foi possível obter uma visão geral dos trabalhos que abordam o CSP.

Figura 19 – Quantidade de artigos obtidos após a pesquisa nas Bases de Dados, o *Download* e aplicação dos Critérios de Inclusão e Exclusão.



Fonte: Elaborada pelo autor (2022)

4.3 PASSO 2 - OBTENÇÃO DAS INSTÂNCIAS

Para encontrar um algoritmo para o CSP, se faz necessário obter os dados de entrada com os valores das informações principais, como: a quantidade de pilhas, quantidade de níveis, configuração inicial e a configuração de partida, os valores de d^t e d^{tu} . Visto que, o intuito desse trabalho é propor um algoritmo que consiga encontrar soluções que possuam um tempo menor para realizar as operações com os contêineres por meio do guindaste, então é importante fazer uso de um conjunto de instâncias para realizar comparações com os resultados encontrados na literatura. Diante deste contexto, nesse trabalho foi utilizado o conjunto de instâncias criado por Meisel e Wichmann (2010). No Anexo A, são apresentadas as informações da primeira instância do conjunto de dados disponibilizados pelos autores desse trabalho.

O conjunto de instâncias possui 990 arquivos, de forma que os mesmos estão organizados conforme: o valor de n , que representa a quantidade de pilhas e níveis; e o cenário. Nesse conjunto, a baía tem o mesmo valor para a quantidade de pilhas e de níveis, então o n

representa esses dois valores. Com relação ao cenário, as instâncias foram divididas em alto carregamento, baixa importação e baixa exportação. A quantidade de cada instância dessa organização do conjunto pode ser vista na Tabela 7.

Tabela 7 – Quantidade das instâncias do conjunto obtido na literatura.

| n | Cenário | Quantidade |
|--------------|-------------------|-------------------|
| 10 | alto carregamento | 110 |
| 10 | baixa importação | 110 |
| 10 | baixa exportação | 110 |
| 15 | alto carregamento | 110 |
| 15 | baixa importação | 110 |
| 15 | baixa exportação | 110 |
| 20 | alto carregamento | 110 |
| 20 | baixa importação | 110 |
| 20 | baixa exportação | 110 |
| Total | | 990 |

Fonte: Adaptado de Meisel e Wichmann (2010)

4.4 PASSO 3 - DEFINIÇÃO DOS ALGORITMOS

Ao investigar os trabalhos na literatura foi possível observar os principais algoritmos propostos pelos autores dos trabalhos primários, bem como perceber que os trabalhos realizaram comparações com o algoritmo GRASP aplicado ao CSP proposto por Meisel e Wichmann (2010). Então, nesse trabalho foi reimplementado esse algoritmo GRASP aplicado ao CSP seguindo as recomendações propostas pelos autores. A descrição das etapas do GRASP encontra-se na Subseção 2.4.1.1. Em seguida, foi desenvolvido o Algoritmo de Deslocamento Direcionado, que foi integrado ao GRASP. Adicionalmente foi desenvolvido o Algoritmo de Sub-Listas de Operações que também é um dos algoritmos propostos nesse trabalho, que em seguida foi integrado ao BRKGA com o intuito de encontrar a sequência de pilhas a ser explorada. Na Subseção 4.4.1 será descrito o funcionamento do algoritmo de Deslocamento Direcionado. Já a Subseção 4.4.2, possui a descrição do algoritmo de Sub-Listas de Operações. Por fim, a Subseção 4.4.3, possui a descrição do algoritmo de Sub-Listas de Operações e o Algoritmo Genético de Chaves Aleatórias Viciadas.

4.4.1 Algoritmo de Deslocamento Direcionado

Ao analisar o CSP e os dados de entrada obtidos pelo arquivo de configuração, foi possível notar que a transformação das operações VB e BV em uma nova operação VV gera uma solução que possui uma sequência de contêineres com um tempo total menor para a realização das operações com guindaste, de forma que foi desenvolvido um algoritmo chamando de Deslocamento Direcionado (DD), que busca encontrar as operações VV em uma solução. No Algoritmo 6, é possível observar o pseudocódigo do Algoritmo de Deslocamento Direcionado.

Algoritmo 6: Deslocamento Direcionado

```

Entrada:  $S$ 
1 inicio
2   solução = Empty, lista_VB = Empty, lista_BV = Empty
3   for  $i$  de 0 até  $length(S)$  do
4     if  $S[i].getType == "VB"$  then
5        $j_i, j_f = definirJanela(S, i)$ 
6       lista_VB.add( $(i, j_i, j_f)$ )
7       if  $j_f > i$  then
8         | solução = deslocar( $S, i, j_f$ )
9       end
10    end
11    if  $S[i].getType == "BV"$  then
12       $j_i, j_f = definirJanela(S, i)$ 
13      lista_BV.add( $(i, j_i, j_f)$ )
14      if  $j_i < i$  then
15        | solução = deslocar( $S, i, j_i$ )
16      end
17    end
18  end
19  for  $vb$  in lista_VB do
20    for  $bv$  in lista_BV do
21      /* Verificando se  $vb$  vem antes de  $bv$  */
22      if  $vb[0] < bv[0]$  then
23        /* Verificando se  $j_i$  de  $bv$  vem antes de  $j_f$  de  $vb$  */
24        if  $bv[1] < vb[2]$  then
25          | solução = transformar( $vb, bv$ )
26        end
27      end
28    end
29  end
30  return solução
31 fin

```

O algoritmo de Deslocamento Direcionado recebe uma solução como entrada, em seguida, essa solução será iterada, conforme mostra o for da linha 3. Dessa forma, serão identificadas as operações VB (linha 4) e BV (linha 11). Ao identificar uma operação desse tipo será chamada a função `definirJanela`, que retorna o intervalo que a operação atual poderá ser deslocada, assim cada operação será deslocada em uma direção, ou seja, a operação VB será deslocada em direção do final das operações (linha 7). Já a operação BV será deslocada em direção do início das operações (linha 14). Ao identificar esses dois tipos de operações, o algoritmo armazena a tupla (i, j_i, j_f) , onde i é a posição da operação na solução, j_i o início do intervalo de movimentação e j_f a posição final do intervalo da operação da posição i . De posse da tupla, essa será adicionada na lista de cada tipo, ou seja, a `lista_VB` possuirá as tuplas, do índice e os valores do intervalo da janela, das operações VB encontradas na solução, da mesma forma, a `lista_BV` possuirá as tuplas das operações BV.

Em seguida, será verificado se a operação atual pode ser deslocada, sendo assim, o `if` (linha 7) verifica se o valor do limite final (o valor de j_f), será maior que o valor de i , caso seja verdadeiro, isso significa que a operação pode ser deslocada em direção ao final da solução. Já o `if` da linha 11, verifica se o valor do limite inicial (o valor de j_i) será menor que o valor de i , caso seja verdadeiro, significa que a operação pode ser deslocada em direção ao início da solução. Caso as condições sejam verdadeiras, a função `deslocar` será chamada para colocar a operação na posição desejada.

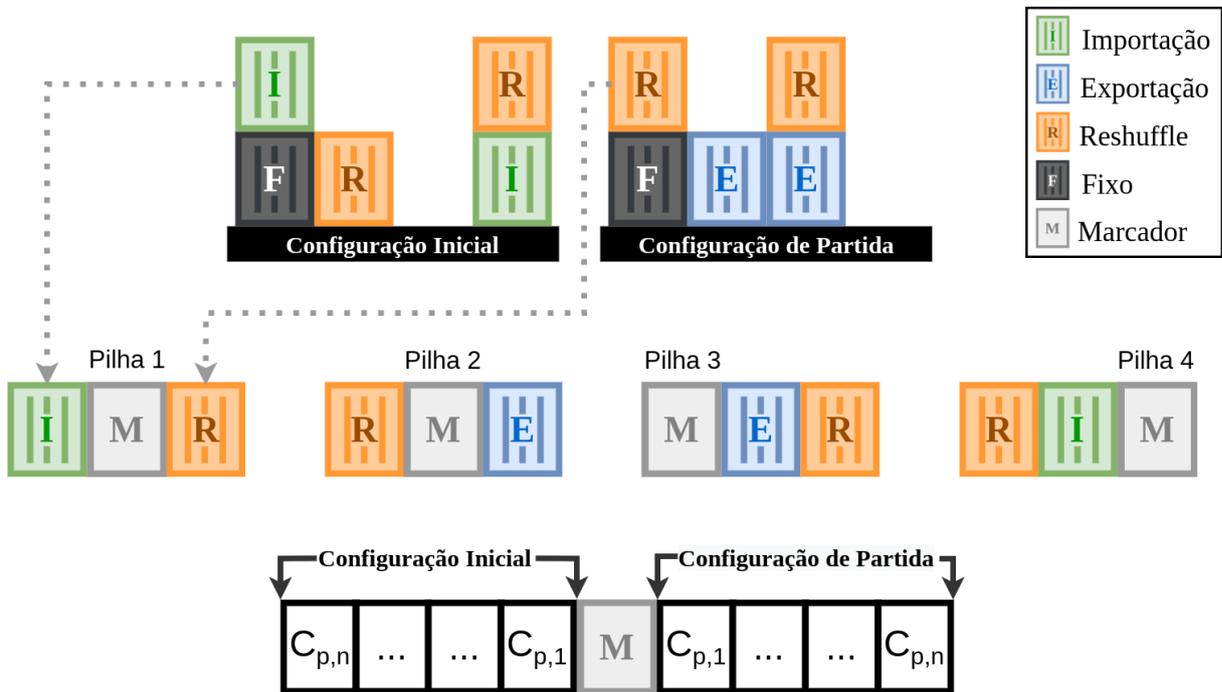
Com as listas das operações VB e BV preenchidas, o próximo passo consiste em identificar quais operações VB podem ser combinadas com as operações BV, assim o for da linha 19 percorre `lista_VB` e o for da linha 20 percorre `lista_BV`. Dessa forma, será verificado pelo `if` da linha 22, se o índice da operação VB é menor que o índice da operação BV, pois essa é uma condição necessária, visto que não poderá ocorrer recuperação de um contêiner de *reshuffle* do *buffer* antes da inserção do mesmo nessa região. Em seguida, o `if` da linha 24 verifica se o limite inicial de `bv` é menor que o limite final de `vb`. Caso essas condições sejam verdadeiras, então a função `transformar` será chamada para criar uma operação VV que será atualizada na solução.

4.4.2 Algoritmo de Sub-Listas de Operações

Ao analisar uma solução do CSP, ou seja, uma sequência de operações com os contêineres a ser realizada pelo guindaste, foi possível notar que uma pilha de uma baía possui uma sequência

de operações a ser realizada, de forma que essa sequência deverá aparecer na mesma ordem na solução encontrada pelos algoritmos. Na Figura 20, é possível observar um exemplo de construção da ordem de operações de cada pilha.

Figura 20 – Exemplo de construção de sequência de contêineres de cada pilha do Algoritmo de Sub-Listas de Operações.



Fonte: Elaborada pelo autor (2023)

Conforme mostra a Figura 20, é possível observar a sequência de contêineres de cada pilha. Para obter essa sequência, inicia-se pelo nível mais alto da Configuração Inicial de uma determinada pilha, de forma que os *slots* vazios e fixos são ignorados, pois não possuem operações associadas. Já os contêineres de importação e de *reshuffle* são adicionados conforme a ordem que os mesmos aparecem, esse processo continua até que o nível 1 seja atingido ou um contêiner fixo apareça. Ao consumir todos os contêineres de uma pilha da Configuração Inicial, será adicionado um Marcador, para demarcar a finalização dos contêineres pertencentes a Configuração Inicial. Em seguida, será analisada a mesma pilha da Configuração de Partida, no entanto, a ordem será do nível mais baixo para o mais alto, assim cada contêiner de exportação e de *reshuffle* serão adicionados, conforme a ordem que os mesmos aparecem, na sequência da pilha analisada.

Ao analisar a figura é possível notar a sequência genérica para todas as pilhas, ou seja, $C_{p,n}, \dots, C_{p,1}, M, C_{p,1}, \dots, C_{p,n}$. A sequência de cada pilha começa do nível mais alto da Configuração Inicial, depois insere o marcador e continua do nível mais baixo da Configuração

de Partida. No exemplo apresentado na figura é possível observar esse comportamento na construção da sequência de cada pilha. Em casos especiais, como foram apresentados na Pilha 3 e 4, nota-se que na Pilha 3 não existem contêineres na Configuração Inicial, assim a sequência começa com o Marcador e continua com os contêineres presentes da Configuração de Partida, essa por sua vez, apresenta um contêiner de exportação seguido de um contêiner de *reshuffle*, ou seja, a sequência será do nível mais baixo para o mais alto, quando a Configuração de Partida é analisada. Já a Pilha 4, possui contêineres na Configuração Inicial e nenhum contêiner na Configuração de Partida.

A ordem de construção da sequência dos contêineres de cada pilha segue a ordem que o guindaste irá realizar as operações na pilha analisada. É importante destacar que essa sequência deve aparecer nessa ordem na solução final, pois a mudança dessa ordem gera uma solução infactível para o CSP. Vale ressaltar que a combinação das operações entre as pilhas poderá gerar uma solução com menor tempo, no entanto, ao observar a sequência final da solução gerada por um algoritmo, a mesma deverá possuir a sequência de contêineres de cada pilha conforme foi apresentada.

Diante desse cenário foi possível desenvolver um algoritmo chamado de Sub-Listas de Operações que faz uso dessa característica de construção da sequência de contêineres de cada pilha. No Algoritmo 7, é possível observar o pseudocódigo do Algoritmo de Sub-Listas de Operações. O algoritmo recebe como entrada os dados presentes nos arquivos com as instâncias do CSP e a quantidade de interações. No `while` da linha 5, o algoritmo irá chamar as funções `CriarListas` e `EscolherConteineres` e a cada iteração irá avaliar a solução e atualizá-la conforme a mesma for melhorando.

No Algoritmo 8, é possível observar o pseudocódigo da função `CriarListas`, que recebe como entrada os dados da instância do problema. Essa função é responsável por construir cada sequência encontrada nas configurações de entrada, como foi descrito na Figura 20. Na linha 2, a lista `pilhas` é criada. O `for` da linha 3, inicia a construção da sequência de cada pilha, de forma que o `for` da linha 5 adiciona os contêineres de importação e de *reshuffle* na ordem que os mesmos aparecem na Configuração Inicial, após essa etapa o marcador é adicionado a sequência dessa pilha (linha 10), em seguida o `for` da linha 11, adiciona os contêineres de exportação e *reshuffle* conforme eles aparecem na Configuração de Partida. Com a sequência de contêineres de uma pilha construída, a variável `pilha` será adicionada na lista `pilhas`, assim a cada iteração do `for` da linha 3, uma sequência de contêineres de uma das pilhas será adicionada a lista `pilhas`, de forma que ao final de todas as iterações, o tamanho da lista

pilhas será igual a m , ou seja, a quantidade de pilhas da instância de entrada.

Algoritmo 7: *Sub-Listas de Operações*

Entrada: $m, n, AC, DC, d, dd, \text{iterações}$

```

1 inicio
2   melhorSolução = Empty
3   melhorValor = Infinty
4   cont = 0
5   while cont < iterações do
6     listas = CriarListas()
7     solução = EscolherConteineres(listas)
8     valor = avaliar(solução)
9     if valor < melhorValor then
10      melhorSolução = solução.copy()
11      melhorValor = valor
12      cont = 0
13    end
14    cont = cont + 1
15  end
16  return solução
17 fin

```

Fonte: Elaborado pelo autor (2023)

Algoritmo 8: *CriarListas*

Entrada: m, n, AC, DC

```

1 inicio
2   pilhas = Empty
3   for  $i$  de 0 até  $m$  do
4     pilha = Empty
5     for  $j$  de  $n$  até 1 do
6       if  $AC[i][j] == "I"$  or  $AC[i][j] == "R"$  then
7         pilha.add( $AC[i][j]$ )
8       end
9     end
10    pilha.add("M")
11    for  $j$  de 1 até  $n$  do
12      if  $DC[i][j] == "E"$  or  $DC[i][j] == "R"$  then
13        pilha.add( $DC[i][j]$ )
14      end
15    end
16    pilhas.add(pilha)
17  end
18  shuffle(pilhas)
19  return pilhas
20 fin

```

Fonte: Elaborado pelo autor (2023)

Após a conclusão do laço `for`, será chamada a função `shuffle` que irá escolher aleatoriamente a ordem que as pilhas que serão analisadas. Por fim, a função `CriarListas`, retornará a lista `pilhas`, que possui as sequências geradas pelo algoritmo conforme a descrição apresentada na Figura 20. Em seguida, o Algoritmo 7, chamará a função `EscolherConteineres`, que o seu pseudocódigo encontra-se no Algoritmo 9.

A função `EscolherConteineres`, recebe como entrada a lista com as sequências de cada pilha. Na linha 2 do Algoritmo 9, as listas `retirar`, `inserir` e `aguardar` são criadas. Em seguida, a lista `retirar` recebe uma cópia da lista `pilhas`. Na linha 4, o laço de repetição `while` será executado enquanto as três listas tiverem elementos. A proposta desse algoritmo é conseguir combinar os contêineres, dessa forma o bloco do `while` possui três `ifs` principais, um para cada lista.

O `if` da linha 5, irá escolher um contêiner de importação ou de *reshuffle*, presentes no primeiro elemento da lista `retirar`, caso o contêiner seja de *reshuffle* e a lista `inserir` estiver vazia, esse será adicionado na solução, pois o intuito é liberar uma pilha para que ocorra a combinação com os contêineres de exportação. Caso a lista `inserir` tenha elementos, então é possível adicionar a primeira pilha da lista `retirar` na lista `aguardar`, com o intuito de encontrar um contêiner de *reshuffle* na lista `inserir`, para assim gerar uma operação VV. Ao encontrar o Marcador, a primeira pilha da lista `retirar`, será removida e adicionada na lista `inserir`, pois o marcador simboliza que todos os contêineres da Configuração Inicial já foram processados, e já estão disponíveis para receber os contêineres do Pátio, do *Buffer* ou de outra pilha.

O `if` da linha 21, possui o tratamento da lista `inserir`, que ao encontrar os contêineres de exportação, os mesmos serão adicionados na solução. Caso o contêiner seja de *reshuffle*, será verificado se a lista `aguardar` possui alguma pilha, pois a combinação com o contêiner de *reshuffle* da lista `aguardar`, que é do tipo VB, com o contêiner de *reshuffle* da lista `inserir` do tipo BV, gera uma operação VV. Na linha 30, será verificado se o primeiro contêiner da lista `retirar` é de *reshuffle*, pois também possibilita a transformação em uma operação VV. Adicionalmente, será verificado se a primeira pilha da lista `inserir` encontra-se vazia, visto que, nesse caso ela pode ser removida.

O `if` da linha 35, é responsável pela lista `aguardar`, no entanto, esse trecho será executado se as listas `retirar` e `inserir` estiverem vazias, pois significa que o algoritmo inseriu a pilha na lista `aguardar` e conseguiu processar as demais, então a pilha volta para a lista `inserir` para ser concluída.

Algoritmo 9: *EscolherConteineres*

Entrada: pilhas

```

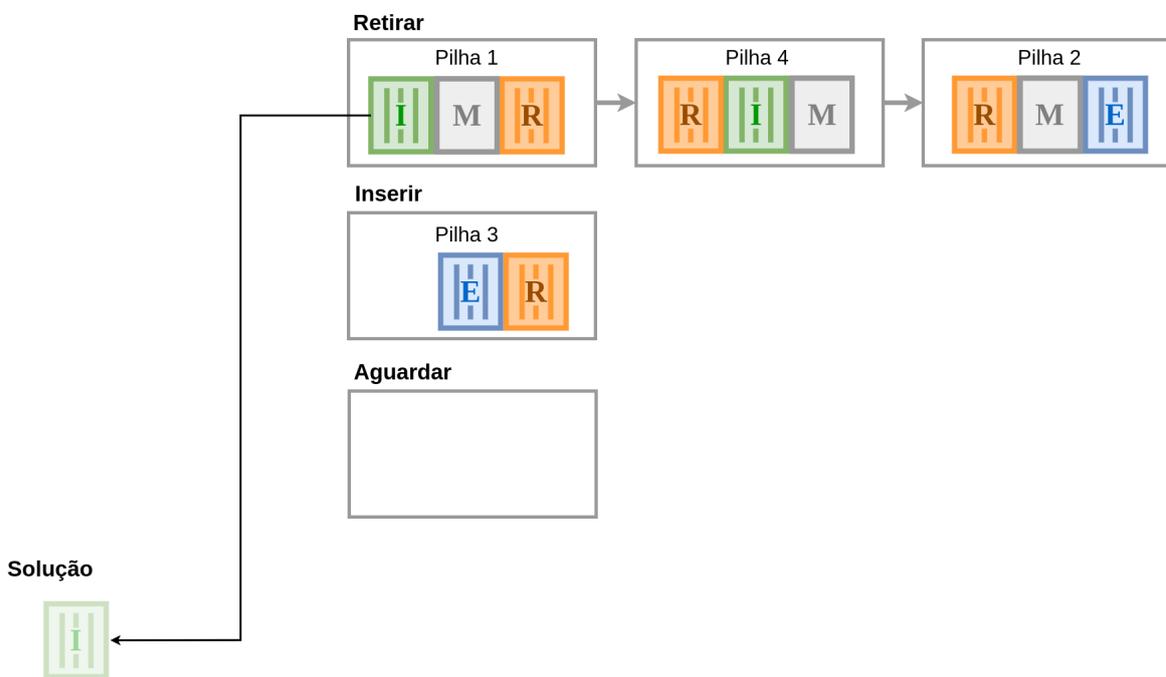
1 inicio
2   retirar = Empty, inserir = Empty, aguardar = Empty
3   retirar = pilhas.copy()
4   while len(retirar) <> 0 or len(inserir) <> 0 or len(aguardar) <> 0 do
5       if len(retirar) <> 0 then
6           if retirar[0][0] == "I" then
7               | solucao.add(retirar[0].pop([0]))
8           end
9           if retirar[0][0] == "R" then
10              if len(inserir) == 0 then
11                  | solucao.add(retirar[0].pop([0]))
12              end
13              else
14                  | aguardar.add(pop(retirar[0]))
15              end
16          end
17          if retirar[0][0] == "M" then
18              | inserir.add(pop(retirar[0]))
19          end
20      end
21      if len(inserir) <> 0 then
22          if inserir[0][0] == "E" then
23              | solucao.add(inserir[0].pop([0]))
24          end
25          if inserir[0][0] == "R" then
26              if len(aguardar) <> 0 then
27                  | solucao.add(tranformar(aguardar[0].pop([0]), inserir[0].pop([0]))
28              end
29              else
30                  if retirar[0][0] == "R" then
31                      | solucao.add(tranformar(retirar[0].pop([0]), inserir[0].pop([0]))
32                  end
33              end
34          end
35          if len(inserir[0][0]) == 0 then
36              | pop(inserir[0])
37          end
38      end
39      if len(aguardar) <> 0 then
40          if len(retirar) == 0 and len(inserir) == 0 then
41              | retirar.add(pop(aguardar[0]))
42          end
43      end
44  end
45  return solucao
46 fin

```

Em seguida a função EscolherConteineres retorna a solução encontrada, que será avaliada, caso possua um tempo total de realização das operações menor, essa solução será armazenada na variável melhorSolução. Considerando o exemplo apresentado na Figura 8, a seguir serão detalhadas as operações realizadas pelo Algoritmo Sub-Listas de Operações.

Inicialmente o algoritmo realiza a criação das listas de sequências de cada pilha conforme foi apresentado na Figura 20. Em seguida, as pilhas são adicionadas na lista Retirar, conforme mostra a Figura 21.

Figura 21 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VY da Pilha 1.

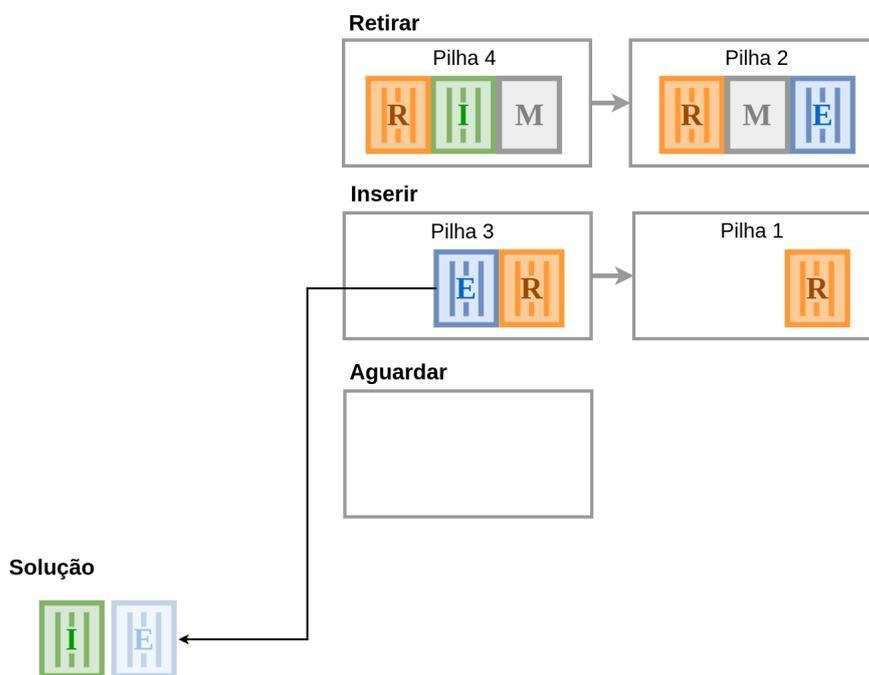


Fonte: Elaborada pelo autor (2023)

Conforme mostra a figura, estão presentes na lista Retirar as Pilhas 1, 4 e 2, é importante destacar que ao finalizar a construção das pilhas na função CriarListas, essa função aplica a função shuffle, de forma que, a ordem que as pilhas serão processadas de forma aleatória. Nesse exemplo, foi escolhida a ordem 3, 1, 4 e 2, como a Pilha 3 não possui elementos antes do Marcador, então a Pilha 3 foi adicionada na lista Inserir. Nessa etapa, inicia-se analisando se a lista possui algum contêiner para ser retirado do navio, assim o contêiner de importação da Pilha 1 é adicionado na solução. Em seguida é verificado que o próximo elemento da Pilha 1, é o Marcador, dessa forma a Pilha 1, é removida da lista Retirar e adicionada na lista Inserir.

Na Figura 22, é possível observar os próximos passos, a Pilha 1 já foi adicionada na lista Inserir. Após adicionar um contêiner na solução da lista Retirar, agora é o momento de verificar se é possível adicionar um elemento da lista Inserir, pois o intuito do algoritmo é conseguir combinar um contêiner da lista Retirar e um contêiner da lista Inserir, formando assim um duplo ciclo com o guindaste, gerando uma sequência com o tempo total menor. Assim, conforme a Figura 22, é possível verificar o que contêiner de exportação da Pilha 3 foi adicionado na solução.

Figura 22 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação YV da Pilha 3.



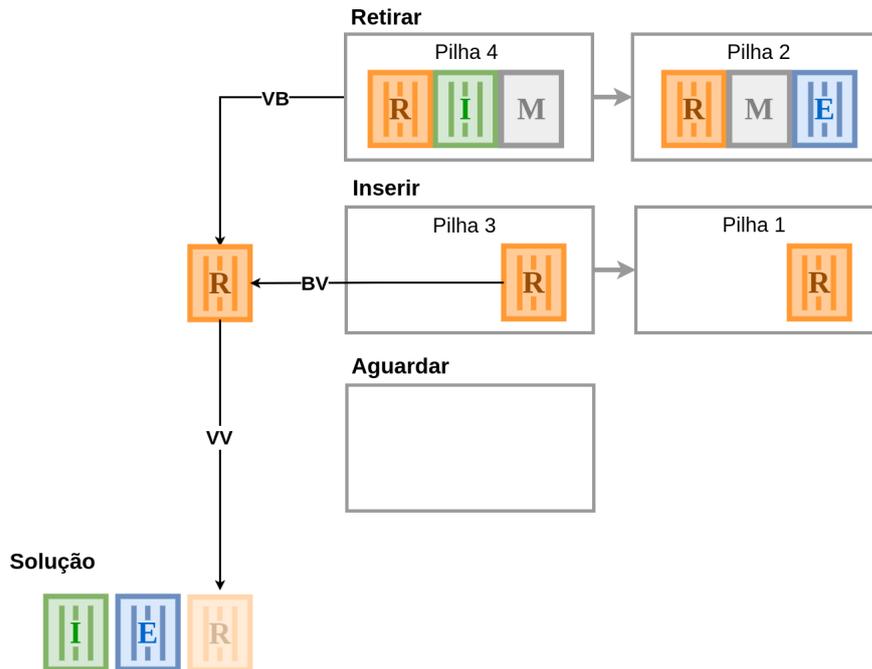
Fonte: Elaborada pelo autor (2023)

Na Figura 23, é possível observar que o primeiro elemento da lista Retirar é do tipo *reshuffle*, e que a lista Inserir também possui um contêiner *reshuffle*. Sendo assim, é possível realizar a transformação em uma operação VV, conforme mostra a figura.

Na Figura 24, é possível notar a atualização das listas. Após a operação VV, a Pilha 3 foi removida da lista Inserir, pois não possuía mais contêineres. Então, será analisada a lista Retirar que possui um contêiner de importação na Pilha, logo o mesmo será adicionado na solução.

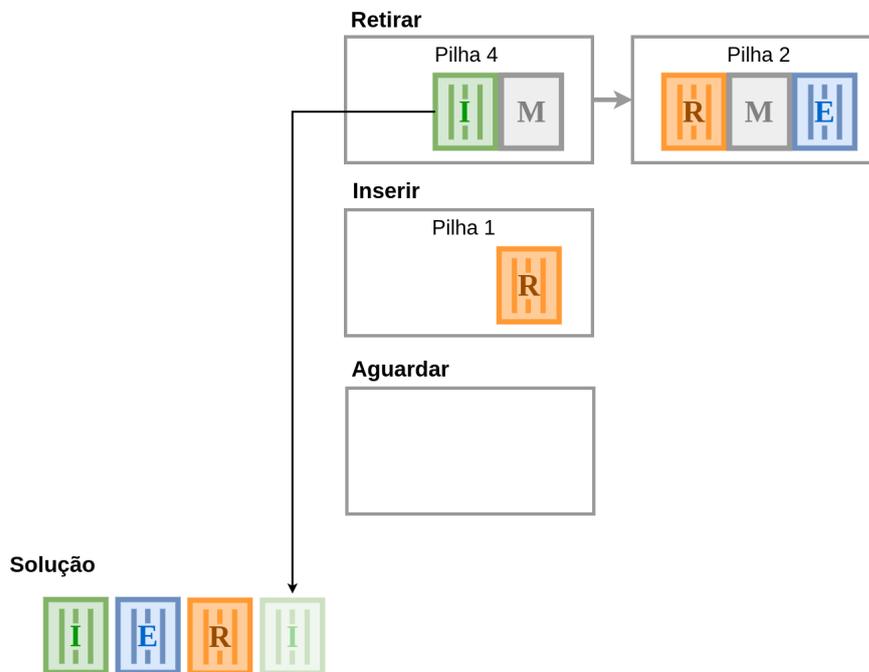
Na Figura 25, é apresentada uma operação VV, entre as Pilhas 2 e 1. Adicionalmente, a Pilha 4 foi removida da lista Retirar, no entanto, como a mesma não possui mais contêineres após o marcador, então foi concluída.

Figura 23 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VV com as Pilhas 4 e 3.



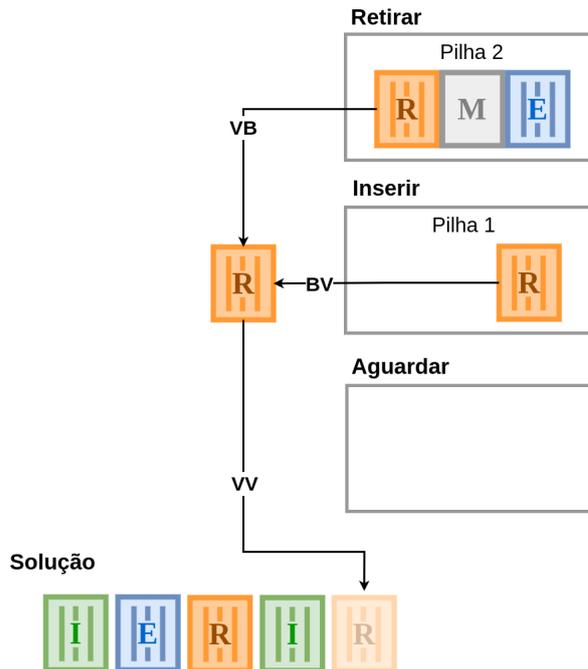
Fonte: Elaborada pelo autor (2023)

Figura 24 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VY da Pilha 4.



Fonte: Elaborada pelo autor (2023)

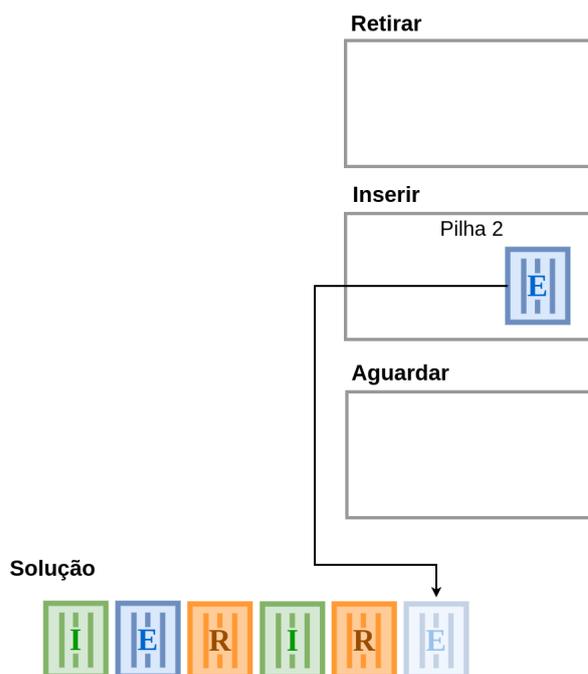
Figura 25 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VV com as Pilhas 2 e 1.



Fonte: Elaborada pelo autor (2023)

Por fim, na Figura 26, é apresentada a operação YV com a Pilha 2, presente na lista Inserir.

Figura 26 – Exemplo de execução do Algoritmo Sub-Listas de Operações. Operação VV com as Pilhas 2 e 1.



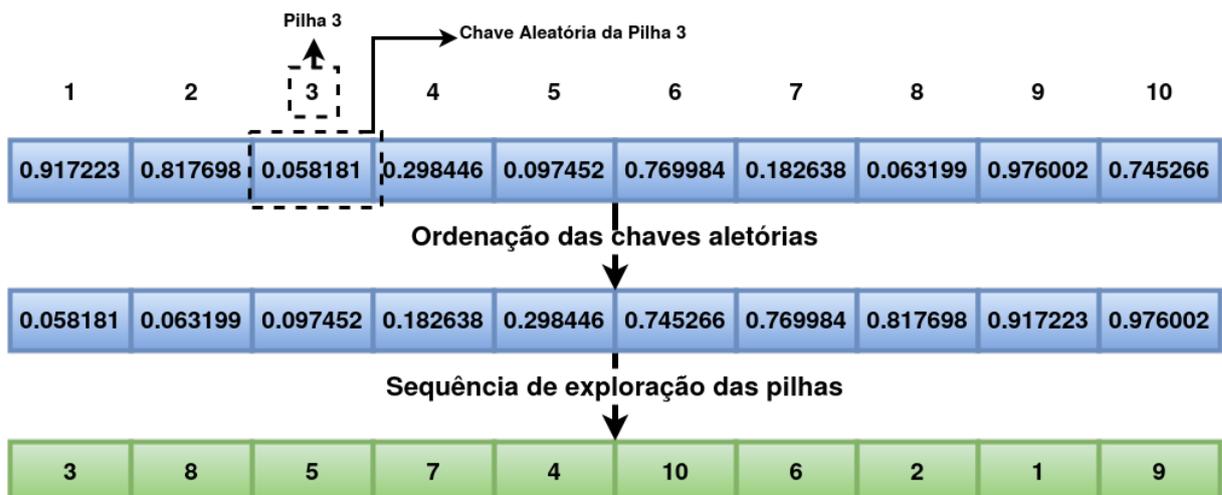
Fonte: Elaborada pelo autor (2023)

É válido ressaltar que o algoritmo busca combinar as operações, de forma que, ao manipular cada pilha nas três listas, isso permite que o algoritmo combine os contêineres que estão prontos para serem retirados do navio com os contêineres que estão prontos para serem inseridos no navio. Adicionalmente, essa abordagem prioriza o processamento de uma pilha que precisa ser descarregada de forma prioritária, pois permite que essa pilha esteja disponível para ser carregada nas próximas operações.

4.4.3 Algoritmo de Sub-Listas de Operações + BRKGA

Após o desenvolvimento do Algoritmo de Sub-Listas de Operações, foi possível observar que o mesmo poderia ser combinado ao BRKGA, com o intuito de encontrar uma sequência de exploração de pilhas onde fosse possível melhorar as soluções encontradas, dessa forma, o SLO foi combinado ao BRKGA. Nesse trabalho, foi adotado o BRKGA desenvolvido por Silva, Resende e Pardalos (2015), que segue a descrição da SubSeção 2.4.2.1, nessa implementação, o BRKGA possui uma etapa de decodificação das chaves aleatórias. Para integrar o BRKGA com o SLO foi preciso desenvolver o decode, que nesse problema foi proposto a identificação da sequência de exploração das pilhas da baía. Na Figura 27, é possível observar um exemplo dos passos realizados no decode desenvolvido.

Figura 27 – Exemplo do decode do BRKGA.



Fonte: Elaborada pelo autor (2023)

Inicialmente o decode recebe um vetor de números aleatórios correspondentes as chaves, então cada chave será associada uma pilha conforme a ordem gerada. Em seguida, as chaves são ordenadas, dessa forma a sequência de pilhas serão embaralhados, assim esse novo vetor

com as pilhas será passado para o SLO, pois este corresponde a sequência de exploração das pilhas. Sabendo que o algoritmo SLO retorna o tempo total de realização das operações com os contêineres realizadas pelo guindaste, então esse valor será adotado como o resultado da função *fitness* do BRKGA.

4.5 DEFINIÇÃO DO *FRAMEWORK*

Durante o desenvolvimento dos algoritmos foi identificada a necessidade de uma estratégia para auxiliar na construção das soluções, ou seja, conforme a realização de uma operação seria interessante verificar qual o efeito gerado após a realização dessa operação pelo guindaste. Diante dessa problemática e após a revisão da literatura, foi proposta a criação de um *framework* que fosse genérico e que possibilitasse a integração com outros problemas relacionados à otimização em terminais portuários, visto que, uma solução do CSP determina uma sequência de operações entre o navio e o pátio, e esta por sua vez deve ser alinhadas as soluções dos demais problemas encontrados no terminal portuário, como alocação de berço, problema de recuperação de contêiner, entre outros. O *framework* busca facilitar a construção dos experimentos, possibilitando que os especialistas observem qual o impacto de suas soluções ao aplicá-las no problema, bem como acompanhar cada operação durante a execução da solução. Para a realização dos experimentos, pode-se seguir os seguintes passos, detalhados na próxima Subseção.

4.5.1 Passos para realização de um experimento

Visto que o *framework* possibilitará a construção dos experimentos, é importante definir quais os passos necessários para a criação dos mesmos:

Passo 1 - Definição das configurações da instância: neste passo, será definido os dados de entrada para os algoritmos. Será necessário definir a quantidade de pilhas, o número de níveis, valores como: o tempo para realizar cada operação; os valores do tempo gasto da transição entre as operações; a configuração inicial e de partida.

Passo 2 - Definição do algoritmo de otimização: Diante da quantidade de algoritmos que podem construir soluções para o problema, neste passo será definido qual o algoritmo deve ser utilizado no experimento.

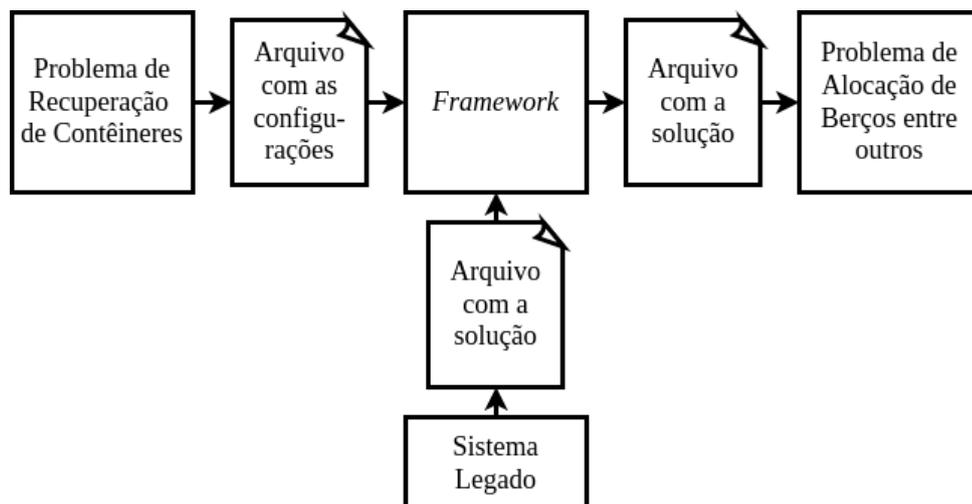
Passo 3 - Definição dos parâmetros e execução dos algoritmos: neste passo, será definido os valores dos parâmetros dos algoritmos, como: quantidade de interações, condição de parada, número de repetições com sementes distintas, entre outros. Em seguida, os algoritmos serão executados conforme os parâmetros definidos.

Passo 4 - Obtenção dos resultados: no último passo, serão apresentados os resultados adicionais, tais como: o comportamento da solução conforme as iterações, os valores obtidos, a média, o desvio padrão, o tempo de execução, a variância, e os valores máximo e mínimo.

4.5.2 Descrição conceitual do *framework*

Para possibilitar a integração do *framework* com sistemas legados, algoritmos implementados em outras linguagens de programação e problemas relacionados, foi necessário investigar uma alternativa para possibilitar a comunicação entre os mesmos. Com a revisão da literatura, observou-se que os trabalhos que abordam o CSP adotam as instâncias utilizadas no trabalho de Meisel e Wichmann (2010). Conforme mostra a Figura 28, o *framework* recebe um arquivo com as configurações que podem utilizar informações geradas por outros problemas, como o Problema de Recuperação de Contêineres. Esse arquivo segue o padrão adotado por Meisel e Wichmann (2010), visto que é utilizado em trabalhos relacionados para fazer comparações.

Figura 28 – Visão geral da comunicação do *framework* com os demais problemas e sistemas legados.



Fonte: Elaborada pelo autor (2022)

Com relação à técnica de otimização, é importante destacar que novos algoritmos podem ser incorporados ao *framework* ou os algoritmos de sistemas legados podem ser integrados ao

framework por meio dos arquivos. Nos dois casos se faz necessário que os algoritmos sejam capazes de realizar a leitura do arquivo de configuração e retornem o arquivo com a solução. De posse do arquivo, será possível visualizar a solução encontrada executando cada operação, possibilitando que os especialistas possam observar a ação gerada após a execução de cada operação.

4.6 ANÁLISE DOS RESULTADOS

Neste último passo foi verificado o comportamento dos algoritmos propostos nesse trabalho, ou seja, a reimplementação do GRASP presente na literatura, da implementação do GRASP+*Deslocamento Direcionado*, do Algoritmo Sub-Listas de Operações e do Algoritmo de Sub-Listas de Operações combinado com o Algoritmo Genético de Chaves Aleatórias Viciadas, em seguida foi realizada a comparação dos resultados obtidos desses algoritmos. Diante do objetivo do trabalho que consiste em propor um algoritmo capaz de reduzir tempo de realização das operações com os contêineres pelo guindaste, então, a comparação dos resultados obtidos nessa etapa será para verificar se os algoritmos propostos têm um desempenho melhor do que o algoritmo encontrado na literatura.

Cada algoritmo foi executado 30 vezes para cada instância de entrada, variando as sementes em cada execução. A geração das sementes foi realizada com base nas casas decimais do número π . Como foi citado na Subseção 4.3, o conjunto de instâncias utilizado foi o do trabalho Meisel e Wichmann (2010). A descrição do ambiente e dos experimentos realizados com cada algoritmo encontra-se no Capítulo 5

4.7 CONSIDERAÇÕES FINAIS

Nesse capítulo da metodologia foi possível identificar os principais passos realizados nesse trabalho, como: a quantidade de trabalhos primários obtidos na revisão da literatura; o conjunto de instâncias adotados nos trabalhos que abordam o problema de sequenciamento de contêiner, bem como, a organização desse conjunto com relação a tamanho e cenário.

Foi possível observar os passos dos algoritmos propostos nesse trabalho, como o Algoritmo de Deslocamento Direcionado que foi combinado com o GRASP, o Algoritmo de Sub-Listas de Operações e o Algoritmo de Sub-Listas de Operações combinado com o Algoritmo Genético de Chaves Aleatórias Viciadas.

Em seguida, foi possível observar a descrição do *framework* para auxiliar na construção de soluções para o CSP. Por fim, uma descrição breve do planejamento dos experimentos para comparar os algoritmos propostos com os demais algoritmos encontrados na literatura.

5 RESULTADOS COMPUTACIONAIS

Nesse capítulo será descrito o ambiente de execução dos experimentos, os resultados obtidos de cada algoritmo e um exemplo de solução no *framework*.

5.1 CONFIGURAÇÃO DO AMBIENTE

Os experimentos realizados nesse trabalho foram executados em um computador Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz x 4, memória de 8 GB e com o sistema operacional Ubuntu 18.04.6 LTS - Versão do Kernel: 5.4.0-124-generic. Os algoritmos foram implementados na linguagem de programação Python, na versão: Python 3.7.3. Para a geração dos números pseudo-aleatórios foi utilizado o algoritmo Mersenne Twister. Para a criação das sementes, foram utilizadas as casas decimais de $\pi = 3,14159265358979323846\dots$, ou seja, a cada 6 casas decimais uma nova semente foi gerada, por exemplo: $s_1 = 141592$, $s_2 = 653589$, $s_3 = 793238$, ... e $s_{30} = 105559$.

5.2 EXECUÇÃO DOS ALGORITMOS

Para verificar o comportamento do algoritmo de otimização, os mesmos receberam como entrada o conjunto de instâncias utilizado no trabalho de Meisel e Wichmann (2010). O nome de cada instância apresenta as informações sobre o número de pilhas e níveis, e a quantidade de contêineres. Por exemplo, o arquivo `n10_I40_E70_R02_001` possui 10 pilhas, 10 níveis e, com relação à quantidade de contêineres: `I40` - informa que essa instância possui 40 contêineres de importação; `E70` - que possui 70 contêineres de exportação; `R02` - 2 contêineres de *reshuffle* e `001` é a variação da instância, ou seja, para essas quantidades de contêineres existem 10 instâncias que variam a posição dos contêineres na baía, dessa forma, o `001` representa a primeira variação. Sendo assim, cada instância foi executada 30 vezes, onde cada execução recebeu como entrada o número de interações, que nesse experimento foi 30, e o valor da semente obtida através das casas decimais de π .

Para este experimento, as instâncias foram selecionadas de forma aleatória e sistemática, ou seja, entre as 10 variações, foram selecionadas 3, sendo uma instância pertencente ao intervalo de `001` a `003`, outra instância de `004` a `007` e a terceira instância de `008` a `010`.

Essa tabela apresenta a síntese dos resultados, ela foi criada para que o capítulo não ficasse extenso, no entanto, no Apêndice A encontram-se os resultados detalhadamente, onde cada linha possui os resultados de uma instância executada 30 vezes. É importante destacar que o conjunto de instâncias de Meisel e Wichmann (2010), possui 990 instâncias, onde cada variação de n possui 330 instâncias distintas, ou seja, $n_{10} + n_{15} + n_{20} = 330 + 330 + 330 = 990$ instâncias, conforme foi descrito na Tabela 7.

Na Tabela 8, é possível identificar os resultados obtidos em cada algoritmo, sendo que cada algoritmo possui duas colunas:

- **Valor:** Média do tempo total de realização da sequência de operações com os contêineres realizada pelo guindaste, em segundos;
- **T:** Média do tempo gasto para executar o algoritmo, em minutos no formato (mm:ss).

Com relação aos algoritmos, nesse experimento foram utilizados e propostos os seguintes algoritmos:

CPLEX: modelagem proposta por Meisel e Wichmann (2010) e reimplementada por Liu et al. (2015);

GRASP*: proposto por Meisel e Wichmann (2010) e reimplementado por Liu et al. (2015);

GRASP: proposto por Meisel e Wichmann (2010) e reimplementado pelo autor dessa pesquisa;

GRASP+DD: proposto neste trabalho;

IRD: proposto por Liu et al. (2015);

SLO: proposto neste trabalho;

SLO+BRKGA: proposto neste trabalho.

Tabela 8 – Resultados obtidos com a realização dos experimentos com os algoritmos.

| Instância | CPLEX | | GRASP* | | GRASP | | GRASP+DD | | IRD | | SLO | | SLO+BRKGA | |
|---------------------|-------|-------|--------|-------|-------|-------|----------|-------|-------|-------|--------------|-------|--------------|-------|
| | Valor | T | Valor | T | Valor | T | Valor | T | Valor | T | Valor | T | Valor | T |
| CSP_n10_I40_E70_R06 | 14270 | 10:00 | 13035 | 02:08 | 14153 | 00:23 | 14142 | 00:22 | 13645 | 00:01 | 13372 | 00:05 | 13352 | 00:27 |
| CSP_n10_I40_E70_R10 | 14170 | 10:00 | 12740 | 06:02 | 15064 | 00:25 | 15029 | 00:23 | 13200 | 00:01 | 13862 | 00:05 | 13836 | 00:28 |
| CSP_n10_I40_E70_R14 | 14200 | 10:00 | 13140 | 09:20 | 16251 | 00:32 | 16169 | 00:27 | 13580 | 00:01 | 14565 | 00:05 | 14449 | 00:29 |
| CSP_n10_I70_E40_R00 | - | - | - | - | 12768 | 00:17 | 12768 | 00:17 | - | - | 12399 | 00:04 | 12390 | 00:25 |
| CSP_n10_I70_E40_R10 | 14050 | 10:00 | 13285 | 02:47 | 14994 | 00:26 | 14971 | 00:24 | 13190 | 00:01 | 13894 | 00:06 | 13725 | 00:28 |
| CSP_n10_I70_E40_R20 | 14190 | 10:00 | 12260 | 01:44 | 17278 | 00:38 | 17127 | 00:32 | 12690 | 00:01 | 15113 | 00:06 | 14953 | 00:31 |
| CSP_n10_I70_E70_R02 | - | 10:00 | 18230 | 07:46 | 16773 | 00:32 | 16773 | 00:32 | 18915 | 00:01 | 15960 | 00:05 | 15852 | 00:30 |
| CSP_n10_I70_E70_R10 | - | 10:00 | 17425 | 04:27 | 18616 | 00:40 | 18589 | 00:37 | 17935 | 00:01 | 17069 | 00:06 | 16940 | 00:32 |
| CSP_n10_I70_E70_R20 | 18990 | 10:00 | 17015 | 01:23 | 20967 | 00:50 | 20886 | 00:44 | 17345 | 00:01 | 18449 | 00:07 | 18288 | 00:35 |
| CSP_n15_I40_E70_R06 | - | 10:00 | 30265 | 06:38 | 32305 | 03:00 | 32287 | 02:55 | 32135 | 00:01 | 29974 | 00:11 | 29906 | 00:59 |
| CSP_n15_I40_E70_R12 | - | 10:00 | 29910 | 04:22 | 35599 | 03:55 | 35499 | 03:36 | 30210 | 00:01 | 31719 | 00:11 | 31643 | 01:03 |
| CSP_n15_I40_E70_R18 | - | 10:00 | 28200 | 04:06 | 38714 | 04:34 | 38534 | 03:51 | 29000 | 00:01 | 33300 | 00:12 | 33109 | 01:06 |
| CSP_n15_I70_E40_R00 | - | - | - | - | 28958 | 02:21 | 28958 | 02:22 | - | - | 27990 | 00:08 | 27990 | 00:55 |
| CSP_n15_I70_E40_R12 | - | 10:00 | 30570 | 06:54 | 35464 | 03:58 | 35381 | 03:26 | 30090 | 00:01 | 31659 | 00:12 | 31409 | 01:04 |
| CSP_n15_I70_E40_R18 | 32090 | 10:00 | 28295 | 03:39 | 38433 | 04:26 | 38175 | 03:42 | 29825 | 00:01 | 33133 | 00:13 | 32864 | 01:07 |
| CSP_n20_I40_E70_R00 | - | - | - | - | 51696 | 10:32 | 51696 | 10:32 | - | - | 49650 | 00:14 | 49643 | 01:37 |
| CSP_n20_I40_E70_R10 | - | 10:00 | 51235 | 06:42 | 61206 | 14:52 | 61132 | 13:40 | 54830 | 00:01 | 54929 | 00:20 | 54653 | 01:50 |
| CSP_n20_I40_E70_R16 | - | 10:00 | 51300 | 11:26 | 66881 | 18:54 | 66727 | 16:34 | 53350 | 00:01 | 57835 | 00:21 | 57490 | 01:56 |
| CSP_n20_I70_E40_R00 | - | - | - | - | 51526 | 14:16 | 51526 | 14:24 | - | - | 49662 | 00:14 | 49670 | 01:39 |
| CSP_n20_I70_E40_R08 | - | 10:00 | 53415 | 04:56 | 59036 | 14:45 | 58721 | 14:27 | 53205 | 00:01 | 53822 | 00:20 | 53547 | 01:49 |
| CSP_n20_I70_E40_R14 | - | 10:00 | 50500 | 02:51 | 64965 | 17:49 | 64443 | 15:21 | 51215 | 00:01 | 56659 | 00:21 | 56297 | 01:55 |
| CSP_n20_I70_E70_R02 | - | 10:00 | 70635 | 00:20 | 67710 | 17:19 | 67698 | 16:43 | 72240 | 00:01 | 63342 | 00:22 | 63108 | 01:57 |
| CSP_n20_I70_E70_R08 | - | 10:00 | 68965 | 09:47 | 73495 | 19:01 | 73495 | 19:35 | 69085 | 00:01 | 66602 | 00:23 | 66237 | 02:04 |

Fonte: Elaborada pelo autor (2023)

GRASP*: resultados obtidos dos experimentos de Liu et al. (2015).

GRASP: resultados obtidos dos experimentos do autor.

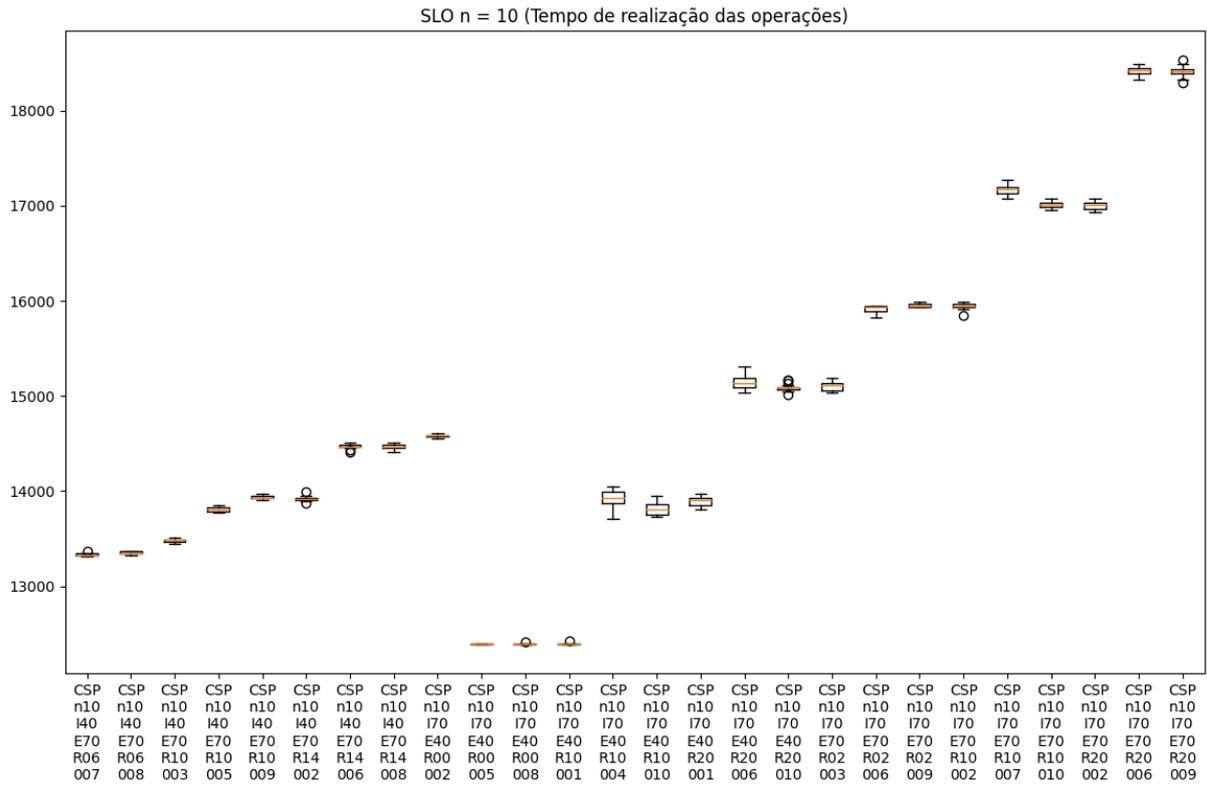
Com os resultados obtidos foi possível identificar que o algoritmo GRASP + *Deslocamento Direcionado*(DD) conseguiu encontrar um tempo total de sequência de operações com os contêineres menor do que o algoritmo GRASP reimplementado nesse trabalho, visto que esse algoritmo busca encontrar operações VV e para isso ele verifica quais operações VB e BV podem ser transformadas em uma operação VV. Assim, ao encontrar uma operação VV o algoritmo DD consegue reduzir o tempo total da realização das operações que foi retornada pelo GRASP.

Com relação ao algoritmo *Internal-Reshuffle-Densed*(IRD) proposto por Liu et al. (2015), o mesmo foi executado em um tempo menor. O CPLEX executado por Liu et al. (2015), teve um tempo limite para ficar processando, os autores definiram que o algoritmo buscaria uma solução até encontrar a mesma ou até completar 10 minutos, sendo assim, é possível notar que algumas instâncias conseguiram chegar em um resultado, já as instâncias com o caractere '-' não conseguiram encontrar uma solução no intervalo de tempo definido. Em seguida, foi realizada a comparação com Algoritmo de Sub-Listas de Operações(SLO). Bem como a versão do SLO+BRKGA.

Com o intuito de observar o comportamento das soluções geradas pelos algoritmos, da Figura 29 até a Figura 40 são apresentados os Box plots das soluções geradas e o tempo de execução de cada algoritmo com base na instância analisada. Na Figura 29, é possível observar que o tempo necessário para o guindaste realizar uma sequência de operações vai aumentando conforme aumenta a quantidade de contêineres de *reshuffle*, para o algoritmo SLO para $n = 10$. Apesar da mudança de cenário, ou seja, a alteração da quantidade de contêiner e importação e exportação, o tempo de execução do algoritmo SLO se manteve reduzida, conforme mostra a Figura 30. O comportamento do SLO com $n = 10$ se repete ao alterar o valor de n , conforme pode ser visto na Figura 31 e Figura 32, para $n = 15$, e na Figura 33 e Figura 34 para $n = 20$.

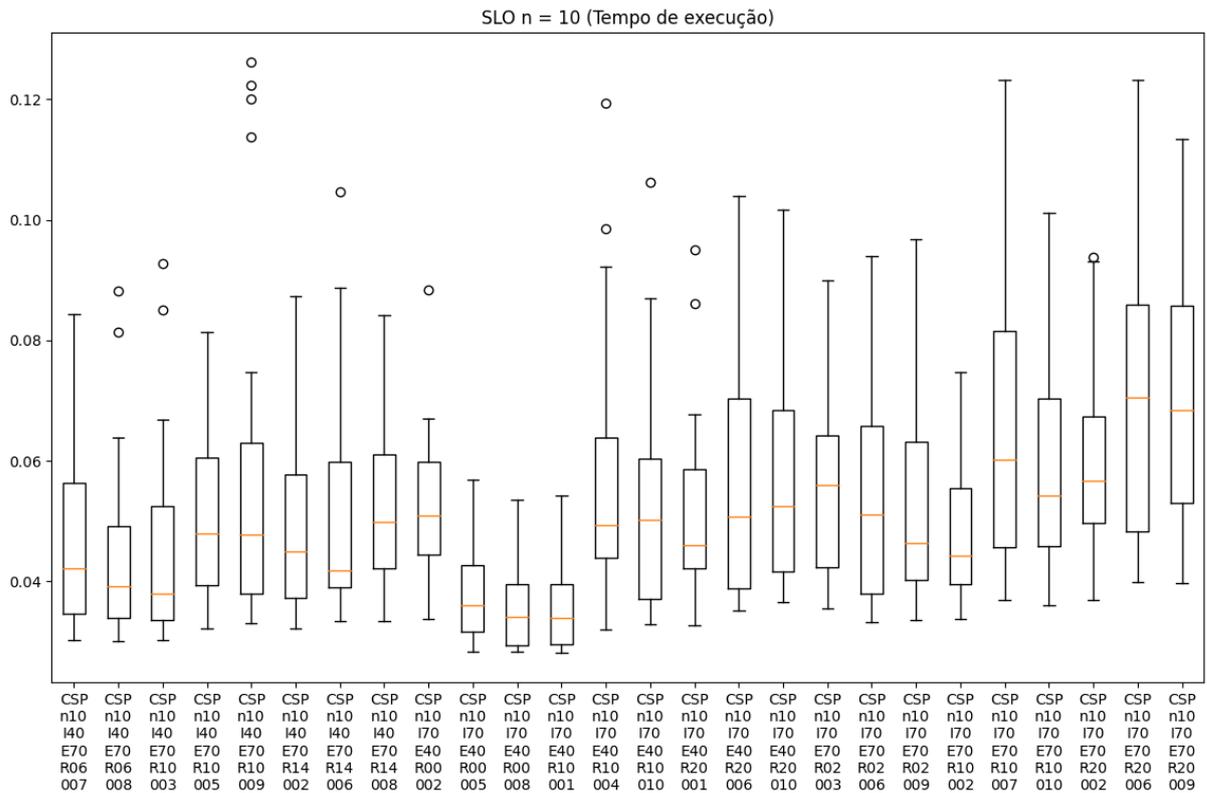
Com relação ao algoritmo BRKGA+SLO, o comportamento se repete, o que já era esperado visto que o BRKGA faz uso do algoritmo SLO, conforme mostra a Figura 35. No entanto, devido a esse processamento adicional referente ao algoritmo do BRKGA, o tempo computacional do algoritmo foi maior, conforme mostra a Figura 36. O mesmo acontece para o BRKGA com $n = 15$, conforme a Figura 37 e a Figura 38, e para o BRKGA com $n = 20$, conforme a Figura 39 e a Figura 40.

Figura 29 – Box plot do tempo de realização das operações do SLO com n = 10.



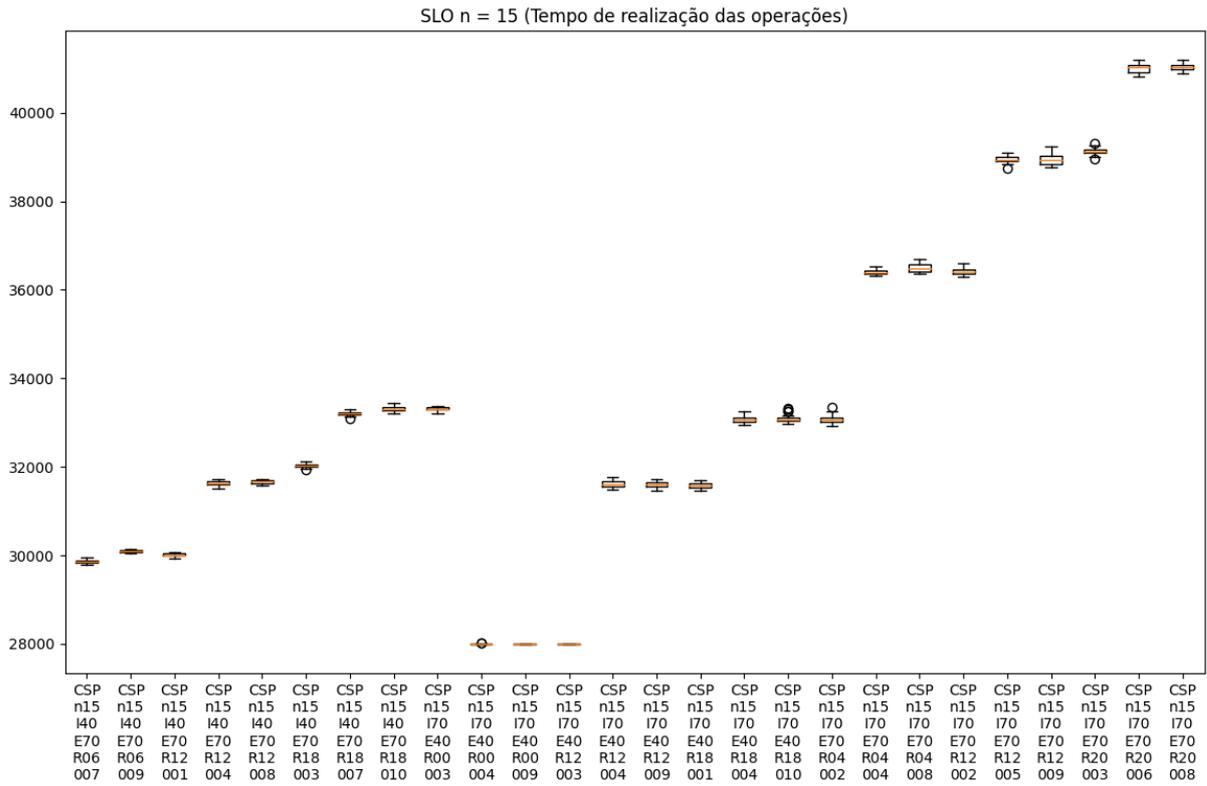
Fonte: Elaborada pelo autor (2023)

Figura 30 – Box plot do tempo de execução do SLO com n = 10.



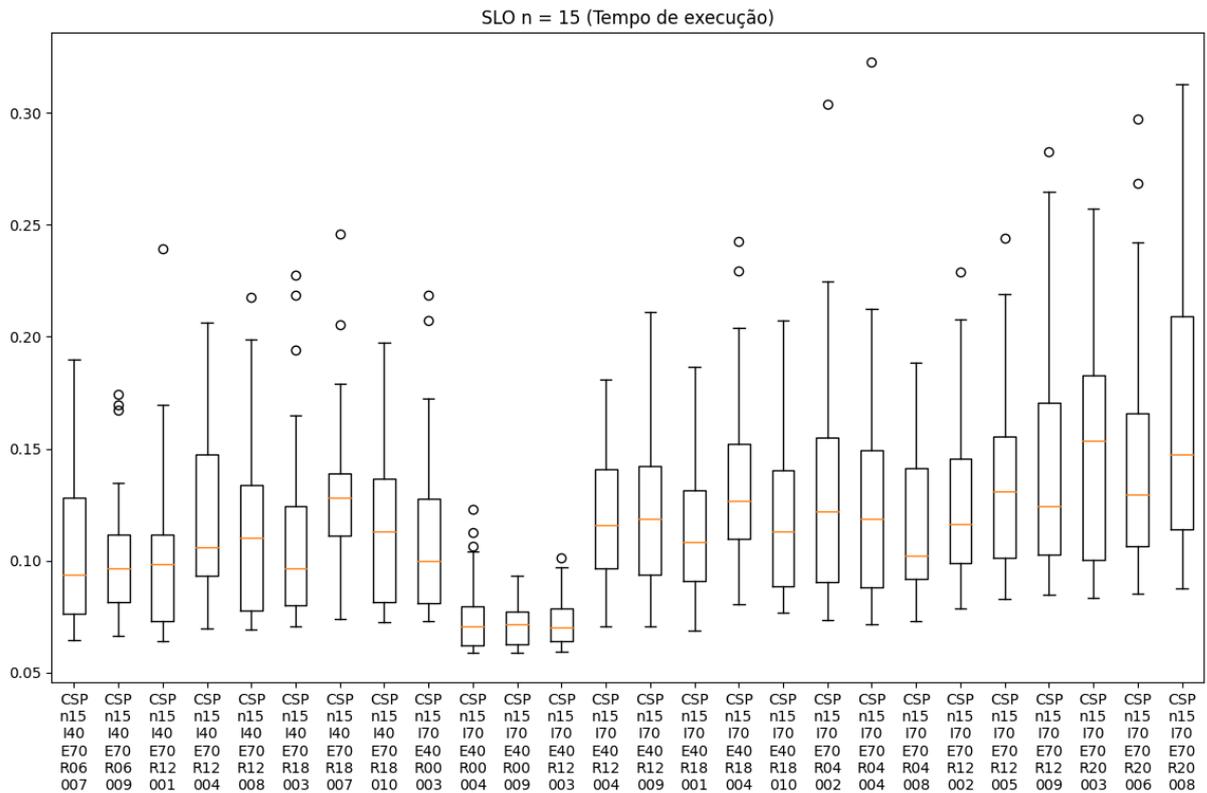
Fonte: Elaborada pelo autor (2023)

Figura 31 – Box plot do tempo de realização das operações do SLO com n = 15.



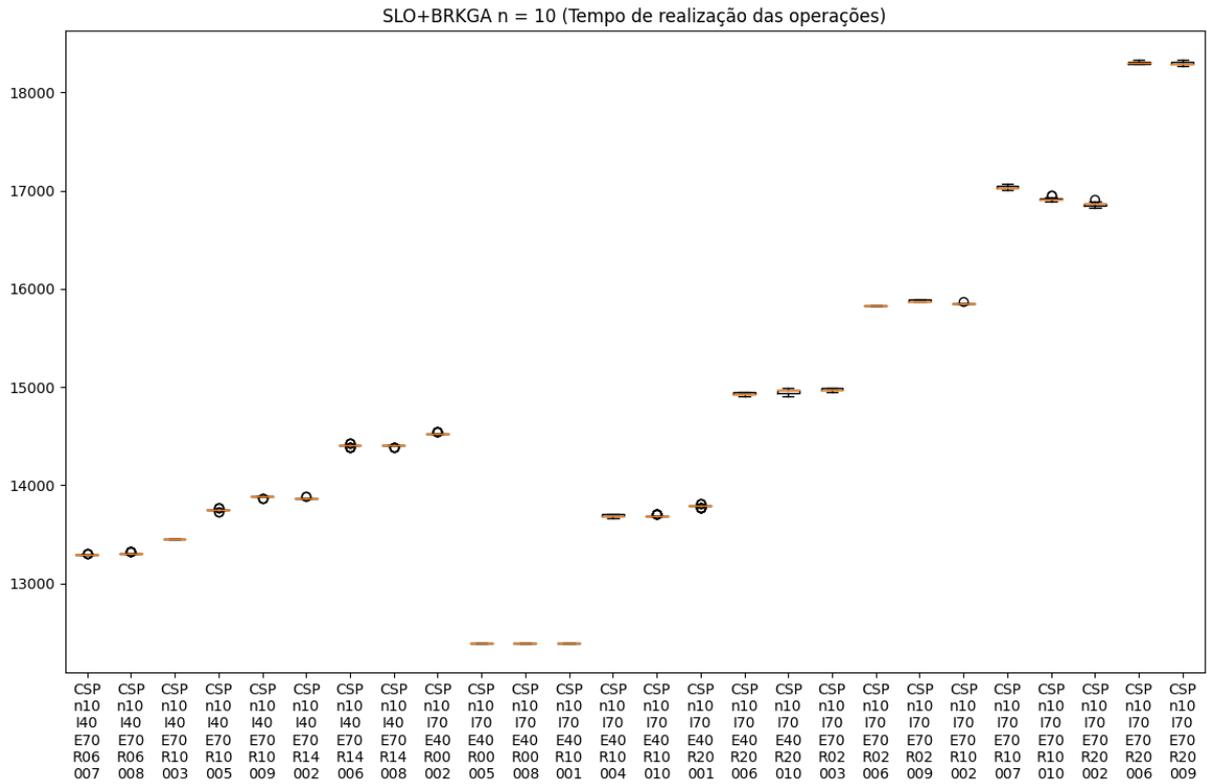
Fonte: Elaborada pelo autor (2023)

Figura 32 – Box plot do tempo de execução do SLO com n = 15.



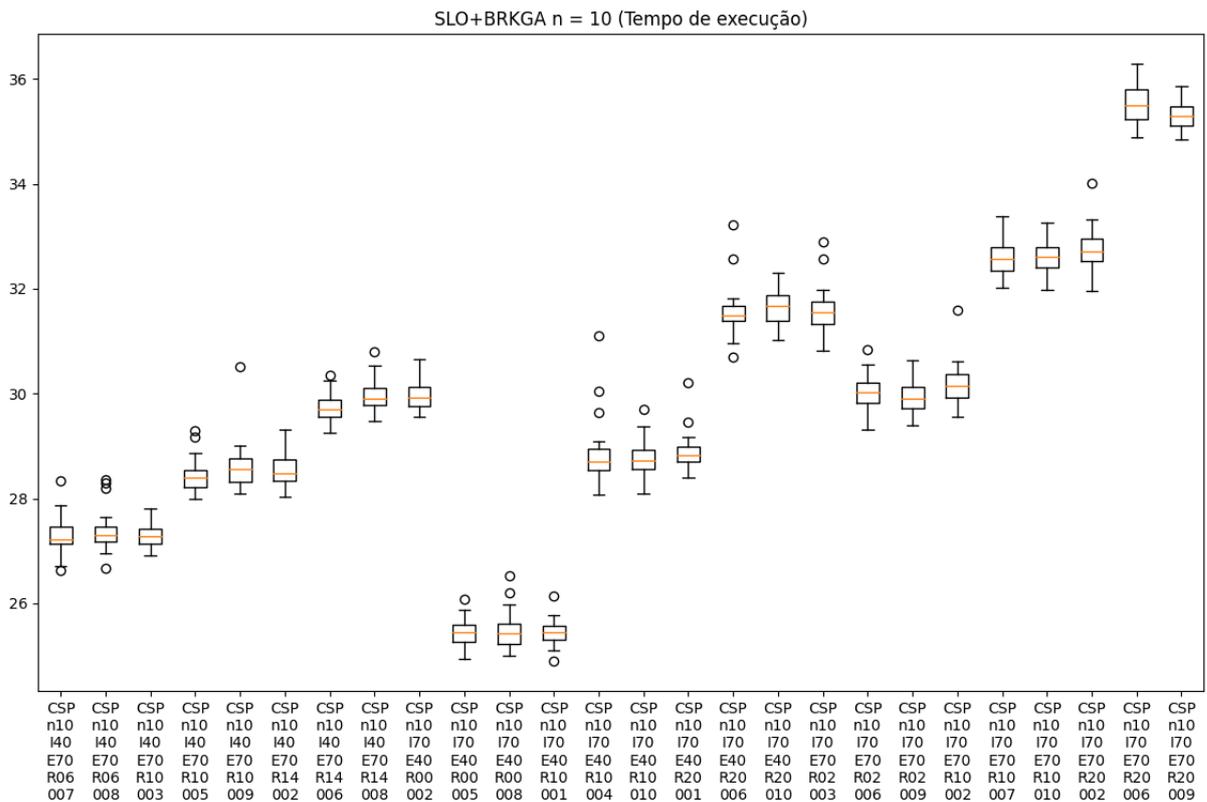
Fonte: Elaborada pelo autor (2023)

Figura 35 – Box plot do tempo de realização das operações do SLO+BRKGA com n = 10.



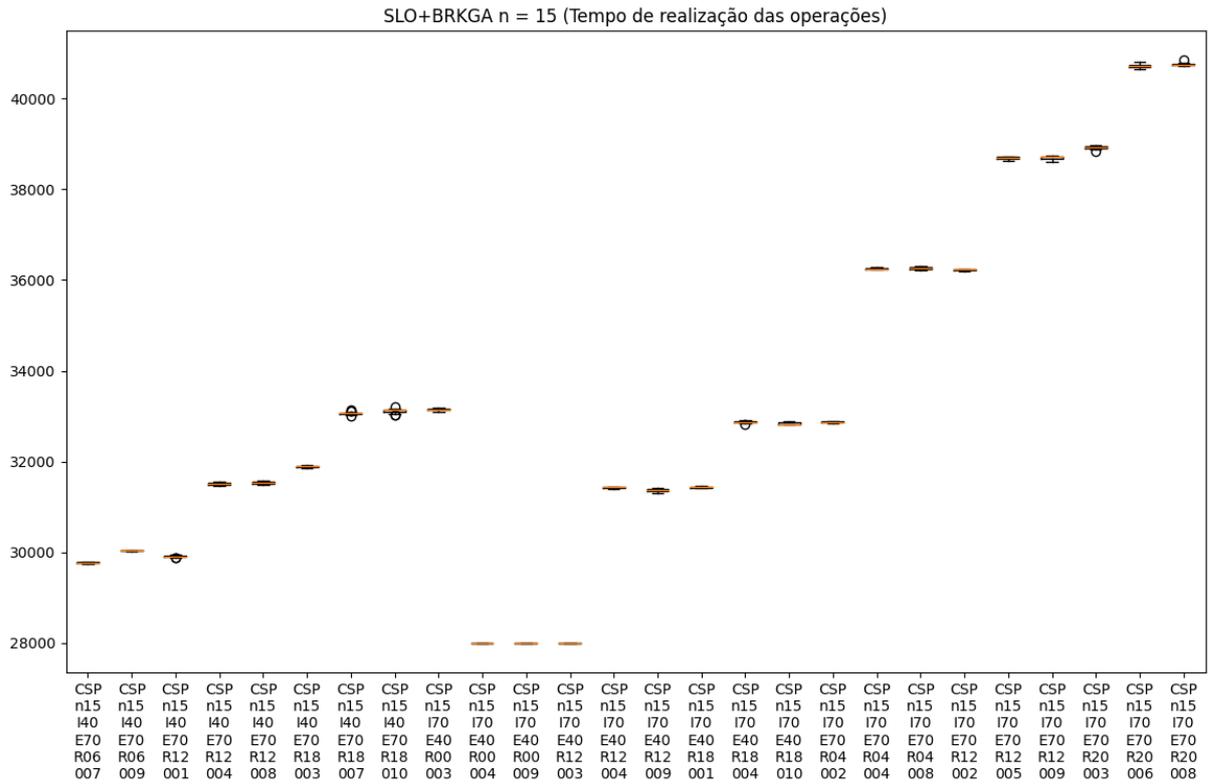
Fonte: Elaborada pelo autor (2023)

Figura 36 – Box plot do tempo de execução do SLO+BRKGA com n = 10.



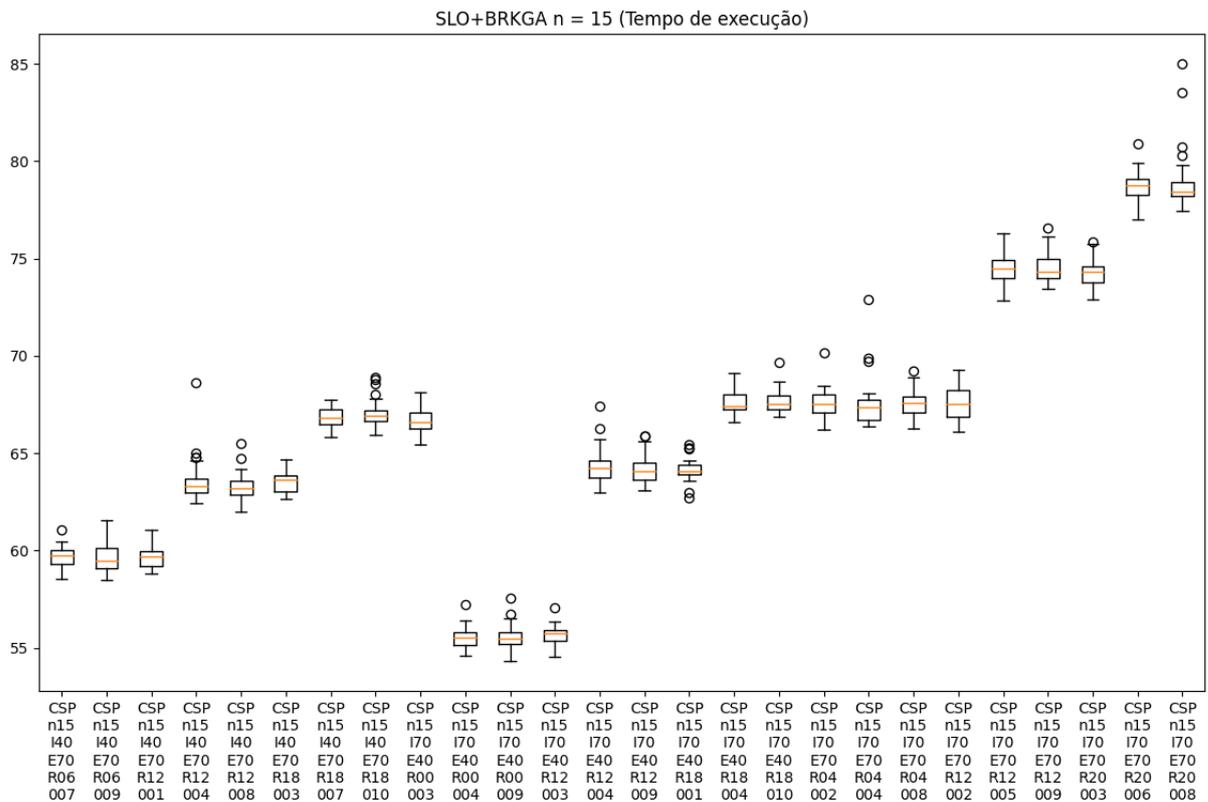
Fonte: Elaborada pelo autor (2023)

Figura 37 – Box plot do tempo de realização das operações do SLO+BRKGA com n = 15.



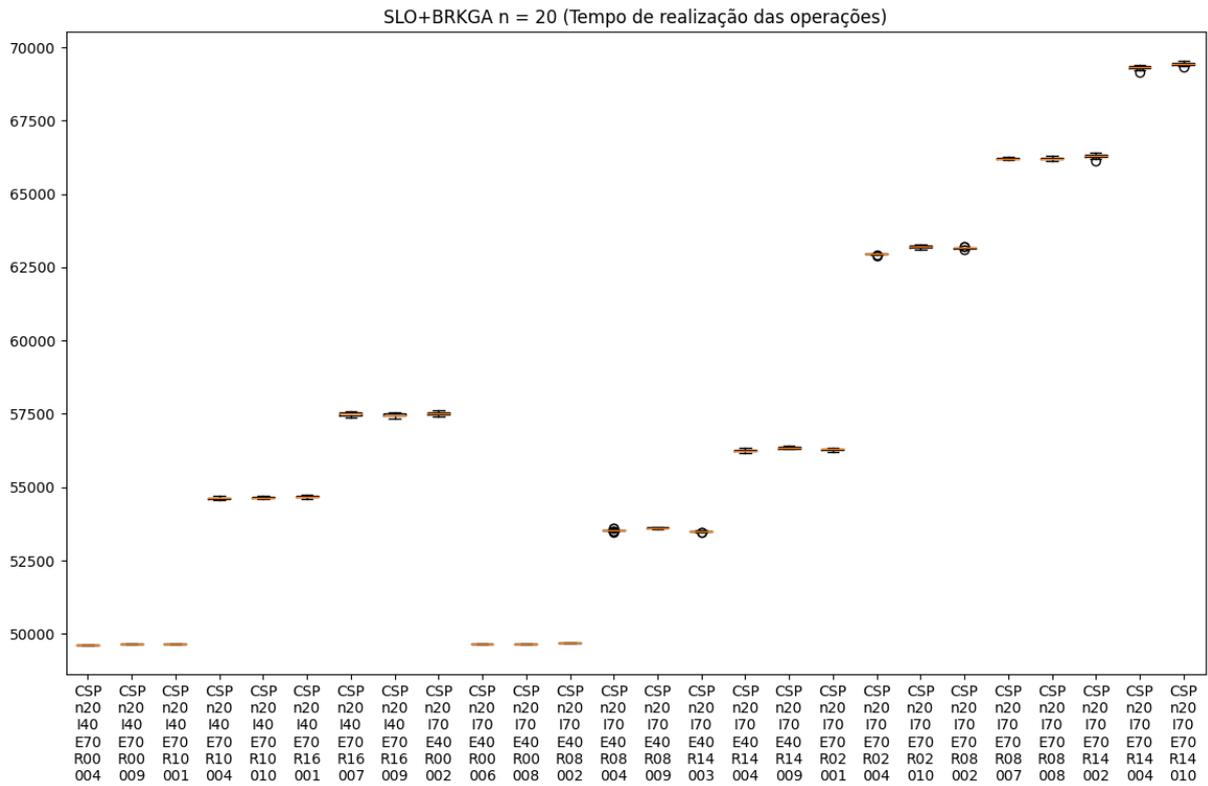
Fonte: Elaborada pelo autor (2023)

Figura 38 – Box plot do tempo de execução do SLO+BRKGA com n = 15.



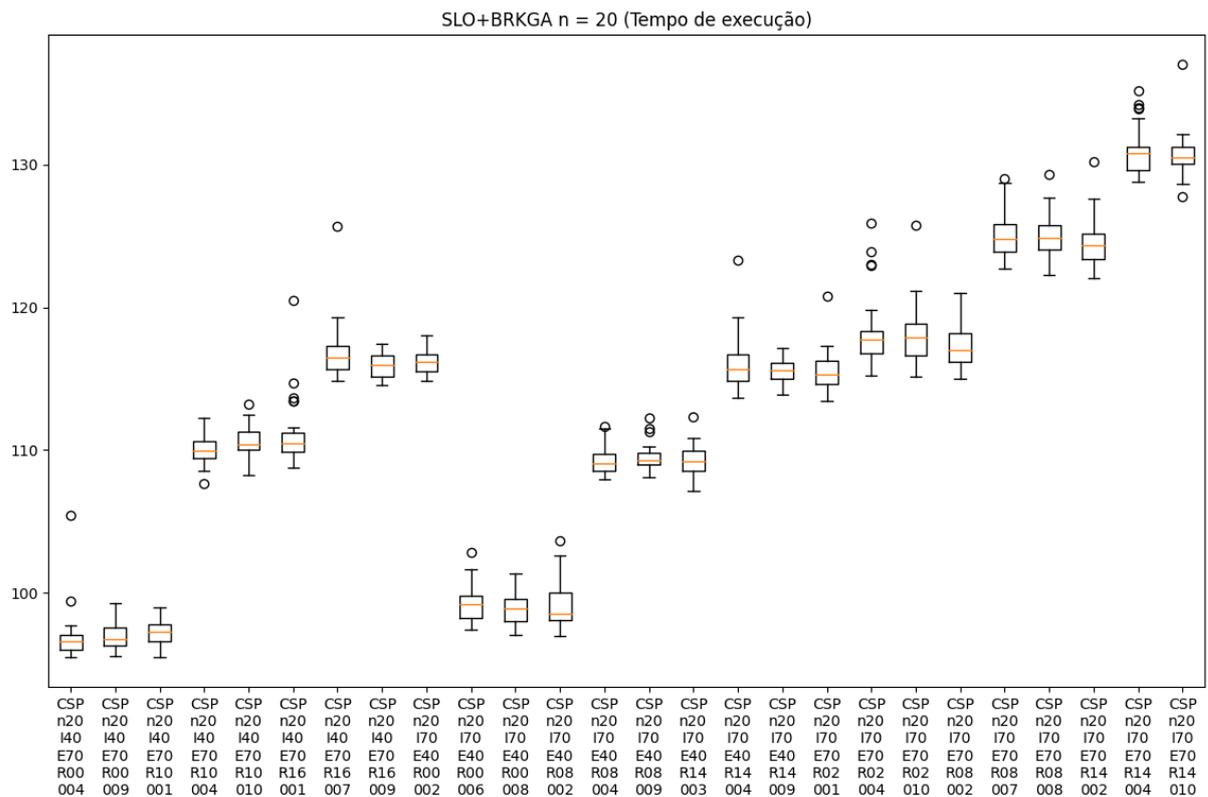
Fonte: Elaborada pelo autor (2023)

Figura 39 – Box plot do tempo de realização das operações do SLO+BRKGA com n = 20.



Fonte: Elaborada pelo autor (2023)

Figura 40 – Box plot do tempo de execução do SLO+BRKGA com n = 20.



Fonte: Elaborada pelo autor (2023)

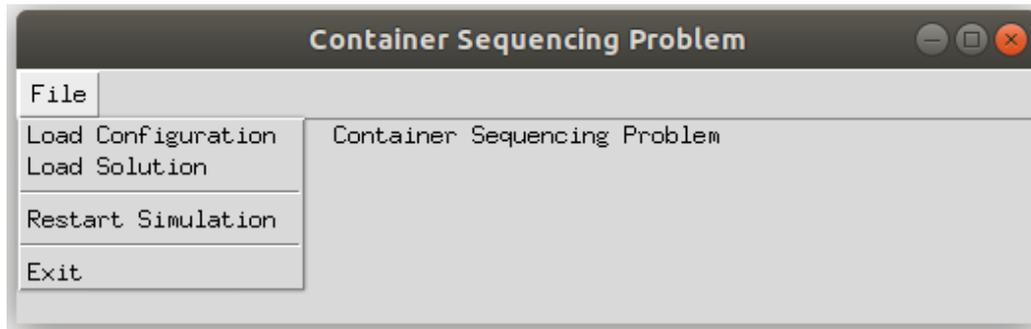
Após a execução dos algoritmos, é possível destacar que o Algoritmo de Deslocamento Direcionado conseguiu melhorar as soluções do GRASP que foi reimplementado nesse trabalho, no entanto, ao realizar a comparação com os algoritmos da literatura o mesmo não conseguiu melhorar os resultados, ou seja, com relação aos algoritmos GRASP* e o IRD. Com relação ao Algoritmo de Sub-Listas de Operações combinado ao Algoritmo Genético de Chaves Aleatórias Viciadas, foi possível notar que ele conseguiu melhorar 34% das instâncias, quando comparado aos algoritmos analisados nesse trabalho.

É importante destacar que o SLO teve um melhor desempenho em instâncias com possibilidades de formação de duplo ciclo, visto que o algoritmo seleciona uma pilha da baía e manipula todos os contêineres da mesma, possibilitando a liberação dessa pilha para que o guindaste possa iniciar a manipulação dos contêineres de exportação, essa por sua vez, forma um duplo ciclo que diminui o tempo total da realização das operações.

5.3 EXEMPLO DE UMA SOLUÇÃO NO *FRAMEWORK*

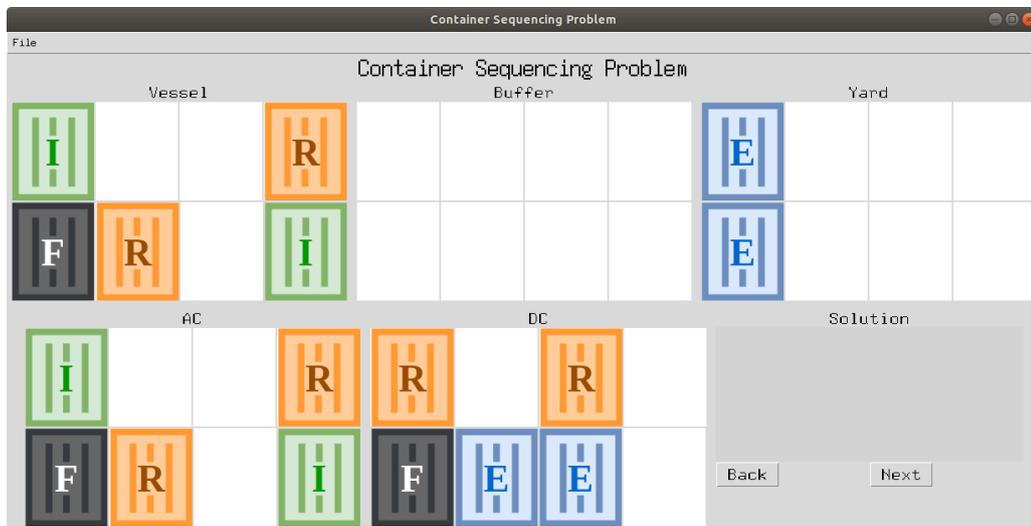
De posse das soluções encontradas é importante obter uma estratégia capaz de auxiliar na visualização da sequência de operações encontrada com o intuito de verificar se a mesma é uma solução viável. Nessa subseção, será visto um exemplo de instância e uma solução para a mesma, assim será possível observar o funcionamento do *framework*. Conforme descrito na Subseção 4.5.1, para criar um experimento no *framework* se faz necessário seguir quatro passos. Primeiro, será carregado o arquivo de configurações, então, neste exemplo será utilizada a instância abordada durante esse trabalho. Na Figura 8, nota-se um exemplo de Configuração Inicial e de Partida dessa instância, que será utilizada nesse exemplo. O arquivo de configuração utilizado com as informações de entrada pode ser observado no Apêndice B.1, que segue a estrutura de instância utilizada por Meisel e Wichmann (2010), visto que, esse padrão de organização das informações no arquivo já é adotado na literatura.

Na Figura 41, é possível observar a tela inicial do *framework* e o menu de opções; no item *LoadConfiguration*, será inserido o arquivo de configuração citado. Ao carregar o arquivo, o *framework* fará a leitura e apresentará para o especialista as informações aplicadas nas principais regiões disponíveis do terminal portuário para esse problema, conforme mostra a Figura 42.

Figura 41 – Tela inicial e menu de opções do *framework*.

Fonte: Elaborada pelo autor (2022)

Figura 42 – Tela de visualização do arquivo de configuração.



Fonte: Elaborada pelo autor (2022)

Ao fazer a leitura do arquivo de configuração, o especialista poderá notar as principais regiões do terminal portuário que estão envolvidas no CSP. A região *Vessel* apresenta o estado atual da baía do navio, então nesse momento que foi inserido apenas o arquivo de configuração, o estado atual é igual à Configuração Inicial, ou seja, o AC, e o objetivo é que após todas as operações o *Vessel* seja alterado até chegar na Configuração de Partida, como é possível observar a região DC. Também é possível observar a região *Buffer* utilizada para realizar as operações com os contêineres de *reshuffle* e a região *Yard* que inicia com todos os contêineres de exportação e após todas as operações terá apenas os contêineres de importação.

Em linhas gerais, as três regiões presentes na parte superior: o *Vessel*, o *Buffer* e o *Yard*, são regiões que vão atualizando conforme as operações com os contêineres são realizadas. Já as regiões na parte inferior: o AC e o DC, permanecem iguais após o arquivo de configuração ser carregado. As regiões AC e DC auxiliam o especialista na verificação da solução, ou seja,

possibilita que o especialista observe se o *Vessel* iniciou com o estado de AC e após todas as operações o *Vessel* terminou com o estado igual a DC.

O próximo passo é a escolha do algoritmo de otimização. Neste trabalho, foi adotado o algoritmo SLO+BRKGA. No passo seguinte, são determinados os valores dos parâmetros do algoritmo. Para esse exemplo, foi definido 100 iterações, com a semente 141592, que se refere às 6 primeiras casas demais do número π . Com a execução do algoritmo, foi gerado o arquivo com a solução conforme pode ser observado no Apêndice B.2. Na Tabela 9 é possível observar essa solução, que é composta por uma sequência de 6 operações, de forma que, ao somar os d^t e d^{tu} de cada operação encontra-se o valor total de 630 segundos que corresponde ao tempo que o guindaste gasta para executar essa solução. De posse do arquivo com a solução, é possível visualizá-la no *framework*, clicando no item *LoadSolution* do menu. Na Figura 43, é possível observar as telas com a solução, ao clicar no botão *Next* cada operação é aplicada e apresentada ao especialista que consegue observar a ação que cada operação está realizando com o contêiner.

Tabela 9 – Tempo gasto em cada operação da solução.

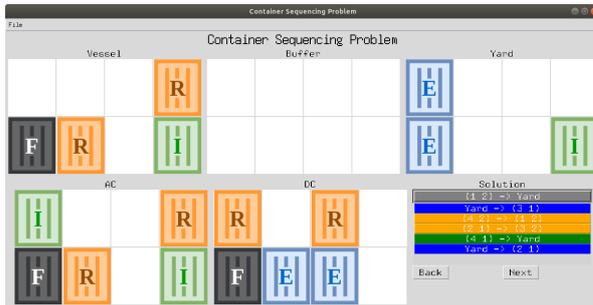
| N | d^{tu} | Origem | Destino | Operação | d^t |
|--------------|----------|--------|---------|----------|------------|
| 1 | - | (1,2) | Yard | VY | 100 |
| 2 | 10 | Yard | (3,1) | YV | 100 |
| 3 | 10 | (4,2) | (1,2) | VV | 90 |
| 4 | 10 | (2,1) | (3,2) | VV | 90 |
| 5 | 10 | (4,1) | Yard | VY | 100 |
| 6 | 10 | Yard | (2,1) | YV | 100 |
| Total | | | | | 630 |

Fonte: Elaborada pelo autor (2023)

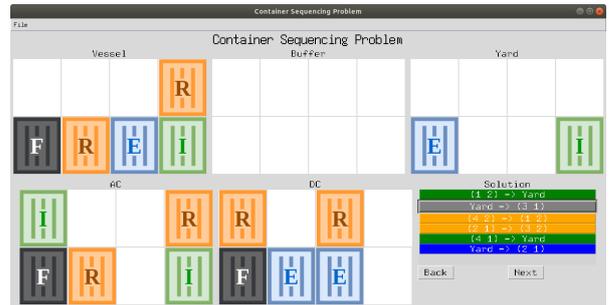
Ao observar da Figura 43a até a Figura 43f, é possível notar os efeitos gerados pela aplicação das operações de 1 até 6, respectivamente, nota-se que as regiões foram atualizadas conforme cada operação. A sequência de operações carregada pelo especialista pode ser observada pelo mesmo no canto inferior direito, na região *Solution*, que possui uma sequência de itens. Quando um item da lista apresentada na região *Solution* fica na cor cinza, isso significa que essa operação foi realizada, adicionalmente é possível observar as próximas operações antes de clicar em *Next*, para facilitar a visualização, cada item possui a cor do tipo de contêiner que a operação movimenta, ou seja, verde para o contêiner de importação, azul para o contêiner de exportação e laranja para o contêiner de *reshuffle*.

Figura 43 – Telas do *Framework* apresentando cada operação da solução encontrada.

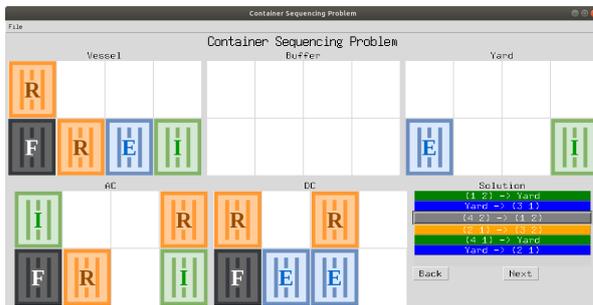
(a) Operação 1.



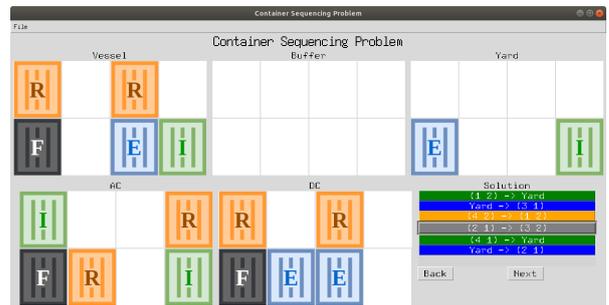
(b) Operação 2.



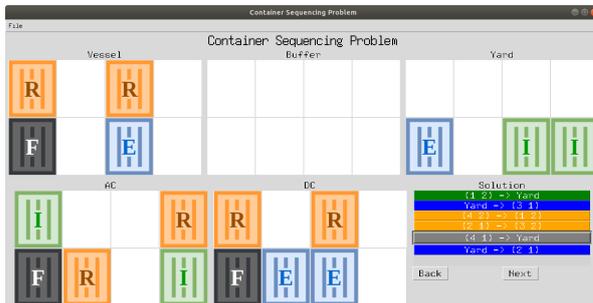
(c) Operação 3.



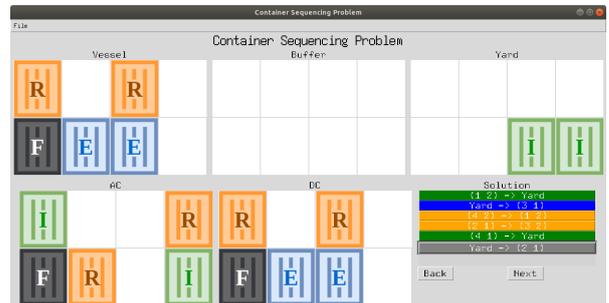
(d) Operação 4.



(e) Operação 5.



(f) Operação 6.



Fonte: Elaborada pelo autor (2022)

Na Figura 43f, nota-se que a solução é viável, pois o *Vessel* possui a mesma configuração de DC. Todos os contêineres de exportação que estavam no *Yard* estão no *Vessel*, todos os contêineres de importação foram transferidos para o *Yard* e nenhum contêiner de *reshuffle* permaneceu no *Buffer*. E durante a aplicação de cada operação os efeitos apresentados são os esperados para essa solução.

5.4 CONSIDERAÇÕES FINAIS

Nessa seção, foi possível observar a descrição dos resultados obtidos com a execução dos algoritmos propostos. Foi possível observar o ambiente de execução utilizado nesse experi-

mento. O conjunto de instâncias adotado, bem como a organização das instâncias e a forma de seleção aleatória e sistemática para realização dos experimentos.

Foi possível comparar os resultados obtidos com os algoritmos propostos, ou seja, o Algoritmo GRASP + Deslocamento Direcionado, o Algoritmo de Sub-Listas de Operações e o Algoritmo de Sub-Listas de Operações + Algoritmo Genético de Chaves Aleatórias Viciadas. Os resultados desses algoritmos foram comparados o aos resultados dos algoritmos encontrados na literatura, ou seja, o GRASP (MEISEL; WICHMANN, 2010), o CPLEX (MEISEL; WICHMANN, 2010) e o IRD (LIU et al., 2015). Em seguida, foi possível verificar uma solução no *framework*, com o intuito de verificar o comportamento da solução encontrada.

6 CONCLUSÃO

Diante do objetivo deste trabalho que consiste em propor um algoritmo para o problema de sequenciamento de contêiner que tem o intuito encontrar uma sequência de operações com os contêineres que serão manipulados pelo guindastes. Nesse trabalho foi possível propor algoritmos que obtiveram uma redução do tempo total das operações.

O Algoritmo de Deslocamento Direcionado, foi identificado após a análise dos resultados do GRASP, visto que seria possível encontrar soluções com um tempo reduzido ao transformar duas operações sendo elas, VB e BV e em uma nova operação VV, assim o algoritmo consegue encontrar uma solução com uma quantidade menor de operações, e por sua vez a operação VV tem um custo menor visto que ela é realizada internamente no navio, sem precisar do auxílio da região do *buffer*.

Outro algoritmo proposto foi de Sub-Listas de Operações que organiza as baias internamente pelas pilhas. De posse de uma pilha, o algoritmo cria uma lista que segue a mesma ordem que os contêineres devem ser manipulado pelo guindaste, assim, essa sequência deve aparecer na mesma ordem na solução final, dessa forma o algoritmo já sabe qual o contêiner está disponível em cada iteração. A ordem de escolha dos contêineres presentes na lista possibilita a criação do duplo ciclo com o guindaste, bem como a liberação de uma pilha, possibilitando que os contêineres que vão entrar no navio sejam manipulados mais rápido, sem precisar que a baia esteja completamente vazia para iniciar o carregamento.

Adicionalmente, foi possível propor o Algoritmo de Sub-Listas de Operações combinado ao Algoritmo Genético de Chaves Aleatórias Viciadas (SLO+BRKGA) que conseguiu reduzir o tempo de realização das operações com os contêineres pelo guindaste. Então, de posse do conjunto de instâncias obtidos da literatura, que possuem uma Configuração de Inicial, o algoritmo proposto nesse trabalho, consegue encontrar uma sequência operações com os contêineres de forma que ao realizar todas as operações determinadas por meio do guindaste a baia resultante é igual à configuração de partida, sendo assim, ao obter o somatório de todos os d^t e d^{tu} o valor encontrado pelo Algoritmo SLO+BRKGA é menor que os algoritmos encontrados na literatura, em 34% das instâncias adotadas nesse trabalho.

O Algoritmo SLO+BRKGA teve melhores resultados nas instâncias que possuíam uma quantidade maior de contêineres de importação e exportação, pois o mesmo conseguiu combinar esses dois tipos de contêineres formando o duplo ciclo. Ao formar o duplo ciclo, o algoritmo

consegue diminuir o tempo total da sequência de operações com os contêineres realizadas pelo guindaste, visto que, ao terminar uma operação, o guindaste ficará na mesma região do que ele precisa se encontrar para iniciar a próxima operação, diminuindo o tempo de movimentação do guindaste sem contêiner.

Durante a revisão da literatura, foi possível identificar os principais trabalhos relacionados, o que possibilitou a construção do *framework*, de forma que ele fosse genérico a ponto de integrar com outras técnicas de otimização ou sistemas legados, além de padronizar os arquivos necessários para a comunicação com outros sistemas. Através do *framework*, os especialistas podem construir suas soluções e verificarem o impacto na solução final, bem como observar a solução de uma forma mais intuitiva. Adicionalmente, o *framework* possibilitou a comunicação com outros problemas de otimização em terminais portuários que podem fornecer novos dados para serem incorporados a novas técnicas de otimização, bem como podem consumir as informações geradas pelo *framework* para usar em outros problemas que tenham ligação com o Problema de Sequenciamento de Contêiner.

Quanto às propostas de trabalhos futuros, pode-se citar:

- Aplicar outras heurísticas no Problema de Sequenciamento de Contêiner para obter melhores resultados;
- Analisar o impacto das soluções geradas pelo *framework* em um contexto global em relação ao terminal portuário;
- Realizar um estudo para verificar a experiência de usuário com os especialistas.

REFERÊNCIAS

- AHMED, E.; EL-ABBASY, M. S.; ZAYED, T.; ALFALAH, G.; ALKASS, S. Synchronized scheduling model for container terminals using simulated double-cycling strategy. *Computers & Industrial Engineering*, Elsevier, v. 154, p. 107118, 2021.
- AVRIEL, M.; PENN, M.; SHPIRER, N.; WITTEBOON, S. Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research*, Springer, v. 76, p. 55–71, 1998.
- AZEVEDO, A.; CHAVES, A.; NETO, L. S. Mini-curso: Introdução à otimização de operações portuárias. 2014.
- BARBOSA, J. P.; CARDOSO, P. A. Modelagem matemática e meta-heurística aplicada à elaboração do plano de estiva em navios porta-contêineres. *32º Congresso de Pesquisa e Ensino em Transportes*, 2018.
- DING, Y.; WEI, X.-J.; YANG, Y.; GU, T.-Y. Decision support based automatic container sequencing system using heuristic rules. *Cluster Computing*, Springer, v. 20, n. 1, p. 239–252, 2017.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.
- FIRMINO, A. d. S. Métodos de otimização aplicados ao problema de recuperação de contêineres. Universidade Federal de Pernambuco, 2019.
- GOLDBARG, E.; GOLDBARG, M.; LUNA, H. *Otimização combinatória e metaheurísticas: algoritmos e aplicações*. [S.l.]: Elsevier Brasil, 2017.
- GUAN, Y.; YANG, K.-H.; ZHOU, Z. The crane scheduling problem: models and solution approaches. *Annals of Operations Research*, Springer, v. 203, n. 1, p. 119–139, 2013.
- HE, X.; WANG, S.; ZHENG, J. A hybrid heuristic algorithm for integrated large-capacity quay crane scheduling problem. In: IEEE. *2011 3rd International Conference on Computer Research and Development*. [S.l.], 2011. v. 1, p. 309–312.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. Citeseer, 2007.
- LEE, C.-Y.; LIU, M.; CHU, C. Optimal algorithm for the general quay crane double-cycling problem. *Transportation Science*, INFORMS, v. 49, n. 4, p. 957–967, 2015.
- LINDEN, R. *Algoritmos genéticos (3ª edição)*. [S.l.]: Brasport, 2012.
- LIU, M.; CHU, F.; ZHANG, Z.; CHU, C. A polynomial-time heuristic for the quay crane double-cycling problem with internal-reshuffling operations. *Transportation Research Part E: Logistics and Transportation Review*, Elsevier, v. 81, p. 52–74, 2015.
- MEISEL, F.; WICHMANN, M. Container sequencing for quay cranes with internal reshuffles. *OR spectrum*, Springer, v. 32, n. 3, p. 569–591, 2010.

-
- SILVA, R. M.; RESENDE, M. G.; PARDALOS, P. M. A python/c++ library for bound-constrained global optimization using a biased random-key genetic algorithm. *Journal of Combinatorial Optimization*, Springer, v. 30, p. 710–728, 2015.
- SILVA, S. D.; TEODORO, P.; FRANCA, L. S.; ALMEIDA, D. Márcio de. Elaboração de jogos de empresa para auxiliar na gestão operacional e ambiental de terminais de contêineres: O caso do jogo tecon-ambiental. *TRANSPORTES*, v. 24, n. 1, p. 31–41, 2016.
- STAHLBOCK, R.; VOSS, S. Operations research at container terminals: a literature update. *OR spectrum*, Springer, v. 30, n. 1, p. 1–52, 2008.
- STEENKEN, D.; VOSS, S.; STAHLBOCK, R. Container terminal operation and operations research—a classification and literature review. *OR spectrum*, Springer, v. 26, n. 1, p. 3–49, 2004.
- ZHANG, H.; KIM, K. H. Maximizing the number of dual-cycle operations of quay cranes in container terminals. *Computers & Industrial Engineering*, Elsevier, v. 56, n. 3, p. 979–992, 2009.
- ZHENG, F.; PANG, Y.; LIU, M.; XU, Y. Dynamic programming algorithms for the general quay crane double-cycling problem with internal-reshuffles. *Journal of Combinatorial Optimization*, Springer, v. 39, n. 3, p. 708–724, 2020.

APÊNDICE A – RESULTADOS COMPUTACIONAIS

Tabela 10 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R00_001 e CSP_n10_I40_E70_R04_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I40_E70_R00_001 | 12411.33 | 00:03 | 4.99 | 12410 | 12430 |
| CSP_n10_I40_E70_R00_002 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I40_E70_R00_003 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I40_E70_R00_004 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I40_E70_R00_005 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I40_E70_R00_006 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I40_E70_R00_007 | 12430.00 | 00:03 | 0.00 | 12430 | 12430 |
| CSP_n10_I40_E70_R00_008 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I40_E70_R00_009 | 12392.00 | 00:03 | 6.00 | 12390 | 12410 |
| CSP_n10_I40_E70_R00_010 | 12390.00 | 00:03 | 0.00 | 12390 | 12390 |
| CSP_n10_I40_E70_R02_001 | 12806.67 | 00:03 | 7.45 | 12790 | 12810 |
| CSP_n10_I40_E70_R02_002 | 12782.67 | 00:04 | 10.93 | 12770 | 12810 |
| CSP_n10_I40_E70_R02_003 | 12798.00 | 00:03 | 12.22 | 12770 | 12810 |
| CSP_n10_I40_E70_R02_004 | 12789.33 | 00:03 | 6.29 | 12770 | 12810 |
| CSP_n10_I40_E70_R02_005 | 12799.33 | 00:03 | 12.36 | 12770 | 12810 |
| CSP_n10_I40_E70_R02_006 | 12890.00 | 00:03 | 0.00 | 12890 | 12890 |
| CSP_n10_I40_E70_R02_007 | 12930.00 | 00:03 | 0.00 | 12930 | 12930 |
| CSP_n10_I40_E70_R02_008 | 12950.00 | 00:03 | 0.00 | 12950 | 12950 |
| CSP_n10_I40_E70_R02_009 | 12751.33 | 00:03 | 4.99 | 12750 | 12770 |
| CSP_n10_I40_E70_R02_010 | 12794.00 | 00:03 | 8.00 | 12790 | 12810 |
| CSP_n10_I40_E70_R04_001 | 12974.00 | 00:03 | 10.83 | 12970 | 13010 |
| CSP_n10_I40_E70_R04_002 | 13090.00 | 00:03 | 0.00 | 13090 | 13090 |
| CSP_n10_I40_E70_R04_003 | 13090.67 | 00:04 | 15.90 | 13070 | 13130 |
| CSP_n10_I40_E70_R04_004 | 12993.33 | 00:04 | 14.68 | 12970 | 13030 |
| CSP_n10_I40_E70_R04_005 | 13129.33 | 00:04 | 13.15 | 13110 | 13150 |
| CSP_n10_I40_E70_R04_006 | 13007.33 | 00:04 | 14.36 | 12990 | 13030 |
| CSP_n10_I40_E70_R04_007 | 13095.33 | 00:04 | 14.54 | 13070 | 13130 |
| CSP_n10_I40_E70_R04_008 | 13119.33 | 00:03 | 11.23 | 13110 | 13150 |
| CSP_n10_I40_E70_R04_009 | 12990.67 | 00:04 | 14.13 | 12950 | 13010 |
| CSP_n10_I40_E70_R04_010 | 12996.67 | 00:04 | 10.75 | 12970 | 13010 |

Fonte: Elaborada pelo autor (2023)

Tabela 11 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R06_001 e CSP_n10_I40_E70_R10_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I40_E70_R06_001 | 13315.33 | 00:04 | 13.60 | 13310 | 13350 |
| CSP_n10_I40_E70_R06_002 | 13383.33 | 00:04 | 13.00 | 13350 | 13410 |
| CSP_n10_I40_E70_R06_003 | 13333.33 | 00:04 | 13.74 | 13310 | 13370 |
| CSP_n10_I40_E70_R06_004 | 13405.33 | 00:04 | 13.35 | 13370 | 13430 |
| CSP_n10_I40_E70_R06_005 | 13364.67 | 00:04 | 13.60 | 13350 | 13390 |
| CSP_n10_I40_E70_R06_006 | 13364.67 | 00:04 | 17.84 | 13330 | 13410 |
| CSP_n10_I40_E70_R06_007 | 13354.67 | 00:04 | 9.91 | 13330 | 13370 |
| CSP_n10_I40_E70_R06_008 | 13482.67 | 00:04 | 12.09 | 13450 | 13510 |
| CSP_n10_I40_E70_R06_009 | 13340.00 | 00:04 | 13.42 | 13310 | 13370 |
| CSP_n10_I40_E70_R06_010 | 13378.67 | 00:04 | 26.67 | 13330 | 13430 |
| CSP_n10_I40_E70_R08_001 | 13552.67 | 00:04 | 12.36 | 13530 | 13570 |
| CSP_n10_I40_E70_R08_002 | 13672.00 | 00:04 | 15.79 | 13630 | 13710 |
| CSP_n10_I40_E70_R08_003 | 13572.67 | 00:04 | 16.92 | 13550 | 13610 |
| CSP_n10_I40_E70_R08_004 | 13817.33 | 00:04 | 13.15 | 13790 | 13830 |
| CSP_n10_I40_E70_R08_005 | 13686.00 | 00:04 | 22.74 | 13630 | 13710 |
| CSP_n10_I40_E70_R08_006 | 13688.67 | 00:05 | 12.58 | 13670 | 13710 |
| CSP_n10_I40_E70_R08_007 | 13601.33 | 00:04 | 14.31 | 13570 | 13630 |
| CSP_n10_I40_E70_R08_008 | 13545.33 | 00:04 | 13.35 | 13530 | 13590 |
| CSP_n10_I40_E70_R08_009 | 13721.33 | 00:05 | 15.22 | 13690 | 13770 |
| CSP_n10_I40_E70_R08_010 | 13574.00 | 00:04 | 18.18 | 13550 | 13610 |
| CSP_n10_I40_E70_R10_001 | 13769.33 | 00:05 | 30.32 | 13730 | 13830 |
| CSP_n10_I40_E70_R10_002 | 13804.67 | 00:05 | 30.08 | 13750 | 13870 |
| CSP_n10_I40_E70_R10_003 | 13809.33 | 00:05 | 23.37 | 13770 | 13850 |
| CSP_n10_I40_E70_R10_004 | 13882.67 | 00:04 | 14.13 | 13870 | 13910 |
| CSP_n10_I40_E70_R10_005 | 13933.33 | 00:05 | 17.19 | 13910 | 13970 |
| CSP_n10_I40_E70_R10_006 | 13796.67 | 00:04 | 14.91 | 13770 | 13830 |
| CSP_n10_I40_E70_R10_007 | 13869.33 | 00:04 | 21.59 | 13830 | 13910 |
| CSP_n10_I40_E70_R10_008 | 13922.00 | 00:04 | 22.27 | 13890 | 13950 |
| CSP_n10_I40_E70_R10_009 | 13922.00 | 00:04 | 21.04 | 13870 | 13990 |
| CSP_n10_I40_E70_R10_010 | 13912.00 | 00:04 | 14.00 | 13890 | 13950 |

Fonte: Elaborada pelo autor (2023)

Tabela 12 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R12_001 e CSP_n10_I40_E70_R16_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I40_E70_R12_001 | 14266.67 | 00:05 | 16.40 | 14230 | 14290 |
| CSP_n10_I40_E70_R12_002 | 14398.67 | 00:04 | 16.88 | 14370 | 14430 |
| CSP_n10_I40_E70_R12_003 | 14284.67 | 00:05 | 17.84 | 14250 | 14310 |
| CSP_n10_I40_E70_R12_004 | 14345.33 | 00:05 | 23.49 | 14290 | 14390 |
| CSP_n10_I40_E70_R12_005 | 14233.33 | 00:04 | 18.68 | 14190 | 14270 |
| CSP_n10_I40_E70_R12_006 | 14358.00 | 00:04 | 21.66 | 14310 | 14410 |
| CSP_n10_I40_E70_R12_007 | 14241.33 | 00:04 | 28.13 | 14170 | 14310 |
| CSP_n10_I40_E70_R12_008 | 14146.67 | 00:05 | 22.56 | 14110 | 14210 |
| CSP_n10_I40_E70_R12_009 | 14431.33 | 00:05 | 18.57 | 14390 | 14470 |
| CSP_n10_I40_E70_R12_010 | 14378.00 | 00:05 | 19.04 | 14350 | 14410 |
| CSP_n10_I40_E70_R14_001 | 14828.00 | 00:05 | 27.01 | 14770 | 14890 |
| CSP_n10_I40_E70_R14_002 | 14472.67 | 00:05 | 19.82 | 14410 | 14510 |
| CSP_n10_I40_E70_R14_003 | 14574.67 | 00:05 | 25.66 | 14510 | 14610 |
| CSP_n10_I40_E70_R14_004 | 14583.33 | 00:05 | 27.00 | 14510 | 14630 |
| CSP_n10_I40_E70_R14_005 | 14597.33 | 00:05 | 28.51 | 14530 | 14630 |
| CSP_n10_I40_E70_R14_006 | 14468.00 | 00:05 | 26.00 | 14410 | 14510 |
| CSP_n10_I40_E70_R14_007 | 14481.33 | 00:04 | 26.67 | 14430 | 14550 |
| CSP_n10_I40_E70_R14_008 | 14586.00 | 00:05 | 18.18 | 14550 | 14610 |
| CSP_n10_I40_E70_R14_009 | 14560.67 | 00:05 | 26.70 | 14510 | 14610 |
| CSP_n10_I40_E70_R14_010 | 14504.00 | 00:05 | 24.85 | 14470 | 14550 |
| CSP_n10_I40_E70_R16_001 | 14562.00 | 00:05 | 19.73 | 14530 | 14610 |
| CSP_n10_I40_E70_R16_002 | 14714.00 | 00:04 | 25.51 | 14650 | 14770 |
| CSP_n10_I40_E70_R16_003 | 14598.67 | 00:05 | 25.66 | 14550 | 14670 |
| CSP_n10_I40_E70_R16_004 | 14611.33 | 00:05 | 35.75 | 14510 | 14670 |
| CSP_n10_I40_E70_R16_005 | 14553.33 | 00:05 | 22.56 | 14510 | 14590 |
| CSP_n10_I40_E70_R16_006 | 14612.00 | 00:05 | 26.00 | 14550 | 14670 |
| CSP_n10_I40_E70_R16_007 | 14721.33 | 00:05 | 24.05 | 14670 | 14770 |
| CSP_n10_I40_E70_R16_008 | 14732.00 | 00:05 | 22.72 | 14690 | 14790 |
| CSP_n10_I40_E70_R16_009 | 14703.33 | 00:05 | 23.85 | 14650 | 14750 |
| CSP_n10_I40_E70_R16_010 | 14602.00 | 00:05 | 19.04 | 14550 | 14630 |

Fonte: Elaborada pelo autor (2023)

Tabela 13 – Resultados obtidos do SLO entre as instâncias CSP_n10_I40_E70_R18_001 e CSP_n10_I70_E40_R00_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I40_E70_R18_001 | 14914.00 | 00:06 | 26.03 | 14870 | 14970 |
| CSP_n10_I40_E70_R18_002 | 14910.67 | 00:05 | 28.98 | 14830 | 14970 |
| CSP_n10_I40_E70_R18_003 | 14909.33 | 00:05 | 27.07 | 14850 | 14950 |
| CSP_n10_I40_E70_R18_004 | 14786.00 | 00:05 | 24.98 | 14730 | 14830 |
| CSP_n10_I40_E70_R18_005 | 14832.00 | 00:05 | 30.27 | 14770 | 14890 |
| CSP_n10_I40_E70_R18_006 | 15048.67 | 00:05 | 18.57 | 15010 | 15090 |
| CSP_n10_I40_E70_R18_007 | 14828.67 | 00:05 | 21.87 | 14770 | 14850 |
| CSP_n10_I40_E70_R18_008 | 15044.67 | 00:05 | 27.29 | 14970 | 15090 |
| CSP_n10_I40_E70_R18_009 | 14938.00 | 00:05 | 22.86 | 14890 | 14990 |
| CSP_n10_I40_E70_R18_010 | 15066.67 | 00:06 | 19.38 | 15030 | 15110 |
| CSP_n10_I40_E70_R20_001 | 15183.33 | 00:05 | 32.39 | 15110 | 15250 |
| CSP_n10_I40_E70_R20_002 | 15270.67 | 00:05 | 25.02 | 15210 | 15330 |
| CSP_n10_I40_E70_R20_003 | 15114.00 | 00:05 | 22.74 | 15070 | 15170 |
| CSP_n10_I40_E70_R20_004 | 15039.33 | 00:05 | 28.16 | 14950 | 15090 |
| CSP_n10_I40_E70_R20_005 | 15043.33 | 00:05 | 20.22 | 14990 | 15090 |
| CSP_n10_I40_E70_R20_006 | 15082.00 | 00:05 | 26.13 | 15030 | 15130 |
| CSP_n10_I40_E70_R20_007 | 15060.00 | 00:05 | 24.63 | 15010 | 15110 |
| CSP_n10_I40_E70_R20_008 | 15119.33 | 00:05 | 28.63 | 15050 | 15190 |
| CSP_n10_I40_E70_R20_009 | 15266.67 | 00:05 | 31.45 | 15210 | 15370 |
| CSP_n10_I40_E70_R20_010 | 15135.33 | 00:05 | 23.06 | 15090 | 15190 |
| CSP_n10_I70_E40_R00_001 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I70_E40_R00_002 | 12390.00 | 00:03 | 0.00 | 12390 | 12390 |
| CSP_n10_I70_E40_R00_003 | 12392.67 | 00:03 | 11.23 | 12390 | 12450 |
| CSP_n10_I70_E40_R00_004 | 12410.00 | 00:03 | 0.00 | 12410 | 12410 |
| CSP_n10_I70_E40_R00_005 | 12390.67 | 00:03 | 3.59 | 12390 | 12410 |
| CSP_n10_I70_E40_R00_006 | 12411.33 | 00:03 | 4.99 | 12410 | 12430 |
| CSP_n10_I70_E40_R00_007 | 12410.67 | 00:03 | 3.59 | 12410 | 12430 |
| CSP_n10_I70_E40_R00_008 | 12391.33 | 00:03 | 7.18 | 12390 | 12430 |
| CSP_n10_I70_E40_R00_009 | 12391.33 | 00:03 | 7.18 | 12390 | 12430 |
| CSP_n10_I70_E40_R00_010 | 12390.67 | 00:03 | 3.59 | 12390 | 12410 |

Fonte: Elaborada pelo autor (2023)

Tabela 14 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R02_001 e CSP_n10_I70_E40_R06_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E40_R02_001 | 12762.67 | 00:04 | 14.13 | 12730 | 12770 |
| CSP_n10_I70_E40_R02_002 | 12740.00 | 00:04 | 13.42 | 12730 | 12790 |
| CSP_n10_I70_E40_R02_003 | 12752.00 | 00:04 | 27.01 | 12730 | 12810 |
| CSP_n10_I70_E40_R02_004 | 12870.00 | 00:03 | 0.00 | 12870 | 12870 |
| CSP_n10_I70_E40_R02_005 | 12861.33 | 00:04 | 21.09 | 12850 | 12930 |
| CSP_n10_I70_E40_R02_006 | 12770.67 | 00:03 | 3.59 | 12770 | 12790 |
| CSP_n10_I70_E40_R02_007 | 12752.00 | 00:04 | 6.00 | 12750 | 12770 |
| CSP_n10_I70_E40_R02_008 | 12752.00 | 00:04 | 27.01 | 12730 | 12810 |
| CSP_n10_I70_E40_R02_009 | 12770.67 | 00:03 | 3.59 | 12770 | 12790 |
| CSP_n10_I70_E40_R02_010 | 12755.33 | 00:04 | 10.24 | 12750 | 12790 |
| CSP_n10_I70_E40_R04_001 | 13096.67 | 00:04 | 13.98 | 13070 | 13130 |
| CSP_n10_I70_E40_R04_002 | 13090.67 | 00:03 | 3.59 | 13090 | 13110 |
| CSP_n10_I70_E40_R04_003 | 13160.00 | 00:05 | 38.21 | 13090 | 13230 |
| CSP_n10_I70_E40_R04_004 | 13108.67 | 00:04 | 13.60 | 13090 | 13150 |
| CSP_n10_I70_E40_R04_005 | 12996.00 | 00:05 | 24.30 | 12970 | 13070 |
| CSP_n10_I70_E40_R04_006 | 13005.33 | 00:04 | 16.88 | 12970 | 13030 |
| CSP_n10_I70_E40_R04_007 | 13134.67 | 00:04 | 37.12 | 13090 | 13250 |
| CSP_n10_I70_E40_R04_008 | 13161.33 | 00:04 | 43.42 | 13110 | 13250 |
| CSP_n10_I70_E40_R04_009 | 12965.33 | 00:05 | 24.59 | 12930 | 13050 |
| CSP_n10_I70_E40_R04_010 | 13094.67 | 00:04 | 17.65 | 13070 | 13150 |
| CSP_n10_I70_E40_R06_001 | 13245.33 | 00:05 | 51.04 | 13170 | 13350 |
| CSP_n10_I70_E40_R06_002 | 13374.67 | 00:04 | 20.45 | 13330 | 13410 |
| CSP_n10_I70_E40_R06_003 | 13328.67 | 00:04 | 33.04 | 13290 | 13430 |
| CSP_n10_I70_E40_R06_004 | 13364.00 | 00:04 | 56.13 | 13270 | 13490 |
| CSP_n10_I70_E40_R06_005 | 13435.33 | 00:04 | 24.73 | 13410 | 13510 |
| CSP_n10_I70_E40_R06_006 | 13380.67 | 00:04 | 52.85 | 13270 | 13490 |
| CSP_n10_I70_E40_R06_007 | 13309.33 | 00:05 | 25.02 | 13250 | 13370 |
| CSP_n10_I70_E40_R06_008 | 13443.33 | 00:05 | 58.73 | 13350 | 13550 |
| CSP_n10_I70_E40_R06_009 | 13318.00 | 00:05 | 35.63 | 13250 | 13390 |
| CSP_n10_I70_E40_R06_010 | 13380.67 | 00:04 | 52.85 | 13270 | 13490 |

Fonte: Elaborada pelo autor (2023)

Tabela 15 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R08_001 e CSP_n10_I70_E40_R12_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E40_R08_001 | 13561.33 | 00:04 | 43.72 | 13510 | 13650 |
| CSP_n10_I70_E40_R08_002 | 13567.33 | 00:05 | 47.25 | 13510 | 13690 |
| CSP_n10_I70_E40_R08_003 | 13563.33 | 00:05 | 53.25 | 13490 | 13690 |
| CSP_n10_I70_E40_R08_004 | 13672.00 | 00:04 | 47.15 | 13570 | 13770 |
| CSP_n10_I70_E40_R08_005 | 13561.33 | 00:05 | 36.03 | 13510 | 13630 |
| CSP_n10_I70_E40_R08_006 | 13590.67 | 00:04 | 49.12 | 13510 | 13710 |
| CSP_n10_I70_E40_R08_007 | 13582.67 | 00:05 | 43.66 | 13530 | 13730 |
| CSP_n10_I70_E40_R08_008 | 13560.67 | 00:05 | 22.35 | 13510 | 13590 |
| CSP_n10_I70_E40_R08_009 | 13573.33 | 00:05 | 31.02 | 13530 | 13650 |
| CSP_n10_I70_E40_R08_010 | 13575.33 | 00:04 | 32.63 | 13510 | 13650 |
| CSP_n10_I70_E40_R10_001 | 13928.00 | 00:05 | 81.38 | 13710 | 14050 |
| CSP_n10_I70_E40_R10_002 | 13788.00 | 00:05 | 43.62 | 13710 | 13870 |
| CSP_n10_I70_E40_R10_003 | 13776.00 | 00:05 | 47.09 | 13690 | 13890 |
| CSP_n10_I70_E40_R10_004 | 13811.33 | 00:05 | 59.76 | 13730 | 13950 |
| CSP_n10_I70_E40_R10_005 | 13914.67 | 00:06 | 40.56 | 13850 | 14010 |
| CSP_n10_I70_E40_R10_006 | 14010.67 | 00:05 | 39.15 | 13910 | 14070 |
| CSP_n10_I70_E40_R10_007 | 13962.67 | 00:05 | 30.32 | 13910 | 14050 |
| CSP_n10_I70_E40_R10_008 | 14047.33 | 00:05 | 36.05 | 13970 | 14130 |
| CSP_n10_I70_E40_R10_009 | 13815.33 | 00:05 | 37.21 | 13750 | 13890 |
| CSP_n10_I70_E40_R10_010 | 13892.00 | 00:05 | 45.42 | 13810 | 13970 |
| CSP_n10_I70_E40_R12_001 | 14346.00 | 00:05 | 47.44 | 14270 | 14430 |
| CSP_n10_I70_E40_R12_002 | 14344.00 | 00:05 | 25.38 | 14290 | 14410 |
| CSP_n10_I70_E40_R12_003 | 14230.67 | 00:06 | 38.12 | 14170 | 14330 |
| CSP_n10_I70_E40_R12_004 | 14350.00 | 00:05 | 41.95 | 14270 | 14410 |
| CSP_n10_I70_E40_R12_005 | 14342.00 | 00:05 | 41.18 | 14270 | 14430 |
| CSP_n10_I70_E40_R12_006 | 14296.67 | 00:05 | 44.22 | 14210 | 14410 |
| CSP_n10_I70_E40_R12_007 | 14174.00 | 00:05 | 56.90 | 14070 | 14290 |
| CSP_n10_I70_E40_R12_008 | 14316.67 | 00:05 | 31.55 | 14250 | 14410 |
| CSP_n10_I70_E40_R12_009 | 14226.67 | 00:05 | 54.18 | 14130 | 14310 |
| CSP_n10_I70_E40_R12_010 | 14236.67 | 00:05 | 33.20 | 14170 | 14310 |

Fonte: Elaborada pelo autor (2023)

Tabela 16 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R14_001 e CSP_n10_I70_E40_R18_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E40_R14_001 | 14660.00 | 00:05 | 36.79 | 14590 | 14730 |
| CSP_n10_I70_E40_R14_002 | 14464.00 | 00:05 | 36.93 | 14370 | 14530 |
| CSP_n10_I70_E40_R14_003 | 14438.00 | 00:05 | 33.70 | 14390 | 14530 |
| CSP_n10_I70_E40_R14_004 | 14438.67 | 00:05 | 37.48 | 14370 | 14550 |
| CSP_n10_I70_E40_R14_005 | 14456.67 | 00:05 | 51.21 | 14390 | 14570 |
| CSP_n10_I70_E40_R14_006 | 14668.00 | 00:05 | 40.45 | 14590 | 14750 |
| CSP_n10_I70_E40_R14_007 | 14350.67 | 00:05 | 52.02 | 14250 | 14470 |
| CSP_n10_I70_E40_R14_008 | 14424.67 | 00:05 | 33.44 | 14370 | 14510 |
| CSP_n10_I70_E40_R14_009 | 14431.33 | 00:06 | 43.49 | 14370 | 14530 |
| CSP_n10_I70_E40_R14_010 | 14439.33 | 00:05 | 47.81 | 14330 | 14550 |
| CSP_n10_I70_E40_R16_001 | 14658.67 | 00:05 | 45.51 | 14590 | 14770 |
| CSP_n10_I70_E40_R16_002 | 14816.67 | 00:04 | 33.60 | 14750 | 14890 |
| CSP_n10_I70_E40_R16_003 | 14540.67 | 00:05 | 43.43 | 14450 | 14650 |
| CSP_n10_I70_E40_R16_004 | 14681.33 | 00:05 | 38.88 | 14610 | 14750 |
| CSP_n10_I70_E40_R16_005 | 14660.00 | 00:04 | 39.58 | 14590 | 14770 |
| CSP_n10_I70_E40_R16_006 | 14555.33 | 00:05 | 45.59 | 14470 | 14630 |
| CSP_n10_I70_E40_R16_007 | 14556.67 | 00:06 | 46.86 | 14470 | 14670 |
| CSP_n10_I70_E40_R16_008 | 14668.00 | 00:06 | 43.92 | 14570 | 14750 |
| CSP_n10_I70_E40_R16_009 | 14693.33 | 00:05 | 35.06 | 14610 | 14770 |
| CSP_n10_I70_E40_R16_010 | 14676.67 | 00:05 | 60.74 | 14570 | 14810 |
| CSP_n10_I70_E40_R18_001 | 14875.33 | 00:05 | 55.84 | 14770 | 14990 |
| CSP_n10_I70_E40_R18_002 | 14876.67 | 00:05 | 43.31 | 14770 | 14950 |
| CSP_n10_I70_E40_R18_003 | 14882.00 | 00:06 | 40.20 | 14810 | 14950 |
| CSP_n10_I70_E40_R18_004 | 14991.33 | 00:05 | 47.03 | 14870 | 15050 |
| CSP_n10_I70_E40_R18_005 | 14882.00 | 00:05 | 33.31 | 14830 | 14970 |
| CSP_n10_I70_E40_R18_006 | 14765.33 | 00:05 | 55.30 | 14670 | 14890 |
| CSP_n10_I70_E40_R18_007 | 14756.00 | 00:06 | 53.95 | 14690 | 14950 |
| CSP_n10_I70_E40_R18_008 | 14884.00 | 00:05 | 44.47 | 14790 | 14950 |
| CSP_n10_I70_E40_R18_009 | 14776.00 | 00:06 | 60.48 | 14650 | 14930 |
| CSP_n10_I70_E40_R18_010 | 14876.00 | 00:05 | 46.52 | 14790 | 14970 |

Fonte: Elaborada pelo autor (2023)

Tabela 17 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E40_R20_001 e CSP_n10_I70_E70_R02_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E40_R20_001 | 15143.33 | 00:05 | 66.20 | 15030 | 15310 |
| CSP_n10_I70_E40_R20_002 | 15100.00 | 00:05 | 49.46 | 14990 | 15170 |
| CSP_n10_I70_E40_R20_003 | 15100.67 | 00:06 | 53.35 | 15010 | 15190 |
| CSP_n10_I70_E40_R20_004 | 15020.67 | 00:05 | 35.30 | 14930 | 15110 |
| CSP_n10_I70_E40_R20_005 | 15196.00 | 00:05 | 32.72 | 15130 | 15270 |
| CSP_n10_I70_E40_R20_006 | 15087.33 | 00:05 | 31.30 | 15010 | 15170 |
| CSP_n10_I70_E40_R20_007 | 15088.67 | 00:05 | 33.84 | 15010 | 15150 |
| CSP_n10_I70_E40_R20_008 | 15202.00 | 00:05 | 45.49 | 15110 | 15310 |
| CSP_n10_I70_E40_R20_009 | 15091.33 | 00:06 | 60.21 | 14970 | 15210 |
| CSP_n10_I70_E40_R20_010 | 15100.67 | 00:05 | 48.09 | 15030 | 15190 |
| CSP_n10_I70_E70_R00_001 | 15490.67 | 00:04 | 19.65 | 15470 | 15530 |
| CSP_n10_I70_E70_R00_002 | 15484.67 | 00:04 | 20.61 | 15450 | 15510 |
| CSP_n10_I70_E70_R00_003 | 15494.67 | 00:04 | 11.18 | 15490 | 15530 |
| CSP_n10_I70_E70_R00_004 | 15506.00 | 00:04 | 9.52 | 15490 | 15530 |
| CSP_n10_I70_E70_R00_005 | 15490.67 | 00:04 | 18.96 | 15470 | 15530 |
| CSP_n10_I70_E70_R00_006 | 15499.33 | 00:04 | 9.98 | 15490 | 15510 |
| CSP_n10_I70_E70_R00_007 | 15499.33 | 00:04 | 11.23 | 15490 | 15530 |
| CSP_n10_I70_E70_R00_008 | 15501.33 | 00:04 | 11.18 | 15490 | 15530 |
| CSP_n10_I70_E70_R00_009 | 15514.00 | 00:04 | 10.83 | 15490 | 15530 |
| CSP_n10_I70_E70_R00_010 | 15510.00 | 00:04 | 0.00 | 15510 | 15510 |
| CSP_n10_I70_E70_R02_001 | 16013.33 | 00:05 | 20.06 | 15970 | 16050 |
| CSP_n10_I70_E70_R02_002 | 15974.00 | 00:05 | 36.30 | 15910 | 16090 |
| CSP_n10_I70_E70_R02_003 | 15917.33 | 00:05 | 30.32 | 15830 | 15950 |
| CSP_n10_I70_E70_R02_004 | 15940.00 | 00:05 | 30.88 | 15870 | 15990 |
| CSP_n10_I70_E70_R02_005 | 16016.00 | 00:05 | 35.83 | 15970 | 16110 |
| CSP_n10_I70_E70_R02_006 | 15954.00 | 00:05 | 19.60 | 15930 | 15990 |
| CSP_n10_I70_E70_R02_007 | 15956.67 | 00:05 | 32.39 | 15870 | 16030 |
| CSP_n10_I70_E70_R02_008 | 15922.00 | 00:05 | 23.44 | 15870 | 15970 |
| CSP_n10_I70_E70_R02_009 | 15946.00 | 00:04 | 28.47 | 15850 | 15990 |
| CSP_n10_I70_E70_R02_010 | 15964.67 | 00:05 | 26.80 | 15870 | 16030 |

Fonte: Elaborada pelo autor (2023)

Tabela 18 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E70_R04_001 e CSP_n10_I70_E70_R08_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E70_R04_001 | 16167.33 | 00:05 | 21.75 | 16130 | 16230 |
| CSP_n10_I70_E70_R04_002 | 16143.33 | 00:05 | 22.71 | 16110 | 16190 |
| CSP_n10_I70_E70_R04_003 | 16150.00 | 00:05 | 26.33 | 16110 | 16210 |
| CSP_n10_I70_E70_R04_004 | 16157.33 | 00:05 | 22.20 | 16130 | 16210 |
| CSP_n10_I70_E70_R04_005 | 16326.00 | 00:05 | 20.91 | 16290 | 16390 |
| CSP_n10_I70_E70_R04_006 | 16319.33 | 00:05 | 27.68 | 16270 | 16390 |
| CSP_n10_I70_E70_R04_007 | 16302.00 | 00:05 | 22.86 | 16250 | 16350 |
| CSP_n10_I70_E70_R04_008 | 16165.33 | 00:05 | 17.65 | 16130 | 16210 |
| CSP_n10_I70_E70_R04_009 | 16285.33 | 00:05 | 27.17 | 16230 | 16330 |
| CSP_n10_I70_E70_R04_010 | 16286.00 | 00:05 | 24.98 | 16230 | 16330 |
| CSP_n10_I70_E70_R06_001 | 16365.33 | 00:05 | 43.11 | 16290 | 16450 |
| CSP_n10_I70_E70_R06_002 | 16532.00 | 00:05 | 35.16 | 16450 | 16650 |
| CSP_n10_I70_E70_R06_003 | 16558.67 | 00:05 | 42.80 | 16410 | 16630 |
| CSP_n10_I70_E70_R06_004 | 16515.33 | 00:05 | 35.75 | 16450 | 16590 |
| CSP_n10_I70_E70_R06_005 | 16504.00 | 00:06 | 39.04 | 16430 | 16590 |
| CSP_n10_I70_E70_R06_006 | 16634.67 | 00:05 | 29.07 | 16570 | 16710 |
| CSP_n10_I70_E70_R06_007 | 16642.67 | 00:05 | 36.33 | 16590 | 16730 |
| CSP_n10_I70_E70_R06_008 | 16516.00 | 00:05 | 22.60 | 16470 | 16550 |
| CSP_n10_I70_E70_R06_009 | 16411.33 | 00:05 | 30.96 | 16350 | 16470 |
| CSP_n10_I70_E70_R06_010 | 16536.67 | 00:05 | 32.39 | 16470 | 16630 |
| CSP_n10_I70_E70_R08_001 | 16884.00 | 00:06 | 52.45 | 16790 | 17010 |
| CSP_n10_I70_E70_R08_002 | 16906.67 | 00:05 | 31.02 | 16830 | 16990 |
| CSP_n10_I70_E70_R08_003 | 16790.67 | 00:06 | 44.87 | 16710 | 16890 |
| CSP_n10_I70_E70_R08_004 | 16917.33 | 00:06 | 37.41 | 16830 | 16970 |
| CSP_n10_I70_E70_R08_005 | 16854.67 | 00:06 | 38.19 | 16770 | 16930 |
| CSP_n10_I70_E70_R08_006 | 16908.00 | 00:06 | 33.21 | 16850 | 16970 |
| CSP_n10_I70_E70_R08_007 | 16776.00 | 00:06 | 49.57 | 16630 | 16830 |
| CSP_n10_I70_E70_R08_008 | 17046.00 | 00:05 | 36.66 | 16990 | 17170 |
| CSP_n10_I70_E70_R08_009 | 16773.33 | 00:05 | 31.45 | 16690 | 16810 |
| CSP_n10_I70_E70_R08_010 | 16924.00 | 00:05 | 58.23 | 16830 | 17050 |

Fonte: Elaborada pelo autor (2023)

Tabela 19 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E70_R10_001 e CSP_n10_I70_E70_R14_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E70_R10_001 | 17231.33 | 00:05 | 53.65 | 17130 | 17350 |
| CSP_n10_I70_E70_R10_002 | 17164.67 | 00:06 | 47.59 | 17070 | 17270 |
| CSP_n10_I70_E70_R10_003 | 17026.67 | 00:06 | 39.02 | 16950 | 17110 |
| CSP_n10_I70_E70_R10_004 | 16981.33 | 00:06 | 49.18 | 16870 | 17110 |
| CSP_n10_I70_E70_R10_005 | 16997.33 | 00:05 | 31.19 | 16930 | 17070 |
| CSP_n10_I70_E70_R10_006 | 17149.33 | 00:05 | 32.03 | 17070 | 17190 |
| CSP_n10_I70_E70_R10_007 | 17010.00 | 00:06 | 29.66 | 16950 | 17070 |
| CSP_n10_I70_E70_R10_008 | 17128.00 | 00:05 | 65.19 | 17030 | 17270 |
| CSP_n10_I70_E70_R10_009 | 17002.67 | 00:06 | 51.76 | 16890 | 17090 |
| CSP_n10_I70_E70_R10_010 | 17006.00 | 00:05 | 34.41 | 16930 | 17070 |
| CSP_n10_I70_E70_R12_001 | 17499.33 | 00:06 | 36.78 | 17410 | 17590 |
| CSP_n10_I70_E70_R12_002 | 17390.67 | 00:05 | 49.12 | 17270 | 17450 |
| CSP_n10_I70_E70_R12_003 | 17502.67 | 00:06 | 45.46 | 17410 | 17610 |
| CSP_n10_I70_E70_R12_004 | 17393.33 | 00:05 | 53.44 | 17290 | 17510 |
| CSP_n10_I70_E70_R12_005 | 17480.00 | 00:06 | 59.50 | 17390 | 17650 |
| CSP_n10_I70_E70_R12_006 | 17477.33 | 00:06 | 40.82 | 17390 | 17570 |
| CSP_n10_I70_E70_R12_007 | 17506.67 | 00:06 | 52.17 | 17390 | 17630 |
| CSP_n10_I70_E70_R12_008 | 17624.67 | 00:06 | 49.51 | 17530 | 17710 |
| CSP_n10_I70_E70_R12_009 | 17482.67 | 00:05 | 43.35 | 17390 | 17570 |
| CSP_n10_I70_E70_R12_010 | 17606.00 | 00:06 | 53.27 | 17490 | 17710 |
| CSP_n10_I70_E70_R14_001 | 17724.00 | 00:06 | 48.76 | 17610 | 17850 |
| CSP_n10_I70_E70_R14_002 | 17824.67 | 00:06 | 54.14 | 17710 | 17930 |
| CSP_n10_I70_E70_R14_003 | 17740.00 | 00:06 | 52.35 | 17630 | 17870 |
| CSP_n10_I70_E70_R14_004 | 17714.67 | 00:06 | 37.48 | 17630 | 17770 |
| CSP_n10_I70_E70_R14_005 | 17732.00 | 00:06 | 36.64 | 17650 | 17810 |
| CSP_n10_I70_E70_R14_006 | 17955.33 | 00:06 | 52.90 | 17830 | 18050 |
| CSP_n10_I70_E70_R14_007 | 17949.33 | 00:06 | 50.72 | 17850 | 18050 |
| CSP_n10_I70_E70_R14_008 | 17863.33 | 00:06 | 67.79 | 17750 | 17990 |
| CSP_n10_I70_E70_R14_009 | 17869.33 | 00:05 | 53.54 | 17770 | 17990 |
| CSP_n10_I70_E70_R14_010 | 17704.00 | 00:06 | 45.06 | 17610 | 17830 |

Fonte: Elaborada pelo autor (2023)

Tabela 20 – Resultados obtidos do SLO entre as instâncias CSP_n10_I70_E70_R16_001 e CSP_n10_I70_E70_R20_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I70_E70_R16_001 | 17800.67 | 00:06 | 40.57 | 17710 | 17890 |
| CSP_n10_I70_E70_R16_002 | 17959.33 | 00:06 | 44.94 | 17870 | 18030 |
| CSP_n10_I70_E70_R16_003 | 17952.00 | 00:07 | 47.71 | 17850 | 18010 |
| CSP_n10_I70_E70_R16_004 | 17948.00 | 00:06 | 34.00 | 17890 | 18010 |
| CSP_n10_I70_E70_R16_005 | 17973.33 | 00:06 | 34.29 | 17890 | 18030 |
| CSP_n10_I70_E70_R16_006 | 17834.67 | 00:05 | 28.60 | 17770 | 17870 |
| CSP_n10_I70_E70_R16_007 | 17957.33 | 00:06 | 71.08 | 17850 | 18130 |
| CSP_n10_I70_E70_R16_008 | 17818.67 | 00:06 | 51.82 | 17730 | 17910 |
| CSP_n10_I70_E70_R16_009 | 17974.00 | 00:06 | 43.02 | 17850 | 18070 |
| CSP_n10_I70_E70_R16_010 | 17958.00 | 00:06 | 48.33 | 17850 | 18070 |
| CSP_n10_I70_E70_R18_001 | 18211.33 | 00:06 | 51.88 | 18130 | 18350 |
| CSP_n10_I70_E70_R18_002 | 18295.33 | 00:06 | 50.58 | 18150 | 18370 |
| CSP_n10_I70_E70_R18_003 | 18128.00 | 00:06 | 46.58 | 18050 | 18230 |
| CSP_n10_I70_E70_R18_004 | 18175.33 | 00:06 | 43.80 | 18090 | 18270 |
| CSP_n10_I70_E70_R18_005 | 18169.33 | 00:06 | 32.86 | 18110 | 18250 |
| CSP_n10_I70_E70_R18_006 | 18172.67 | 00:06 | 42.81 | 18070 | 18230 |
| CSP_n10_I70_E70_R18_007 | 18094.67 | 00:06 | 53.09 | 17970 | 18170 |
| CSP_n10_I70_E70_R18_008 | 18338.00 | 00:06 | 56.94 | 18230 | 18430 |
| CSP_n10_I70_E70_R18_009 | 18150.00 | 00:06 | 48.17 | 18070 | 18270 |
| CSP_n10_I70_E70_R18_010 | 18188.67 | 00:06 | 42.56 | 18110 | 18270 |
| CSP_n10_I70_E70_R20_001 | 18552.67 | 00:06 | 48.37 | 18450 | 18630 |
| CSP_n10_I70_E70_R20_002 | 18417.33 | 00:06 | 43.35 | 18330 | 18490 |
| CSP_n10_I70_E70_R20_003 | 18382.67 | 00:07 | 50.99 | 18270 | 18470 |
| CSP_n10_I70_E70_R20_004 | 18555.33 | 00:06 | 54.88 | 18430 | 18650 |
| CSP_n10_I70_E70_R20_005 | 18412.00 | 00:06 | 41.10 | 18330 | 18490 |
| CSP_n10_I70_E70_R20_006 | 18413.33 | 00:07 | 49.55 | 18290 | 18530 |
| CSP_n10_I70_E70_R20_007 | 18399.33 | 00:06 | 55.55 | 18250 | 18530 |
| CSP_n10_I70_E70_R20_008 | 18564.00 | 00:06 | 36.93 | 18510 | 18650 |
| CSP_n10_I70_E70_R20_009 | 18376.67 | 00:06 | 38.76 | 18290 | 18450 |
| CSP_n10_I70_E70_R20_010 | 18425.33 | 00:05 | 41.85 | 18350 | 18510 |

Fonte: Elaborada pelo autor (2023)

Tabela 21 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R00_001 e CSP_n15_I40_E70_R04_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I40_E70_R00_001 | 28010.00 | 00:06 | 0.00 | 28010 | 28010 |
| CSP_n15_I40_E70_R00_002 | 27971.33 | 00:08 | 4.99 | 27970 | 27990 |
| CSP_n15_I40_E70_R00_003 | 27971.33 | 00:07 | 4.99 | 27970 | 27990 |
| CSP_n15_I40_E70_R00_004 | 27971.33 | 00:08 | 4.99 | 27970 | 27990 |
| CSP_n15_I40_E70_R00_005 | 28000.67 | 00:07 | 9.98 | 27990 | 28010 |
| CSP_n15_I40_E70_R00_006 | 27990.00 | 00:06 | 0.00 | 27990 | 27990 |
| CSP_n15_I40_E70_R00_007 | 28010.00 | 00:06 | 0.00 | 28010 | 28010 |
| CSP_n15_I40_E70_R00_008 | 27975.33 | 00:07 | 13.60 | 27970 | 28010 |
| CSP_n15_I40_E70_R00_009 | 27972.67 | 00:07 | 8.54 | 27970 | 28010 |
| CSP_n15_I40_E70_R00_010 | 27970.00 | 00:07 | 0.00 | 27970 | 27970 |
| CSP_n15_I40_E70_R02_001 | 28675.33 | 00:09 | 40.31 | 28630 | 28750 |
| CSP_n15_I40_E70_R02_002 | 28715.33 | 00:09 | 19.96 | 28670 | 28770 |
| CSP_n15_I40_E70_R02_003 | 28950.67 | 00:09 | 18.96 | 28910 | 28990 |
| CSP_n15_I40_E70_R02_004 | 28721.33 | 00:09 | 16.88 | 28690 | 28750 |
| CSP_n15_I40_E70_R02_005 | 28836.00 | 00:09 | 10.52 | 28830 | 28870 |
| CSP_n15_I40_E70_R02_006 | 28878.00 | 00:09 | 16.81 | 28850 | 28910 |
| CSP_n15_I40_E70_R02_007 | 28732.00 | 00:09 | 16.61 | 28710 | 28790 |
| CSP_n15_I40_E70_R02_008 | 28861.33 | 00:09 | 14.31 | 28850 | 28890 |
| CSP_n15_I40_E70_R02_009 | 28874.67 | 00:09 | 16.88 | 28850 | 28910 |
| CSP_n15_I40_E70_R02_010 | 28856.67 | 00:08 | 21.50 | 28830 | 28910 |
| CSP_n15_I40_E70_R04_001 | 29254.67 | 00:11 | 19.79 | 29210 | 29290 |
| CSP_n15_I40_E70_R04_002 | 29221.33 | 00:09 | 30.41 | 29150 | 29290 |
| CSP_n15_I40_E70_R04_003 | 29307.33 | 00:09 | 18.43 | 29270 | 29350 |
| CSP_n15_I40_E70_R04_004 | 29280.67 | 00:10 | 17.69 | 29250 | 29310 |
| CSP_n15_I40_E70_R04_005 | 29276.00 | 00:10 | 15.62 | 29250 | 29310 |
| CSP_n15_I40_E70_R04_006 | 29277.33 | 00:09 | 23.37 | 29230 | 29330 |
| CSP_n15_I40_E70_R04_007 | 29280.00 | 00:10 | 19.15 | 29230 | 29310 |
| CSP_n15_I40_E70_R04_008 | 29316.00 | 00:10 | 34.31 | 29270 | 29410 |
| CSP_n15_I40_E70_R04_009 | 29303.33 | 00:10 | 25.47 | 29270 | 29410 |
| CSP_n15_I40_E70_R04_010 | 29286.00 | 00:09 | 18.18 | 29250 | 29330 |

Fonte: Elaborada pelo autor (2023)

Tabela 22 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R06_001 e CSP_n15_I40_E70_R10_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I40_E70_R06_001 | 30184.00 | 00:09 | 28.82 | 30130 | 30250 |
| CSP_n15_I40_E70_R06_002 | 29864.00 | 00:10 | 38.70 | 29790 | 29950 |
| CSP_n15_I40_E70_R06_003 | 29861.33 | 00:10 | 20.45 | 29830 | 29910 |
| CSP_n15_I40_E70_R06_004 | 29968.67 | 00:11 | 30.08 | 29930 | 30030 |
| CSP_n15_I40_E70_R06_005 | 29913.33 | 00:10 | 33.50 | 29850 | 29990 |
| CSP_n15_I40_E70_R06_006 | 29981.33 | 00:10 | 31.70 | 29930 | 30070 |
| CSP_n15_I40_E70_R06_007 | 30095.33 | 00:10 | 21.87 | 30050 | 30130 |
| CSP_n15_I40_E70_R06_008 | 29958.67 | 00:11 | 28.13 | 29910 | 30010 |
| CSP_n15_I40_E70_R06_009 | 30014.00 | 00:10 | 35.55 | 29930 | 30070 |
| CSP_n15_I40_E70_R06_010 | 29908.67 | 00:11 | 44.70 | 29830 | 29990 |
| CSP_n15_I40_E70_R08_001 | 30476.00 | 00:11 | 37.65 | 30410 | 30610 |
| CSP_n15_I40_E70_R08_002 | 30586.67 | 00:10 | 22.56 | 30550 | 30630 |
| CSP_n15_I40_E70_R08_003 | 30405.33 | 00:12 | 37.48 | 30350 | 30490 |
| CSP_n15_I40_E70_R08_004 | 30700.67 | 00:10 | 26.70 | 30650 | 30750 |
| CSP_n15_I40_E70_R08_005 | 30415.33 | 00:10 | 42.87 | 30310 | 30510 |
| CSP_n15_I40_E70_R08_006 | 30398.67 | 00:11 | 35.66 | 30350 | 30510 |
| CSP_n15_I40_E70_R08_007 | 30420.67 | 00:11 | 52.34 | 30310 | 30550 |
| CSP_n15_I40_E70_R08_008 | 30422.67 | 00:11 | 48.30 | 30330 | 30510 |
| CSP_n15_I40_E70_R08_009 | 30426.67 | 00:10 | 38.67 | 30350 | 30530 |
| CSP_n15_I40_E70_R08_010 | 30415.33 | 00:11 | 46.46 | 30330 | 30530 |
| CSP_n15_I40_E70_R10_001 | 31209.33 | 00:11 | 28.98 | 31150 | 31270 |
| CSP_n15_I40_E70_R10_002 | 31109.33 | 00:12 | 40.16 | 31030 | 31190 |
| CSP_n15_I40_E70_R10_003 | 31046.00 | 00:11 | 55.23 | 30930 | 31150 |
| CSP_n15_I40_E70_R10_004 | 31054.00 | 00:10 | 50.44 | 30930 | 31130 |
| CSP_n15_I40_E70_R10_005 | 31134.00 | 00:11 | 50.96 | 31010 | 31250 |
| CSP_n15_I40_E70_R10_006 | 31076.67 | 00:10 | 40.77 | 30970 | 31170 |
| CSP_n15_I40_E70_R10_007 | 31078.00 | 00:11 | 38.16 | 31010 | 31170 |
| CSP_n15_I40_E70_R10_008 | 31025.33 | 00:11 | 49.98 | 30930 | 31130 |
| CSP_n15_I40_E70_R10_009 | 31112.00 | 00:11 | 20.23 | 31070 | 31150 |
| CSP_n15_I40_E70_R10_010 | 31152.67 | 00:12 | 52.34 | 31030 | 31250 |

Fonte: Elaborada pelo autor (2023)

Tabela 23 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R12_001 e CSP_n15_I40_E70_R16_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I40_E70_R12_001 | 31630.00 | 00:11 | 47.05 | 31510 | 31710 |
| CSP_n15_I40_E70_R12_002 | 31734.00 | 00:11 | 64.37 | 31570 | 31890 |
| CSP_n15_I40_E70_R12_003 | 31770.67 | 00:10 | 37.05 | 31690 | 31850 |
| CSP_n15_I40_E70_R12_004 | 31661.33 | 00:11 | 34.90 | 31570 | 31730 |
| CSP_n15_I40_E70_R12_005 | 31611.33 | 00:10 | 40.64 | 31530 | 31690 |
| CSP_n15_I40_E70_R12_006 | 31665.33 | 00:11 | 37.84 | 31590 | 31770 |
| CSP_n15_I40_E70_R12_007 | 31671.33 | 00:10 | 41.61 | 31570 | 31770 |
| CSP_n15_I40_E70_R12_008 | 32029.33 | 00:11 | 42.11 | 31930 | 32110 |
| CSP_n15_I40_E70_R12_009 | 31642.67 | 00:11 | 45.46 | 31570 | 31770 |
| CSP_n15_I40_E70_R12_010 | 31775.33 | 00:10 | 31.38 | 31710 | 31830 |
| CSP_n15_I40_E70_R14_001 | 32222.00 | 00:11 | 62.31 | 32130 | 32330 |
| CSP_n15_I40_E70_R14_002 | 32240.00 | 00:11 | 34.93 | 32170 | 32310 |
| CSP_n15_I40_E70_R14_003 | 32194.67 | 00:11 | 38.88 | 32090 | 32270 |
| CSP_n15_I40_E70_R14_004 | 32204.67 | 00:12 | 34.23 | 32130 | 32290 |
| CSP_n15_I40_E70_R14_005 | 32348.00 | 00:12 | 27.50 | 32290 | 32410 |
| CSP_n15_I40_E70_R14_006 | 32270.67 | 00:11 | 48.30 | 32170 | 32370 |
| CSP_n15_I40_E70_R14_007 | 32284.67 | 00:11 | 36.85 | 32230 | 32370 |
| CSP_n15_I40_E70_R14_008 | 32242.00 | 00:13 | 51.79 | 32170 | 32350 |
| CSP_n15_I40_E70_R14_009 | 32199.33 | 00:11 | 69.04 | 32010 | 32310 |
| CSP_n15_I40_E70_R14_010 | 32236.67 | 00:12 | 47.42 | 32130 | 32350 |
| CSP_n15_I40_E70_R16_001 | 32691.33 | 00:11 | 56.79 | 32570 | 32810 |
| CSP_n15_I40_E70_R16_002 | 32677.33 | 00:12 | 43.05 | 32590 | 32770 |
| CSP_n15_I40_E70_R16_003 | 32826.00 | 00:11 | 42.39 | 32770 | 32950 |
| CSP_n15_I40_E70_R16_004 | 32647.33 | 00:11 | 38.55 | 32550 | 32710 |
| CSP_n15_I40_E70_R16_005 | 32669.33 | 00:12 | 49.12 | 32570 | 32750 |
| CSP_n15_I40_E70_R16_006 | 32546.00 | 00:12 | 49.37 | 32450 | 32650 |
| CSP_n15_I40_E70_R16_007 | 32682.00 | 00:12 | 66.85 | 32530 | 32870 |
| CSP_n15_I40_E70_R16_008 | 32754.00 | 00:11 | 42.08 | 32670 | 32870 |
| CSP_n15_I40_E70_R16_009 | 32754.67 | 00:11 | 53.34 | 32670 | 32930 |
| CSP_n15_I40_E70_R16_010 | 32706.00 | 00:11 | 38.44 | 32630 | 32790 |

Fonte: Elaborada pelo autor (2023)

Tabela 24 – Resultados obtidos do SLO entre as instâncias CSP_n15_I40_E70_R18_001 e CSP_n15_I70_E40_R00_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I40_E70_R18_001 | 33288.67 | 00:12 | 56.32 | 33170 | 33370 |
| CSP_n15_I40_E70_R18_002 | 33243.33 | 00:11 | 43.92 | 33130 | 33310 |
| CSP_n15_I40_E70_R18_003 | 33209.33 | 00:12 | 44.27 | 33090 | 33290 |
| CSP_n15_I40_E70_R18_004 | 33389.33 | 00:11 | 45.16 | 33290 | 33490 |
| CSP_n15_I40_E70_R18_005 | 33311.33 | 00:12 | 59.31 | 33150 | 33410 |
| CSP_n15_I40_E70_R18_006 | 33315.33 | 00:12 | 43.49 | 33230 | 33390 |
| CSP_n15_I40_E70_R18_007 | 33308.00 | 00:11 | 53.75 | 33210 | 33430 |
| CSP_n15_I40_E70_R18_008 | 33330.67 | 00:13 | 53.04 | 33230 | 33410 |
| CSP_n15_I40_E70_R18_009 | 33292.67 | 00:12 | 47.81 | 33210 | 33390 |
| CSP_n15_I40_E70_R18_010 | 33312.67 | 00:11 | 43.43 | 33210 | 33370 |
| CSP_n15_I40_E70_R20_001 | 33731.33 | 00:11 | 75.00 | 33590 | 33870 |
| CSP_n15_I40_E70_R20_002 | 33655.33 | 00:11 | 55.84 | 33490 | 33730 |
| CSP_n15_I40_E70_R20_003 | 33727.33 | 00:12 | 53.85 | 33570 | 33830 |
| CSP_n15_I40_E70_R20_004 | 33763.33 | 00:12 | 56.41 | 33630 | 33870 |
| CSP_n15_I40_E70_R20_005 | 33759.33 | 00:12 | 69.04 | 33570 | 33870 |
| CSP_n15_I40_E70_R20_006 | 33604.67 | 00:12 | 57.49 | 33490 | 33710 |
| CSP_n15_I40_E70_R20_007 | 33594.67 | 00:13 | 59.93 | 33450 | 33690 |
| CSP_n15_I40_E70_R20_008 | 33632.00 | 00:12 | 41.75 | 33530 | 33730 |
| CSP_n15_I40_E70_R20_009 | 33607.33 | 00:12 | 44.94 | 33510 | 33690 |
| CSP_n15_I40_E70_R20_010 | 33690.67 | 00:12 | 43.05 | 33590 | 33770 |
| CSP_n15_I70_E40_R00_001 | 28010.00 | 00:06 | 0.00 | 28010 | 28010 |
| CSP_n15_I70_E40_R00_002 | 28010.00 | 00:07 | 0.00 | 28010 | 28010 |
| CSP_n15_I70_E40_R00_003 | 27991.33 | 00:07 | 4.99 | 27990 | 28010 |
| CSP_n15_I70_E40_R00_004 | 27990.00 | 00:07 | 0.00 | 27990 | 27990 |
| CSP_n15_I70_E40_R00_005 | 27990.67 | 00:08 | 3.59 | 27990 | 28010 |
| CSP_n15_I70_E40_R00_006 | 27969.33 | 00:08 | 24.49 | 27950 | 28010 |
| CSP_n15_I70_E40_R00_007 | 27992.67 | 00:08 | 6.80 | 27990 | 28010 |
| CSP_n15_I70_E40_R00_008 | 27992.67 | 00:08 | 6.80 | 27990 | 28010 |
| CSP_n15_I70_E40_R00_009 | 27990.00 | 00:07 | 0.00 | 27990 | 27990 |
| CSP_n15_I70_E40_R00_010 | 27972.67 | 00:07 | 9.98 | 27970 | 28010 |

Fonte: Elaborada pelo autor (2023)

Tabela 25 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R02_001 e CSP_n15_I70_E40_R06_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E40_R02_001 | 28894.67 | 00:10 | 58.81 | 28790 | 28970 |
| CSP_n15_I70_E40_R02_002 | 28805.33 | 00:10 | 21.09 | 28770 | 28850 |
| CSP_n15_I70_E40_R02_003 | 28813.33 | 00:10 | 31.02 | 28770 | 28910 |
| CSP_n15_I70_E40_R02_004 | 28741.33 | 00:09 | 40.56 | 28690 | 28810 |
| CSP_n15_I70_E40_R02_005 | 28736.00 | 00:10 | 39.72 | 28670 | 28810 |
| CSP_n15_I70_E40_R02_006 | 28711.33 | 00:09 | 24.18 | 28670 | 28770 |
| CSP_n15_I70_E40_R02_007 | 28836.67 | 00:09 | 35.90 | 28790 | 28910 |
| CSP_n15_I70_E40_R02_008 | 28830.00 | 00:10 | 36.51 | 28790 | 28950 |
| CSP_n15_I70_E40_R02_009 | 28723.33 | 00:10 | 20.87 | 28690 | 28770 |
| CSP_n15_I70_E40_R02_010 | 28908.67 | 00:10 | 78.64 | 28770 | 29030 |
| CSP_n15_I70_E40_R04_001 | 29288.00 | 00:11 | 30.27 | 29250 | 29370 |
| CSP_n15_I70_E40_R04_002 | 29281.33 | 00:10 | 44.63 | 29210 | 29410 |
| CSP_n15_I70_E40_R04_003 | 29450.67 | 00:10 | 58.53 | 29330 | 29590 |
| CSP_n15_I70_E40_R04_004 | 29339.33 | 00:10 | 66.68 | 29210 | 29510 |
| CSP_n15_I70_E40_R04_005 | 29336.00 | 00:09 | 64.53 | 29190 | 29490 |
| CSP_n15_I70_E40_R04_006 | 29348.67 | 00:10 | 72.10 | 29230 | 29510 |
| CSP_n15_I70_E40_R04_007 | 29348.00 | 00:10 | 44.83 | 29290 | 29450 |
| CSP_n15_I70_E40_R04_008 | 29320.00 | 00:10 | 60.39 | 29210 | 29470 |
| CSP_n15_I70_E40_R04_009 | 29374.00 | 00:10 | 70.50 | 29270 | 29590 |
| CSP_n15_I70_E40_R04_010 | 29301.33 | 00:10 | 59.03 | 29230 | 29450 |
| CSP_n15_I70_E40_R06_001 | 29888.67 | 00:11 | 53.15 | 29790 | 29990 |
| CSP_n15_I70_E40_R06_002 | 30008.67 | 00:10 | 74.82 | 29870 | 30150 |
| CSP_n15_I70_E40_R06_003 | 29916.00 | 00:10 | 75.39 | 29810 | 30110 |
| CSP_n15_I70_E40_R06_004 | 29999.33 | 00:10 | 68.07 | 29890 | 30150 |
| CSP_n15_I70_E40_R06_005 | 29960.67 | 00:10 | 48.09 | 29890 | 30050 |
| CSP_n15_I70_E40_R06_006 | 29920.00 | 00:11 | 32.96 | 29850 | 30010 |
| CSP_n15_I70_E40_R06_007 | 29996.00 | 00:11 | 73.96 | 29890 | 30150 |
| CSP_n15_I70_E40_R06_008 | 29848.67 | 00:11 | 51.10 | 29770 | 29970 |
| CSP_n15_I70_E40_R06_009 | 29995.33 | 00:10 | 52.14 | 29910 | 30110 |
| CSP_n15_I70_E40_R06_010 | 30014.00 | 00:10 | 23.89 | 29970 | 30050 |

Fonte: Elaborada pelo autor (2023)

Tabela 26 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R08_001 e CSP_n15_I70_E40_R12_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E40_R08_001 | 30510.00 | 00:10 | 50.86 | 30410 | 30650 |
| CSP_n15_I70_E40_R08_002 | 30538.67 | 00:10 | 64.23 | 30430 | 30630 |
| CSP_n15_I70_E40_R08_003 | 30483.33 | 00:11 | 96.52 | 30330 | 30670 |
| CSP_n15_I70_E40_R08_004 | 30504.67 | 00:10 | 56.08 | 30390 | 30610 |
| CSP_n15_I70_E40_R08_005 | 30505.33 | 00:12 | 76.54 | 30390 | 30690 |
| CSP_n15_I70_E40_R08_006 | 30411.33 | 00:10 | 47.87 | 30330 | 30510 |
| CSP_n15_I70_E40_R08_007 | 30414.67 | 00:10 | 47.24 | 30350 | 30530 |
| CSP_n15_I70_E40_R08_008 | 30420.00 | 00:11 | 73.89 | 30310 | 30570 |
| CSP_n15_I70_E40_R08_009 | 30507.33 | 00:11 | 68.07 | 30370 | 30690 |
| CSP_n15_I70_E40_R08_010 | 30387.33 | 00:10 | 64.86 | 30290 | 30590 |
| CSP_n15_I70_E40_R10_001 | 30972.67 | 00:11 | 63.82 | 30850 | 31070 |
| CSP_n15_I70_E40_R10_002 | 30998.67 | 00:12 | 83.70 | 30850 | 31190 |
| CSP_n15_I70_E40_R10_003 | 30988.67 | 00:12 | 86.09 | 30790 | 31110 |
| CSP_n15_I70_E40_R10_004 | 31093.33 | 00:11 | 112.91 | 30950 | 31330 |
| CSP_n15_I70_E40_R10_005 | 31049.33 | 00:10 | 51.76 | 30970 | 31190 |
| CSP_n15_I70_E40_R10_006 | 31059.33 | 00:12 | 59.04 | 30970 | 31210 |
| CSP_n15_I70_E40_R10_007 | 31076.00 | 00:10 | 60.26 | 30950 | 31210 |
| CSP_n15_I70_E40_R10_008 | 31096.00 | 00:11 | 58.00 | 30950 | 31230 |
| CSP_n15_I70_E40_R10_009 | 31095.33 | 00:11 | 71.73 | 30950 | 31270 |
| CSP_n15_I70_E40_R10_010 | 31160.00 | 00:11 | 63.61 | 31030 | 31310 |
| CSP_n15_I70_E40_R12_001 | 31756.00 | 00:11 | 120.13 | 31570 | 32050 |
| CSP_n15_I70_E40_R12_002 | 31685.33 | 00:10 | 66.27 | 31490 | 31810 |
| CSP_n15_I70_E40_R12_003 | 31616.00 | 00:12 | 83.77 | 31490 | 31770 |
| CSP_n15_I70_E40_R12_004 | 31604.00 | 00:12 | 69.12 | 31470 | 31730 |
| CSP_n15_I70_E40_R12_005 | 31854.00 | 00:11 | 107.88 | 31690 | 32110 |
| CSP_n15_I70_E40_R12_006 | 31587.33 | 00:12 | 89.40 | 31410 | 31790 |
| CSP_n15_I70_E40_R12_007 | 31606.67 | 00:13 | 68.52 | 31470 | 31750 |
| CSP_n15_I70_E40_R12_008 | 31735.33 | 00:12 | 68.69 | 31630 | 31910 |
| CSP_n15_I70_E40_R12_009 | 31579.33 | 00:11 | 61.48 | 31450 | 31690 |
| CSP_n15_I70_E40_R12_010 | 31568.00 | 00:11 | 89.94 | 31430 | 31770 |

Fonte: Elaborada pelo autor (2023)

Tabela 27 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R14_001 e CSP_n15_I70_E40_R18_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E40_R14_001 | 32154.00 | 00:12 | 66.81 | 32030 | 32270 |
| CSP_n15_I70_E40_R14_002 | 32232.67 | 00:12 | 60.82 | 32130 | 32390 |
| CSP_n15_I70_E40_R14_003 | 32136.00 | 00:12 | 79.19 | 32030 | 32290 |
| CSP_n15_I70_E40_R14_004 | 32161.33 | 00:11 | 82.73 | 32010 | 32310 |
| CSP_n15_I70_E40_R14_005 | 32082.67 | 00:10 | 63.14 | 31950 | 32190 |
| CSP_n15_I70_E40_R14_006 | 32098.00 | 00:12 | 79.77 | 31890 | 32230 |
| CSP_n15_I70_E40_R14_007 | 32036.00 | 00:11 | 117.55 | 31810 | 32270 |
| CSP_n15_I70_E40_R14_008 | 32038.67 | 00:11 | 56.02 | 31930 | 32150 |
| CSP_n15_I70_E40_R14_009 | 32034.67 | 00:11 | 73.15 | 31910 | 32230 |
| CSP_n15_I70_E40_R14_010 | 32026.67 | 00:12 | 75.38 | 31910 | 32170 |
| CSP_n15_I70_E40_R16_001 | 32552.00 | 00:12 | 65.19 | 32450 | 32690 |
| CSP_n15_I70_E40_R16_002 | 32531.33 | 00:13 | 81.64 | 32350 | 32730 |
| CSP_n15_I70_E40_R16_003 | 32530.00 | 00:12 | 81.81 | 32310 | 32670 |
| CSP_n15_I70_E40_R16_004 | 32499.33 | 00:12 | 62.34 | 32390 | 32650 |
| CSP_n15_I70_E40_R16_005 | 32578.00 | 00:11 | 60.35 | 32430 | 32690 |
| CSP_n15_I70_E40_R16_006 | 32496.67 | 00:13 | 64.57 | 32290 | 32630 |
| CSP_n15_I70_E40_R16_007 | 32541.33 | 00:12 | 73.34 | 32410 | 32730 |
| CSP_n15_I70_E40_R16_008 | 32476.67 | 00:11 | 86.00 | 32310 | 32650 |
| CSP_n15_I70_E40_R16_009 | 32534.67 | 00:12 | 77.92 | 32390 | 32710 |
| CSP_n15_I70_E40_R16_010 | 32710.67 | 00:12 | 111.41 | 32550 | 32970 |
| CSP_n15_I70_E40_R18_001 | 33076.00 | 00:13 | 75.57 | 32950 | 33250 |
| CSP_n15_I70_E40_R18_002 | 33218.00 | 00:14 | 87.27 | 32990 | 33390 |
| CSP_n15_I70_E40_R18_003 | 33107.33 | 00:12 | 97.94 | 32930 | 33310 |
| CSP_n15_I70_E40_R18_004 | 33086.00 | 00:11 | 94.15 | 32970 | 33330 |
| CSP_n15_I70_E40_R18_005 | 33122.00 | 00:14 | 99.55 | 32930 | 33370 |
| CSP_n15_I70_E40_R18_006 | 33108.00 | 00:11 | 71.25 | 32950 | 33230 |
| CSP_n15_I70_E40_R18_007 | 33232.00 | 00:12 | 78.20 | 33070 | 33350 |
| CSP_n15_I70_E40_R18_008 | 33237.33 | 00:12 | 59.21 | 33150 | 33390 |
| CSP_n15_I70_E40_R18_009 | 33075.33 | 00:13 | 74.82 | 32870 | 33210 |
| CSP_n15_I70_E40_R18_010 | 33076.67 | 00:13 | 88.89 | 32930 | 33350 |

Fonte: Elaborada pelo autor (2023)

Tabela 28 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E40_R20_001 e CSP_n15_I70_E70_R02_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E40_R20_001 | 33386.00 | 00:12 | 62.27 | 33270 | 33490 |
| CSP_n15_I70_E40_R20_002 | 33481.33 | 00:12 | 61.25 | 33350 | 33590 |
| CSP_n15_I70_E40_R20_003 | 33464.00 | 00:11 | 73.06 | 33310 | 33610 |
| CSP_n15_I70_E40_R20_004 | 33626.00 | 00:11 | 79.06 | 33450 | 33830 |
| CSP_n15_I70_E40_R20_005 | 33552.67 | 00:12 | 70.57 | 33370 | 33710 |
| CSP_n15_I70_E40_R20_006 | 33420.00 | 00:13 | 65.47 | 33310 | 33550 |
| CSP_n15_I70_E40_R20_007 | 33617.33 | 00:13 | 88.62 | 33490 | 33930 |
| CSP_n15_I70_E40_R20_008 | 33485.33 | 00:11 | 74.95 | 33310 | 33650 |
| CSP_n15_I70_E40_R20_009 | 33504.67 | 00:11 | 96.74 | 33350 | 33710 |
| CSP_n15_I70_E40_R20_010 | 33517.33 | 00:14 | 73.12 | 33350 | 33630 |
| CSP_n15_I70_E70_R00_001 | 34930.00 | 00:09 | 25.30 | 34870 | 34970 |
| CSP_n15_I70_E70_R00_002 | 34922.67 | 00:09 | 14.13 | 34890 | 34950 |
| CSP_n15_I70_E70_R00_003 | 34936.67 | 00:10 | 13.00 | 34910 | 34950 |
| CSP_n15_I70_E70_R00_004 | 34920.00 | 00:10 | 22.36 | 34890 | 34970 |
| CSP_n15_I70_E70_R00_005 | 34918.67 | 00:10 | 17.65 | 34890 | 34950 |
| CSP_n15_I70_E70_R00_006 | 34930.67 | 00:10 | 17.50 | 34890 | 34970 |
| CSP_n15_I70_E70_R00_007 | 34938.67 | 00:09 | 9.91 | 34930 | 34950 |
| CSP_n15_I70_E70_R00_008 | 34936.00 | 00:10 | 18.00 | 34910 | 34970 |
| CSP_n15_I70_E70_R00_009 | 34940.67 | 00:09 | 12.36 | 34910 | 34970 |
| CSP_n15_I70_E70_R00_010 | 34926.67 | 00:10 | 18.68 | 34890 | 34970 |
| CSP_n15_I70_E70_R02_001 | 35964.67 | 00:11 | 64.69 | 35870 | 36110 |
| CSP_n15_I70_E70_R02_002 | 35772.67 | 00:12 | 30.87 | 35710 | 35850 |
| CSP_n15_I70_E70_R02_003 | 35924.67 | 00:12 | 33.44 | 35870 | 36010 |
| CSP_n15_I70_E70_R02_004 | 35966.00 | 00:12 | 43.33 | 35890 | 36090 |
| CSP_n15_I70_E70_R02_005 | 35776.67 | 00:10 | 27.00 | 35710 | 35830 |
| CSP_n15_I70_E70_R02_006 | 35798.00 | 00:11 | 60.79 | 35650 | 35930 |
| CSP_n15_I70_E70_R02_007 | 35972.67 | 00:12 | 71.69 | 35870 | 36150 |
| CSP_n15_I70_E70_R02_008 | 35948.00 | 00:11 | 49.36 | 35850 | 36050 |
| CSP_n15_I70_E70_R02_009 | 35918.00 | 00:12 | 36.00 | 35870 | 36010 |
| CSP_n15_I70_E70_R02_010 | 36066.00 | 00:11 | 71.26 | 35930 | 36210 |

Fonte: Elaborada pelo autor (2023)

Tabela 29 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E70_R04_001 e CSP_n15_I70_E70_R08_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E70_R04_001 | 36514.67 | 00:11 | 46.96 | 36410 | 36590 |
| CSP_n15_I70_E70_R04_002 | 36393.33 | 00:12 | 50.62 | 36310 | 36530 |
| CSP_n15_I70_E70_R04_003 | 36546.67 | 00:12 | 77.82 | 36450 | 36830 |
| CSP_n15_I70_E70_R04_004 | 36499.33 | 00:11 | 94.62 | 36370 | 36690 |
| CSP_n15_I70_E70_R04_005 | 36518.00 | 00:12 | 49.42 | 36430 | 36610 |
| CSP_n15_I70_E70_R04_006 | 36361.33 | 00:12 | 51.30 | 36270 | 36450 |
| CSP_n15_I70_E70_R04_007 | 36388.00 | 00:12 | 48.26 | 36290 | 36510 |
| CSP_n15_I70_E70_R04_008 | 36413.33 | 00:12 | 82.64 | 36290 | 36590 |
| CSP_n15_I70_E70_R04_009 | 36398.00 | 00:14 | 56.47 | 36290 | 36510 |
| CSP_n15_I70_E70_R04_010 | 36672.00 | 00:13 | 94.14 | 36510 | 36830 |
| CSP_n15_I70_E70_R06_001 | 37216.67 | 00:13 | 76.83 | 37110 | 37410 |
| CSP_n15_I70_E70_R06_002 | 37268.67 | 00:11 | 107.32 | 37090 | 37450 |
| CSP_n15_I70_E70_R06_003 | 37195.33 | 00:13 | 68.30 | 37050 | 37310 |
| CSP_n15_I70_E70_R06_004 | 36998.00 | 00:13 | 85.42 | 36850 | 37150 |
| CSP_n15_I70_E70_R06_005 | 37250.00 | 00:13 | 75.72 | 37090 | 37410 |
| CSP_n15_I70_E70_R06_006 | 37180.67 | 00:13 | 79.96 | 37010 | 37330 |
| CSP_n15_I70_E70_R06_007 | 37292.67 | 00:15 | 65.88 | 37190 | 37430 |
| CSP_n15_I70_E70_R06_008 | 37185.33 | 00:13 | 65.46 | 37050 | 37290 |
| CSP_n15_I70_E70_R06_009 | 37016.00 | 00:12 | 45.94 | 36950 | 37130 |
| CSP_n15_I70_E70_R06_010 | 37234.67 | 00:12 | 116.47 | 37030 | 37490 |
| CSP_n15_I70_E70_R08_001 | 37566.00 | 00:13 | 64.17 | 37430 | 37690 |
| CSP_n15_I70_E70_R08_002 | 37691.33 | 00:12 | 60.21 | 37550 | 37810 |
| CSP_n15_I70_E70_R08_003 | 37652.67 | 00:12 | 63.19 | 37550 | 37830 |
| CSP_n15_I70_E70_R08_004 | 37840.67 | 00:13 | 107.79 | 37650 | 38110 |
| CSP_n15_I70_E70_R08_005 | 37795.33 | 00:13 | 91.06 | 37610 | 37950 |
| CSP_n15_I70_E70_R08_006 | 37800.00 | 00:13 | 74.61 | 37630 | 37970 |
| CSP_n15_I70_E70_R08_007 | 37624.67 | 00:14 | 78.64 | 37410 | 37790 |
| CSP_n15_I70_E70_R08_008 | 37708.67 | 00:12 | 70.60 | 37570 | 37850 |
| CSP_n15_I70_E70_R08_009 | 37798.67 | 00:12 | 88.65 | 37610 | 37990 |
| CSP_n15_I70_E70_R08_010 | 37568.00 | 00:12 | 90.68 | 37410 | 37730 |

Fonte: Elaborada pelo autor (2023)

Tabela 30 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E70_R10_001 e CSP_n15_I70_E70_R14_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E70_R10_001 | 38255.33 | 00:13 | 69.65 | 38150 | 38390 |
| CSP_n15_I70_E70_R10_002 | 38349.33 | 00:12 | 86.79 | 38210 | 38550 |
| CSP_n15_I70_E70_R10_003 | 38377.33 | 00:13 | 87.10 | 38230 | 38570 |
| CSP_n15_I70_E70_R10_004 | 38466.67 | 00:12 | 57.98 | 38330 | 38590 |
| CSP_n15_I70_E70_R10_005 | 38515.33 | 00:13 | 83.74 | 38310 | 38690 |
| CSP_n15_I70_E70_R10_006 | 38283.33 | 00:13 | 68.77 | 38150 | 38410 |
| CSP_n15_I70_E70_R10_007 | 38266.67 | 00:16 | 72.31 | 38110 | 38450 |
| CSP_n15_I70_E70_R10_008 | 38420.67 | 00:13 | 91.90 | 38250 | 38590 |
| CSP_n15_I70_E70_R10_009 | 38246.67 | 00:13 | 73.95 | 38150 | 38410 |
| CSP_n15_I70_E70_R10_010 | 38355.33 | 00:14 | 61.95 | 38230 | 38490 |
| CSP_n15_I70_E70_R12_001 | 39012.67 | 00:14 | 95.74 | 38810 | 39190 |
| CSP_n15_I70_E70_R12_002 | 38947.33 | 00:13 | 77.41 | 38750 | 39090 |
| CSP_n15_I70_E70_R12_003 | 39054.00 | 00:13 | 108.12 | 38870 | 39290 |
| CSP_n15_I70_E70_R12_004 | 38947.33 | 00:14 | 79.96 | 38810 | 39110 |
| CSP_n15_I70_E70_R12_005 | 38946.67 | 00:14 | 116.17 | 38770 | 39250 |
| CSP_n15_I70_E70_R12_006 | 38881.33 | 00:13 | 76.01 | 38730 | 39010 |
| CSP_n15_I70_E70_R12_007 | 38919.33 | 00:13 | 85.44 | 38770 | 39170 |
| CSP_n15_I70_E70_R12_008 | 38951.33 | 00:14 | 64.90 | 38790 | 39090 |
| CSP_n15_I70_E70_R12_009 | 39129.33 | 00:15 | 75.45 | 38950 | 39310 |
| CSP_n15_I70_E70_R12_010 | 38899.33 | 00:13 | 89.85 | 38730 | 39110 |
| CSP_n15_I70_E70_R14_001 | 39436.00 | 00:13 | 78.34 | 39310 | 39570 |
| CSP_n15_I70_E70_R14_002 | 39526.00 | 00:14 | 83.65 | 39330 | 39690 |
| CSP_n15_I70_E70_R14_003 | 39537.33 | 00:13 | 59.44 | 39430 | 39650 |
| CSP_n15_I70_E70_R14_004 | 39538.00 | 00:12 | 115.42 | 39290 | 39770 |
| CSP_n15_I70_E70_R14_005 | 39894.00 | 00:12 | 91.13 | 39710 | 40050 |
| CSP_n15_I70_E70_R14_006 | 39546.67 | 00:13 | 97.85 | 39290 | 39710 |
| CSP_n15_I70_E70_R14_007 | 39489.33 | 00:14 | 93.45 | 39230 | 39650 |
| CSP_n15_I70_E70_R14_008 | 39532.67 | 00:13 | 87.90 | 39390 | 39710 |
| CSP_n15_I70_E70_R14_009 | 39618.00 | 00:12 | 99.14 | 39270 | 39750 |
| CSP_n15_I70_E70_R14_010 | 39726.67 | 00:13 | 123.41 | 39430 | 39930 |

Fonte: Elaborada pelo autor (2023)

Tabela 31 – Resultados obtidos do SLO entre as instâncias CSP_n15_I70_E70_R16_001 e CSP_n15_I70_E70_R20_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I70_E70_R16_001 | 40106.00 | 00:14 | 133.41 | 39890 | 40330 |
| CSP_n15_I70_E70_R16_002 | 40059.33 | 00:14 | 97.53 | 39830 | 40250 |
| CSP_n15_I70_E70_R16_003 | 40016.00 | 00:13 | 78.00 | 39850 | 40190 |
| CSP_n15_I70_E70_R16_004 | 39988.00 | 00:13 | 62.90 | 39790 | 40090 |
| CSP_n15_I70_E70_R16_005 | 40122.00 | 00:13 | 95.58 | 39930 | 40350 |
| CSP_n15_I70_E70_R16_006 | 40076.67 | 00:14 | 78.03 | 39910 | 40210 |
| CSP_n15_I70_E70_R16_007 | 40048.00 | 00:15 | 108.36 | 39850 | 40310 |
| CSP_n15_I70_E70_R16_008 | 40020.67 | 00:13 | 93.63 | 39810 | 40230 |
| CSP_n15_I70_E70_R16_009 | 40157.33 | 00:15 | 111.05 | 39890 | 40410 |
| CSP_n15_I70_E70_R16_010 | 40042.00 | 00:14 | 59.01 | 39950 | 40230 |
| CSP_n15_I70_E70_R18_001 | 40662.00 | 00:13 | 88.33 | 40490 | 40810 |
| CSP_n15_I70_E70_R18_002 | 40646.67 | 00:12 | 71.57 | 40530 | 40910 |
| CSP_n15_I70_E70_R18_003 | 40676.67 | 00:13 | 67.20 | 40530 | 40850 |
| CSP_n15_I70_E70_R18_004 | 40662.67 | 00:15 | 78.06 | 40570 | 40910 |
| CSP_n15_I70_E70_R18_005 | 40688.67 | 00:13 | 130.22 | 40290 | 40970 |
| CSP_n15_I70_E70_R18_006 | 40698.00 | 00:15 | 120.40 | 40370 | 41010 |
| CSP_n15_I70_E70_R18_007 | 40675.33 | 00:13 | 66.12 | 40550 | 40830 |
| CSP_n15_I70_E70_R18_008 | 40619.33 | 00:14 | 88.80 | 40390 | 40810 |
| CSP_n15_I70_E70_R18_009 | 40746.67 | 00:13 | 107.47 | 40470 | 40930 |
| CSP_n15_I70_E70_R18_010 | 40776.00 | 00:13 | 83.45 | 40570 | 40890 |
| CSP_n15_I70_E70_R20_001 | 40980.67 | 00:15 | 97.81 | 40830 | 41170 |
| CSP_n15_I70_E70_R20_002 | 40923.33 | 00:15 | 95.27 | 40750 | 41110 |
| CSP_n15_I70_E70_R20_003 | 41010.67 | 00:14 | 115.76 | 40810 | 41190 |
| CSP_n15_I70_E70_R20_004 | 40956.00 | 00:14 | 84.40 | 40770 | 41090 |
| CSP_n15_I70_E70_R20_005 | 40919.33 | 00:12 | 79.62 | 40750 | 41050 |
| CSP_n15_I70_E70_R20_006 | 41037.33 | 00:16 | 75.45 | 40890 | 41190 |
| CSP_n15_I70_E70_R20_007 | 40936.00 | 00:15 | 73.96 | 40790 | 41070 |
| CSP_n15_I70_E70_R20_008 | 41121.33 | 00:14 | 97.25 | 40890 | 41290 |
| CSP_n15_I70_E70_R20_009 | 40984.00 | 00:14 | 116.18 | 40750 | 41190 |
| CSP_n15_I70_E70_R20_010 | 41171.33 | 00:13 | 72.83 | 40990 | 41290 |

Fonte: Elaborada pelo autor (2023)

Tabela 32 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R00_001 e CSP_n20_I40_E70_R04_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I40_E70_R00_001 | 49618.00 | 00:14 | 20.40 | 49610 | 49670 |
| CSP_n20_I40_E70_R00_002 | 49652.67 | 00:16 | 6.80 | 49650 | 49670 |
| CSP_n20_I40_E70_R00_003 | 49673.33 | 00:15 | 9.07 | 49670 | 49710 |
| CSP_n20_I40_E70_R00_004 | 49650.00 | 00:12 | 0.00 | 49650 | 49650 |
| CSP_n20_I40_E70_R00_005 | 49672.67 | 00:13 | 6.80 | 49670 | 49690 |
| CSP_n20_I40_E70_R00_006 | 49632.00 | 00:14 | 6.00 | 49630 | 49650 |
| CSP_n20_I40_E70_R00_007 | 49651.33 | 00:15 | 4.99 | 49650 | 49670 |
| CSP_n20_I40_E70_R00_008 | 49650.00 | 00:13 | 0.00 | 49650 | 49650 |
| CSP_n20_I40_E70_R00_009 | 49670.67 | 00:12 | 3.59 | 49670 | 49690 |
| CSP_n20_I40_E70_R00_010 | 49638.67 | 00:13 | 19.79 | 49630 | 49690 |
| CSP_n20_I40_E70_R02_001 | 50951.33 | 00:18 | 24.73 | 50910 | 51030 |
| CSP_n20_I40_E70_R02_002 | 50880.67 | 00:17 | 11.23 | 50870 | 50910 |
| CSP_n20_I40_E70_R02_003 | 50890.00 | 00:16 | 20.66 | 50850 | 50930 |
| CSP_n20_I40_E70_R02_004 | 50969.33 | 00:18 | 40.49 | 50910 | 51110 |
| CSP_n20_I40_E70_R02_005 | 50874.67 | 00:17 | 42.48 | 50830 | 50950 |
| CSP_n20_I40_E70_R02_006 | 50884.00 | 00:17 | 14.74 | 50850 | 50910 |
| CSP_n20_I40_E70_R02_007 | 50715.33 | 00:17 | 36.49 | 50650 | 50810 |
| CSP_n20_I40_E70_R02_008 | 51006.67 | 00:17 | 22.56 | 50970 | 51050 |
| CSP_n20_I40_E70_R02_009 | 50876.67 | 00:16 | 41.10 | 50750 | 50950 |
| CSP_n20_I40_E70_R02_010 | 50810.00 | 00:17 | 44.42 | 50710 | 50870 |
| CSP_n20_I40_E70_R04_001 | 51922.00 | 00:18 | 38.51 | 51850 | 51990 |
| CSP_n20_I40_E70_R04_002 | 51888.67 | 00:17 | 51.62 | 51790 | 52030 |
| CSP_n20_I40_E70_R04_003 | 51849.33 | 00:18 | 40.82 | 51770 | 51930 |
| CSP_n20_I40_E70_R04_004 | 51903.33 | 00:18 | 41.42 | 51830 | 52010 |
| CSP_n20_I40_E70_R04_005 | 51903.33 | 00:17 | 35.53 | 51850 | 52010 |
| CSP_n20_I40_E70_R04_006 | 51939.33 | 00:16 | 40.24 | 51870 | 52030 |
| CSP_n20_I40_E70_R04_007 | 51939.33 | 00:18 | 32.96 | 51870 | 52050 |
| CSP_n20_I40_E70_R04_008 | 51839.33 | 00:16 | 32.55 | 51770 | 51930 |
| CSP_n20_I40_E70_R04_009 | 51952.67 | 00:17 | 43.43 | 51850 | 52050 |
| CSP_n20_I40_E70_R04_010 | 51944.67 | 00:16 | 36.12 | 51890 | 52050 |

Fonte: Elaborada pelo autor (2023)

Tabela 33 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R06_001 e CSP_n20_I40_E70_R10_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I40_E70_R06_001 | 52785.33 | 00:17 | 53.34 | 52670 | 52890 |
| CSP_n20_I40_E70_R06_002 | 52811.33 | 00:19 | 95.07 | 52630 | 53010 |
| CSP_n20_I40_E70_R06_003 | 52969.33 | 00:20 | 63.98 | 52890 | 53130 |
| CSP_n20_I40_E70_R06_004 | 52858.67 | 00:18 | 70.37 | 52730 | 53010 |
| CSP_n20_I40_E70_R06_005 | 52865.33 | 00:18 | 73.52 | 52710 | 53030 |
| CSP_n20_I40_E70_R06_006 | 52858.00 | 00:19 | 78.25 | 52730 | 53050 |
| CSP_n20_I40_E70_R06_007 | 52772.00 | 00:20 | 62.47 | 52630 | 52890 |
| CSP_n20_I40_E70_R06_008 | 52897.33 | 00:16 | 38.46 | 52830 | 53010 |
| CSP_n20_I40_E70_R06_009 | 52844.67 | 00:18 | 58.64 | 52690 | 52970 |
| CSP_n20_I40_E70_R06_010 | 52884.67 | 00:20 | 99.72 | 52710 | 53030 |
| CSP_n20_I40_E70_R08_001 | 53839.33 | 00:20 | 71.50 | 53670 | 53990 |
| CSP_n20_I40_E70_R08_002 | 53820.67 | 00:19 | 76.37 | 53670 | 53950 |
| CSP_n20_I40_E70_R08_003 | 53853.33 | 00:18 | 76.61 | 53670 | 53990 |
| CSP_n20_I40_E70_R08_004 | 54090.67 | 00:20 | 43.05 | 54010 | 54170 |
| CSP_n20_I40_E70_R08_005 | 53750.67 | 00:20 | 57.85 | 53670 | 53950 |
| CSP_n20_I40_E70_R08_006 | 53929.33 | 00:18 | 45.16 | 53850 | 53990 |
| CSP_n20_I40_E70_R08_007 | 53804.00 | 00:18 | 51.16 | 53690 | 53890 |
| CSP_n20_I40_E70_R08_008 | 53873.33 | 00:19 | 83.60 | 53730 | 54110 |
| CSP_n20_I40_E70_R08_009 | 53880.00 | 00:20 | 81.61 | 53710 | 54070 |
| CSP_n20_I40_E70_R08_010 | 53887.33 | 00:17 | 84.65 | 53710 | 54090 |
| CSP_n20_I40_E70_R10_001 | 54859.33 | 00:17 | 72.98 | 54730 | 55050 |
| CSP_n20_I40_E70_R10_002 | 55017.33 | 00:19 | 59.66 | 54910 | 55150 |
| CSP_n20_I40_E70_R10_003 | 54974.67 | 00:19 | 34.90 | 54890 | 55030 |
| CSP_n20_I40_E70_R10_004 | 54914.67 | 00:20 | 78.77 | 54750 | 55090 |
| CSP_n20_I40_E70_R10_005 | 54971.33 | 00:21 | 50.84 | 54870 | 55070 |
| CSP_n20_I40_E70_R10_006 | 54885.33 | 00:21 | 68.83 | 54750 | 55010 |
| CSP_n20_I40_E70_R10_007 | 54974.67 | 00:19 | 57.43 | 54870 | 55110 |
| CSP_n20_I40_E70_R10_008 | 54867.33 | 00:23 | 70.75 | 54710 | 55010 |
| CSP_n20_I40_E70_R10_009 | 54916.67 | 00:20 | 130.47 | 54690 | 55210 |
| CSP_n20_I40_E70_R10_010 | 54908.67 | 00:20 | 83.89 | 54750 | 55070 |

Fonte: Elaborada pelo autor (2023)

Tabela 34 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R12_001 e CSP_n20_I40_E70_R16_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I40_E70_R12_001 | 55964.00 | 00:20 | 65.76 | 55850 | 56070 |
| CSP_n20_I40_E70_R12_002 | 56018.00 | 00:20 | 67.84 | 55890 | 56150 |
| CSP_n20_I40_E70_R12_003 | 55917.33 | 00:20 | 69.18 | 55790 | 56030 |
| CSP_n20_I40_E70_R12_004 | 56021.33 | 00:21 | 67.66 | 55890 | 56150 |
| CSP_n20_I40_E70_R12_005 | 56025.33 | 00:19 | 93.76 | 55750 | 56190 |
| CSP_n20_I40_E70_R12_006 | 56104.00 | 00:18 | 49.57 | 55990 | 56190 |
| CSP_n20_I40_E70_R12_007 | 56077.33 | 00:21 | 56.68 | 55970 | 56190 |
| CSP_n20_I40_E70_R12_008 | 55932.00 | 00:21 | 70.68 | 55790 | 56070 |
| CSP_n20_I40_E70_R12_009 | 56108.67 | 00:19 | 60.21 | 56010 | 56270 |
| CSP_n20_I40_E70_R12_010 | 56028.67 | 00:20 | 76.93 | 55890 | 56150 |
| CSP_n20_I40_E70_R14_001 | 57057.33 | 00:20 | 72.75 | 56830 | 57170 |
| CSP_n20_I40_E70_R14_002 | 56836.67 | 00:22 | 58.04 | 56670 | 56930 |
| CSP_n20_I40_E70_R14_003 | 56840.67 | 00:20 | 60.60 | 56730 | 57030 |
| CSP_n20_I40_E70_R14_004 | 56909.33 | 00:20 | 79.75 | 56710 | 57050 |
| CSP_n20_I40_E70_R14_005 | 57022.00 | 00:19 | 62.95 | 56910 | 57170 |
| CSP_n20_I40_E70_R14_006 | 56940.67 | 00:21 | 79.29 | 56770 | 57150 |
| CSP_n20_I40_E70_R14_007 | 56997.33 | 00:21 | 117.92 | 56750 | 57210 |
| CSP_n20_I40_E70_R14_008 | 56976.00 | 00:21 | 91.38 | 56810 | 57190 |
| CSP_n20_I40_E70_R14_009 | 56920.00 | 00:22 | 75.85 | 56790 | 57050 |
| CSP_n20_I40_E70_R14_010 | 56936.67 | 00:21 | 91.99 | 56770 | 57070 |
| CSP_n20_I40_E70_R16_001 | 57802.00 | 00:18 | 80.27 | 57630 | 57950 |
| CSP_n20_I40_E70_R16_002 | 57874.00 | 00:20 | 93.87 | 57650 | 58050 |
| CSP_n20_I40_E70_R16_003 | 57760.00 | 00:22 | 101.42 | 57570 | 57930 |
| CSP_n20_I40_E70_R16_004 | 57808.67 | 00:23 | 93.09 | 57570 | 57970 |
| CSP_n20_I40_E70_R16_005 | 57880.67 | 00:20 | 92.63 | 57670 | 58030 |
| CSP_n20_I40_E70_R16_006 | 57856.00 | 00:21 | 78.17 | 57670 | 58010 |
| CSP_n20_I40_E70_R16_007 | 57750.67 | 00:21 | 76.68 | 57550 | 57890 |
| CSP_n20_I40_E70_R16_008 | 57961.33 | 00:20 | 86.36 | 57730 | 58090 |
| CSP_n20_I40_E70_R16_009 | 57830.67 | 00:23 | 101.65 | 57570 | 57990 |
| CSP_n20_I40_E70_R16_010 | 57830.67 | 00:19 | 71.27 | 57650 | 57950 |

Fonte: Elaborada pelo autor (2023)

Tabela 35 – Resultados obtidos do SLO entre as instâncias CSP_n20_I40_E70_R18_001 e CSP_n20_I70_E40_R00_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I40_E70_R18_001 | 58740.00 | 00:22 | 73.89 | 58550 | 58870 |
| CSP_n20_I40_E70_R18_002 | 58764.00 | 00:20 | 82.49 | 58550 | 58910 |
| CSP_n20_I40_E70_R18_003 | 58880.00 | 00:23 | 92.77 | 58750 | 59050 |
| CSP_n20_I40_E70_R18_004 | 58984.00 | 00:23 | 63.49 | 58830 | 59110 |
| CSP_n20_I40_E70_R18_005 | 58750.00 | 00:23 | 71.74 | 58550 | 58870 |
| CSP_n20_I40_E70_R18_006 | 58833.33 | 00:20 | 74.67 | 58630 | 58950 |
| CSP_n20_I40_E70_R18_007 | 58794.00 | 00:23 | 64.79 | 58630 | 58910 |
| CSP_n20_I40_E70_R18_008 | 58656.00 | 00:22 | 71.58 | 58490 | 58770 |
| CSP_n20_I40_E70_R18_009 | 58748.00 | 00:20 | 72.91 | 58610 | 58890 |
| CSP_n20_I40_E70_R18_010 | 58781.33 | 00:22 | 83.85 | 58590 | 58910 |
| CSP_n20_I40_E70_R20_001 | 59884.67 | 00:22 | 101.45 | 59650 | 60050 |
| CSP_n20_I40_E70_R20_002 | 59644.67 | 00:22 | 81.64 | 59450 | 59770 |
| CSP_n20_I40_E70_R20_003 | 59696.00 | 00:20 | 60.03 | 59570 | 59830 |
| CSP_n20_I40_E70_R20_004 | 59783.33 | 00:21 | 76.83 | 59630 | 59910 |
| CSP_n20_I40_E70_R20_005 | 59688.00 | 00:21 | 55.70 | 59590 | 59870 |
| CSP_n20_I40_E70_R20_006 | 59703.33 | 00:20 | 101.50 | 59450 | 59850 |
| CSP_n20_I40_E70_R20_007 | 59720.67 | 00:22 | 65.88 | 59610 | 59910 |
| CSP_n20_I40_E70_R20_008 | 59607.33 | 00:22 | 73.71 | 59410 | 59810 |
| CSP_n20_I40_E70_R20_009 | 59733.33 | 00:22 | 103.16 | 59470 | 59910 |
| CSP_n20_I40_E70_R20_010 | 59796.67 | 00:18 | 91.26 | 59590 | 59930 |
| CSP_n20_I70_E40_R00_001 | 49655.33 | 00:14 | 13.60 | 49650 | 49690 |
| CSP_n20_I70_E40_R00_002 | 49673.33 | 00:14 | 7.45 | 49670 | 49690 |
| CSP_n20_I70_E40_R00_003 | 49670.67 | 00:13 | 3.59 | 49670 | 49690 |
| CSP_n20_I70_E40_R00_004 | 49670.67 | 00:13 | 3.59 | 49670 | 49690 |
| CSP_n20_I70_E40_R00_005 | 49670.00 | 00:12 | 0.00 | 49670 | 49670 |
| CSP_n20_I70_E40_R00_006 | 49652.67 | 00:15 | 6.80 | 49650 | 49670 |
| CSP_n20_I70_E40_R00_007 | 49622.67 | 00:15 | 19.65 | 49610 | 49670 |
| CSP_n20_I70_E40_R00_008 | 49690.00 | 00:11 | 0.00 | 49690 | 49690 |
| CSP_n20_I70_E40_R00_009 | 49670.00 | 00:12 | 0.00 | 49670 | 49670 |
| CSP_n20_I70_E40_R00_010 | 49652.67 | 00:14 | 6.80 | 49650 | 49670 |

Fonte: Elaborada pelo autor (2023)

Tabela 36 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R02_001 e CSP_n20_I70_E40_R06_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E40_R02_001 | 50952.67 | 00:18 | 73.53 | 50810 | 51070 |
| CSP_n20_I70_E40_R02_002 | 50982.00 | 00:17 | 96.97 | 50830 | 51130 |
| CSP_n20_I70_E40_R02_003 | 50877.33 | 00:19 | 63.35 | 50750 | 50990 |
| CSP_n20_I70_E40_R02_004 | 50902.67 | 00:17 | 63.56 | 50790 | 51030 |
| CSP_n20_I70_E40_R02_005 | 50870.67 | 00:16 | 49.39 | 50790 | 50950 |
| CSP_n20_I70_E40_R02_006 | 50851.33 | 00:18 | 70.41 | 50750 | 51030 |
| CSP_n20_I70_E40_R02_007 | 50883.33 | 00:20 | 46.00 | 50810 | 50950 |
| CSP_n20_I70_E40_R02_008 | 50774.67 | 00:19 | 37.48 | 50710 | 50850 |
| CSP_n20_I70_E40_R02_009 | 50904.67 | 00:18 | 45.00 | 50810 | 50990 |
| CSP_n20_I70_E40_R02_010 | 50811.33 | 00:18 | 46.74 | 50690 | 50910 |
| CSP_n20_I70_E40_R04_001 | 51922.00 | 00:20 | 55.76 | 51830 | 52050 |
| CSP_n20_I70_E40_R04_002 | 51965.33 | 00:16 | 76.71 | 51810 | 52110 |
| CSP_n20_I70_E40_R04_003 | 51906.67 | 00:17 | 59.12 | 51830 | 52030 |
| CSP_n20_I70_E40_R04_004 | 51954.00 | 00:20 | 94.99 | 51770 | 52150 |
| CSP_n20_I70_E40_R04_005 | 51857.33 | 00:20 | 59.89 | 51770 | 52010 |
| CSP_n20_I70_E40_R04_006 | 51900.67 | 00:19 | 87.75 | 51750 | 52090 |
| CSP_n20_I70_E40_R04_007 | 51927.33 | 00:18 | 62.77 | 51830 | 52050 |
| CSP_n20_I70_E40_R04_008 | 51907.33 | 00:17 | 97.26 | 51710 | 52090 |
| CSP_n20_I70_E40_R04_009 | 51898.00 | 00:20 | 65.44 | 51770 | 52070 |
| CSP_n20_I70_E40_R04_010 | 51889.33 | 00:18 | 57.85 | 51790 | 51990 |
| CSP_n20_I70_E40_R06_001 | 52809.33 | 00:19 | 93.88 | 52630 | 52990 |
| CSP_n20_I70_E40_R06_002 | 53212.67 | 00:15 | 87.29 | 53030 | 53350 |
| CSP_n20_I70_E40_R06_003 | 52789.33 | 00:20 | 68.21 | 52690 | 52950 |
| CSP_n20_I70_E40_R06_004 | 52893.33 | 00:19 | 67.54 | 52770 | 53110 |
| CSP_n20_I70_E40_R06_005 | 52781.33 | 00:20 | 96.29 | 52630 | 53030 |
| CSP_n20_I70_E40_R06_006 | 52826.00 | 00:19 | 82.04 | 52590 | 52970 |
| CSP_n20_I70_E40_R06_007 | 52979.33 | 00:20 | 143.08 | 52710 | 53290 |
| CSP_n20_I70_E40_R06_008 | 52738.00 | 00:21 | 95.86 | 52590 | 53010 |
| CSP_n20_I70_E40_R06_009 | 52862.00 | 00:21 | 80.60 | 52730 | 53010 |
| CSP_n20_I70_E40_R06_010 | 52914.67 | 00:19 | 81.11 | 52730 | 53090 |

Fonte: Elaborada pelo autor (2023)

Tabela 37 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R08_001 e CSP_n20_I70_E40_R12_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E40_R08_001 | 53816.67 | 00:20 | 143.14 | 53530 | 54050 |
| CSP_n20_I70_E40_R08_002 | 53840.00 | 00:21 | 124.66 | 53610 | 54070 |
| CSP_n20_I70_E40_R08_003 | 53929.33 | 00:20 | 86.95 | 53770 | 54130 |
| CSP_n20_I70_E40_R08_004 | 53926.00 | 00:21 | 96.66 | 53630 | 54230 |
| CSP_n20_I70_E40_R08_005 | 53887.33 | 00:19 | 149.64 | 53610 | 54210 |
| CSP_n20_I70_E40_R08_006 | 53827.33 | 00:20 | 139.88 | 53570 | 54090 |
| CSP_n20_I70_E40_R08_007 | 53828.67 | 00:19 | 101.84 | 53630 | 53990 |
| CSP_n20_I70_E40_R08_008 | 53674.00 | 00:20 | 72.92 | 53530 | 53810 |
| CSP_n20_I70_E40_R08_009 | 53751.33 | 00:20 | 108.68 | 53570 | 53990 |
| CSP_n20_I70_E40_R08_010 | 53748.00 | 00:18 | 93.86 | 53590 | 53990 |
| CSP_n20_I70_E40_R10_001 | 54629.33 | 00:21 | 83.50 | 54410 | 54790 |
| CSP_n20_I70_E40_R10_002 | 54862.00 | 00:20 | 154.88 | 54530 | 55210 |
| CSP_n20_I70_E40_R10_003 | 54732.00 | 00:18 | 102.03 | 54490 | 54890 |
| CSP_n20_I70_E40_R10_004 | 54774.00 | 00:20 | 123.76 | 54570 | 55210 |
| CSP_n20_I70_E40_R10_005 | 54802.00 | 00:21 | 128.85 | 54510 | 55050 |
| CSP_n20_I70_E40_R10_006 | 54664.67 | 00:19 | 117.64 | 54490 | 54930 |
| CSP_n20_I70_E40_R10_007 | 54716.67 | 00:19 | 123.86 | 54470 | 55030 |
| CSP_n20_I70_E40_R10_008 | 54733.33 | 00:20 | 99.48 | 54510 | 54930 |
| CSP_n20_I70_E40_R10_009 | 54695.33 | 00:21 | 94.22 | 54490 | 54850 |
| CSP_n20_I70_E40_R10_010 | 54835.33 | 00:21 | 100.12 | 54670 | 55030 |
| CSP_n20_I70_E40_R12_001 | 55673.33 | 00:22 | 104.06 | 55370 | 55830 |
| CSP_n20_I70_E40_R12_002 | 55695.33 | 00:21 | 76.76 | 55550 | 55850 |
| CSP_n20_I70_E40_R12_003 | 55812.67 | 00:22 | 102.86 | 55590 | 55990 |
| CSP_n20_I70_E40_R12_004 | 55877.33 | 00:20 | 112.96 | 55610 | 56050 |
| CSP_n20_I70_E40_R12_005 | 55798.67 | 00:19 | 95.17 | 55630 | 56010 |
| CSP_n20_I70_E40_R12_006 | 55783.33 | 00:20 | 139.46 | 55510 | 56090 |
| CSP_n20_I70_E40_R12_007 | 55743.33 | 00:20 | 100.84 | 55550 | 55910 |
| CSP_n20_I70_E40_R12_008 | 55720.67 | 00:20 | 109.15 | 55450 | 55950 |
| CSP_n20_I70_E40_R12_009 | 55705.33 | 00:20 | 80.11 | 55510 | 55890 |
| CSP_n20_I70_E40_R12_010 | 55647.33 | 00:21 | 93.91 | 55490 | 55910 |

Fonte: Elaborada pelo autor (2023)

Tabela 38 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R14_001 e CSP_n20_I70_E40_R18_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E40_R14_001 | 56585.33 | 00:21 | 105.41 | 56330 | 56830 |
| CSP_n20_I70_E40_R14_002 | 56615.33 | 00:18 | 94.37 | 56310 | 56790 |
| CSP_n20_I70_E40_R14_003 | 56566.00 | 00:24 | 101.90 | 56390 | 56870 |
| CSP_n20_I70_E40_R14_004 | 56658.00 | 00:19 | 95.44 | 56450 | 56870 |
| CSP_n20_I70_E40_R14_005 | 56649.33 | 00:20 | 90.11 | 56490 | 56870 |
| CSP_n20_I70_E40_R14_006 | 56738.67 | 00:20 | 110.60 | 56490 | 56990 |
| CSP_n20_I70_E40_R14_007 | 56649.33 | 00:25 | 128.94 | 56370 | 56910 |
| CSP_n20_I70_E40_R14_008 | 56724.67 | 00:19 | 93.09 | 56570 | 56950 |
| CSP_n20_I70_E40_R14_009 | 56564.67 | 00:20 | 97.83 | 56330 | 56770 |
| CSP_n20_I70_E40_R14_010 | 56848.00 | 00:21 | 125.89 | 56650 | 57110 |
| CSP_n20_I70_E40_R16_001 | 57380.00 | 00:21 | 122.17 | 57170 | 57630 |
| CSP_n20_I70_E40_R16_002 | 57619.33 | 00:22 | 105.80 | 57350 | 57830 |
| CSP_n20_I70_E40_R16_003 | 57352.67 | 00:22 | 104.66 | 57130 | 57570 |
| CSP_n20_I70_E40_R16_004 | 57384.00 | 00:21 | 129.84 | 57090 | 57630 |
| CSP_n20_I70_E40_R16_005 | 57416.00 | 00:22 | 88.42 | 57210 | 57630 |
| CSP_n20_I70_E40_R16_006 | 57359.33 | 00:23 | 112.16 | 57190 | 57590 |
| CSP_n20_I70_E40_R16_007 | 57486.00 | 00:24 | 102.29 | 57290 | 57750 |
| CSP_n20_I70_E40_R16_008 | 57427.33 | 00:21 | 180.94 | 57130 | 57970 |
| CSP_n20_I70_E40_R16_009 | 57475.33 | 00:23 | 94.51 | 57250 | 57650 |
| CSP_n20_I70_E40_R16_010 | 57554.67 | 00:22 | 71.68 | 57430 | 57810 |
| CSP_n20_I70_E40_R18_001 | 58356.00 | 00:22 | 78.68 | 58170 | 58510 |
| CSP_n20_I70_E40_R18_002 | 58212.00 | 00:24 | 89.94 | 58070 | 58410 |
| CSP_n20_I70_E40_R18_003 | 58175.33 | 00:22 | 126.17 | 57990 | 58370 |
| CSP_n20_I70_E40_R18_004 | 58392.00 | 00:20 | 106.75 | 58130 | 58570 |
| CSP_n20_I70_E40_R18_005 | 58356.67 | 00:21 | 100.31 | 58090 | 58550 |
| CSP_n20_I70_E40_R18_006 | 58550.00 | 00:20 | 136.72 | 58310 | 58890 |
| CSP_n20_I70_E40_R18_007 | 58375.33 | 00:24 | 98.78 | 58190 | 58650 |
| CSP_n20_I70_E40_R18_008 | 58415.33 | 00:21 | 117.64 | 58230 | 58650 |
| CSP_n20_I70_E40_R18_009 | 58147.33 | 00:22 | 119.86 | 57850 | 58510 |
| CSP_n20_I70_E40_R18_010 | 58272.00 | 00:22 | 121.91 | 58010 | 58590 |

Fonte: Elaborada pelo autor (2023)

Tabela 39 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E40_R20_001 e CSP_n20_I70_E70_R02_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E40_R20_001 | 59350.00 | 00:23 | 103.41 | 59150 | 59590 |
| CSP_n20_I70_E40_R20_002 | 59447.33 | 00:23 | 88.35 | 59250 | 59610 |
| CSP_n20_I70_E40_R20_003 | 59214.67 | 00:22 | 76.36 | 59110 | 59410 |
| CSP_n20_I70_E40_R20_004 | 59204.00 | 00:21 | 80.52 | 59070 | 59470 |
| CSP_n20_I70_E40_R20_005 | 59197.33 | 00:21 | 145.19 | 58870 | 59710 |
| CSP_n20_I70_E40_R20_006 | 59331.33 | 00:24 | 81.64 | 59190 | 59490 |
| CSP_n20_I70_E40_R20_007 | 59249.33 | 00:20 | 112.84 | 59090 | 59570 |
| CSP_n20_I70_E40_R20_008 | 59444.00 | 00:23 | 110.17 | 59230 | 59690 |
| CSP_n20_I70_E40_R20_009 | 59262.67 | 00:21 | 113.31 | 58990 | 59510 |
| CSP_n20_I70_E40_R20_010 | 59400.00 | 00:23 | 106.68 | 59110 | 59590 |
| CSP_n20_I70_E70_R00_001 | 61849.33 | 00:17 | 18.25 | 61810 | 61890 |
| CSP_n20_I70_E70_R00_002 | 61836.00 | 00:18 | 14.74 | 61810 | 61870 |
| CSP_n20_I70_E70_R00_003 | 61841.33 | 00:18 | 22.91 | 61770 | 61870 |
| CSP_n20_I70_E70_R00_004 | 61849.33 | 00:19 | 18.25 | 61810 | 61890 |
| CSP_n20_I70_E70_R00_005 | 61843.33 | 00:18 | 27.00 | 61790 | 61870 |
| CSP_n20_I70_E70_R00_006 | 61838.00 | 00:16 | 19.73 | 61810 | 61870 |
| CSP_n20_I70_E70_R00_007 | 61850.00 | 00:17 | 20.66 | 61810 | 61890 |
| CSP_n20_I70_E70_R00_008 | 61834.67 | 00:20 | 16.88 | 61810 | 61870 |
| CSP_n20_I70_E70_R00_009 | 61837.33 | 00:17 | 16.72 | 61810 | 61870 |
| CSP_n20_I70_E70_R00_010 | 61841.33 | 00:19 | 21.09 | 61810 | 61890 |
| CSP_n20_I70_E70_R02_001 | 63099.33 | 00:19 | 64.03 | 63010 | 63230 |
| CSP_n20_I70_E70_R02_002 | 63292.00 | 00:20 | 83.32 | 63150 | 63470 |
| CSP_n20_I70_E70_R02_003 | 63369.33 | 00:22 | 105.76 | 63170 | 63610 |
| CSP_n20_I70_E70_R02_004 | 63476.67 | 00:21 | 90.09 | 63310 | 63730 |
| CSP_n20_I70_E70_R02_005 | 63402.00 | 00:23 | 77.39 | 63230 | 63530 |
| CSP_n20_I70_E70_R02_006 | 63326.00 | 00:21 | 75.79 | 63190 | 63510 |
| CSP_n20_I70_E70_R02_007 | 63359.33 | 00:20 | 95.46 | 63230 | 63590 |
| CSP_n20_I70_E70_R02_008 | 63325.33 | 00:20 | 82.09 | 63170 | 63510 |
| CSP_n20_I70_E70_R02_009 | 63334.00 | 00:22 | 99.12 | 63170 | 63630 |
| CSP_n20_I70_E70_R02_010 | 63444.00 | 00:24 | 123.30 | 63230 | 63730 |

Fonte: Elaborada pelo autor (2023)

Tabela 40 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E70_R04_001 e CSP_n20_I70_E70_R08_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E70_R04_001 | 64208.00 | 00:22 | 88.29 | 64050 | 64410 |
| CSP_n20_I70_E70_R04_002 | 64322.00 | 00:21 | 119.73 | 64170 | 64610 |
| CSP_n20_I70_E70_R04_003 | 64580.00 | 00:19 | 90.74 | 64430 | 64850 |
| CSP_n20_I70_E70_R04_004 | 64493.33 | 00:23 | 103.16 | 64330 | 64730 |
| CSP_n20_I70_E70_R04_005 | 64774.00 | 00:23 | 156.24 | 64450 | 65070 |
| CSP_n20_I70_E70_R04_006 | 64402.00 | 00:20 | 77.22 | 64250 | 64570 |
| CSP_n20_I70_E70_R04_007 | 64497.33 | 00:24 | 123.56 | 64270 | 64830 |
| CSP_n20_I70_E70_R04_008 | 64492.67 | 00:23 | 111.20 | 64230 | 64770 |
| CSP_n20_I70_E70_R04_009 | 64224.67 | 00:21 | 139.42 | 63970 | 64710 |
| CSP_n20_I70_E70_R04_010 | 64536.00 | 00:21 | 93.26 | 64390 | 64730 |
| CSP_n20_I70_E70_R06_001 | 65398.67 | 00:21 | 113.92 | 65190 | 65670 |
| CSP_n20_I70_E70_R06_002 | 65719.33 | 00:21 | 117.16 | 65470 | 65910 |
| CSP_n20_I70_E70_R06_003 | 65489.33 | 00:19 | 101.78 | 65350 | 65770 |
| CSP_n20_I70_E70_R06_004 | 65479.33 | 00:23 | 125.08 | 65270 | 65750 |
| CSP_n20_I70_E70_R06_005 | 65439.33 | 00:24 | 97.67 | 65290 | 65630 |
| CSP_n20_I70_E70_R06_006 | 65495.33 | 00:23 | 96.19 | 65310 | 65710 |
| CSP_n20_I70_E70_R06_007 | 65434.67 | 00:19 | 90.14 | 65270 | 65610 |
| CSP_n20_I70_E70_R06_008 | 65415.33 | 00:22 | 80.32 | 65270 | 65650 |
| CSP_n20_I70_E70_R06_009 | 65493.33 | 00:19 | 114.09 | 65290 | 65710 |
| CSP_n20_I70_E70_R06_010 | 65469.33 | 00:22 | 92.73 | 65310 | 65710 |
| CSP_n20_I70_E70_R08_001 | 66486.67 | 00:25 | 115.94 | 66230 | 66690 |
| CSP_n20_I70_E70_R08_002 | 66579.33 | 00:21 | 147.13 | 66330 | 66890 |
| CSP_n20_I70_E70_R08_003 | 66641.33 | 00:21 | 140.07 | 66390 | 66950 |
| CSP_n20_I70_E70_R08_004 | 66598.00 | 00:25 | 119.73 | 66350 | 66830 |
| CSP_n20_I70_E70_R08_005 | 66495.33 | 00:26 | 75.88 | 66350 | 66630 |
| CSP_n20_I70_E70_R08_006 | 66869.33 | 00:20 | 120.50 | 66630 | 67170 |
| CSP_n20_I70_E70_R08_007 | 66584.67 | 00:21 | 119.66 | 66310 | 66790 |
| CSP_n20_I70_E70_R08_008 | 66647.33 | 00:22 | 99.30 | 66470 | 66830 |
| CSP_n20_I70_E70_R08_009 | 66533.33 | 00:19 | 103.81 | 66310 | 66730 |
| CSP_n20_I70_E70_R08_010 | 66590.67 | 00:22 | 112.96 | 66390 | 66830 |

Fonte: Elaborada pelo autor (2023)

Tabela 41 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E70_R10_001 e CSP_n20_I70_E70_R14_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E70_R10_001 | 67553.33 | 00:25 | 144.32 | 67350 | 67990 |
| CSP_n20_I70_E70_R10_002 | 67528.67 | 00:22 | 101.84 | 67310 | 67710 |
| CSP_n20_I70_E70_R10_003 | 67739.33 | 00:22 | 174.41 | 67250 | 68030 |
| CSP_n20_I70_E70_R10_004 | 67560.67 | 00:25 | 112.51 | 67310 | 67830 |
| CSP_n20_I70_E70_R10_005 | 67562.00 | 00:24 | 106.53 | 67350 | 67850 |
| CSP_n20_I70_E70_R10_006 | 67722.00 | 00:23 | 111.19 | 67410 | 67970 |
| CSP_n20_I70_E70_R10_007 | 67682.00 | 00:19 | 178.63 | 67230 | 67990 |
| CSP_n20_I70_E70_R10_008 | 67650.67 | 00:22 | 107.39 | 67450 | 67790 |
| CSP_n20_I70_E70_R10_009 | 67564.00 | 00:21 | 89.47 | 67410 | 67810 |
| CSP_n20_I70_E70_R10_010 | 67579.33 | 00:23 | 101.29 | 67330 | 67770 |
| CSP_n20_I70_E70_R12_001 | 68786.67 | 00:25 | 131.66 | 68530 | 69050 |
| CSP_n20_I70_E70_R12_002 | 68999.33 | 00:24 | 155.24 | 68650 | 69310 |
| CSP_n20_I70_E70_R12_003 | 68786.67 | 00:24 | 167.74 | 68430 | 69130 |
| CSP_n20_I70_E70_R12_004 | 68742.67 | 00:25 | 110.09 | 68450 | 68970 |
| CSP_n20_I70_E70_R12_005 | 68933.33 | 00:24 | 194.33 | 68510 | 69330 |
| CSP_n20_I70_E70_R12_006 | 68934.67 | 00:25 | 113.34 | 68690 | 69170 |
| CSP_n20_I70_E70_R12_007 | 68733.33 | 00:22 | 130.55 | 68350 | 68950 |
| CSP_n20_I70_E70_R12_008 | 68781.33 | 00:24 | 110.35 | 68470 | 69010 |
| CSP_n20_I70_E70_R12_009 | 68706.00 | 00:24 | 129.76 | 68410 | 68990 |
| CSP_n20_I70_E70_R12_010 | 68692.00 | 00:23 | 115.86 | 68330 | 68990 |
| CSP_n20_I70_E70_R14_001 | 69884.67 | 00:22 | 133.36 | 69570 | 70090 |
| CSP_n20_I70_E70_R14_002 | 69764.67 | 00:24 | 108.07 | 69530 | 69910 |
| CSP_n20_I70_E70_R14_003 | 69673.33 | 00:25 | 92.24 | 69450 | 69810 |
| CSP_n20_I70_E70_R14_004 | 69810.00 | 00:24 | 113.37 | 69430 | 70030 |
| CSP_n20_I70_E70_R14_005 | 69700.00 | 00:24 | 132.94 | 69430 | 69950 |
| CSP_n20_I70_E70_R14_006 | 69665.33 | 00:24 | 155.92 | 69270 | 69910 |
| CSP_n20_I70_E70_R14_007 | 69823.33 | 00:23 | 125.26 | 69590 | 70070 |
| CSP_n20_I70_E70_R14_008 | 69811.33 | 00:23 | 121.87 | 69470 | 70070 |
| CSP_n20_I70_E70_R14_009 | 69633.33 | 00:24 | 110.16 | 69410 | 69830 |
| CSP_n20_I70_E70_R14_010 | 69708.00 | 00:25 | 110.68 | 69450 | 69930 |

Fonte: Elaborada pelo autor (2023)

Tabela 42 – Resultados obtidos do SLO entre as instâncias CSP_n20_I70_E70_R16_001 e CSP_n20_I70_E70_R20_010.

| Instância | Valor | Tempo (mm:ss) | SLO | | |
|-------------------------|----------|------------------|------------------|--------|--------|
| | | | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I70_E70_R16_001 | 70631.33 | 00:24 | 153.36 | 70370 | 71130 |
| CSP_n20_I70_E70_R16_002 | 70747.33 | 00:26 | 152.90 | 70370 | 71070 |
| CSP_n20_I70_E70_R16_003 | 70866.00 | 00:24 | 147.64 | 70530 | 71090 |
| CSP_n20_I70_E70_R16_004 | 70670.67 | 00:22 | 143.71 | 70370 | 70950 |
| CSP_n20_I70_E70_R16_005 | 70614.00 | 00:25 | 94.15 | 70470 | 70810 |
| CSP_n20_I70_E70_R16_006 | 70554.00 | 00:25 | 131.09 | 70210 | 70790 |
| CSP_n20_I70_E70_R16_007 | 70672.00 | 00:23 | 104.86 | 70410 | 70850 |
| CSP_n20_I70_E70_R16_008 | 70670.67 | 00:26 | 118.71 | 70430 | 71010 |
| CSP_n20_I70_E70_R16_009 | 70679.33 | 00:25 | 143.55 | 70350 | 70930 |
| CSP_n20_I70_E70_R16_010 | 70525.33 | 00:25 | 118.85 | 70290 | 70710 |
| CSP_n20_I70_E70_R18_001 | 71746.67 | 00:22 | 119.79 | 71530 | 71910 |
| CSP_n20_I70_E70_R18_002 | 71636.00 | 00:26 | 76.79 | 71470 | 71730 |
| CSP_n20_I70_E70_R18_003 | 71618.00 | 00:23 | 119.73 | 71390 | 71850 |
| CSP_n20_I70_E70_R18_004 | 71720.00 | 00:25 | 113.81 | 71470 | 71950 |
| CSP_n20_I70_E70_R18_005 | 71732.00 | 00:25 | 136.66 | 71490 | 72070 |
| CSP_n20_I70_E70_R18_006 | 71549.33 | 00:26 | 123.34 | 71270 | 71810 |
| CSP_n20_I70_E70_R18_007 | 71552.00 | 00:25 | 168.23 | 71150 | 71790 |
| CSP_n20_I70_E70_R18_008 | 71677.33 | 00:25 | 149.26 | 71350 | 71930 |
| CSP_n20_I70_E70_R18_009 | 71600.00 | 00:27 | 114.98 | 71410 | 71790 |
| CSP_n20_I70_E70_R18_010 | 71622.00 | 00:25 | 100.61 | 71410 | 71810 |
| CSP_n20_I70_E70_R20_001 | 72740.00 | 00:25 | 146.95 | 72470 | 72970 |
| CSP_n20_I70_E70_R20_002 | 72621.33 | 00:27 | 180.79 | 72290 | 72970 |
| CSP_n20_I70_E70_R20_003 | 72447.33 | 00:22 | 129.48 | 72210 | 72690 |
| CSP_n20_I70_E70_R20_004 | 72739.33 | 00:25 | 128.66 | 72510 | 72970 |
| CSP_n20_I70_E70_R20_005 | 72881.33 | 00:26 | 124.97 | 72650 | 73110 |
| CSP_n20_I70_E70_R20_006 | 72372.67 | 00:25 | 112.40 | 72130 | 72690 |
| CSP_n20_I70_E70_R20_007 | 72546.67 | 00:27 | 148.69 | 72170 | 72890 |
| CSP_n20_I70_E70_R20_008 | 72679.33 | 00:25 | 169.21 | 72210 | 72910 |
| CSP_n20_I70_E70_R20_009 | 72646.00 | 00:24 | 129.04 | 72390 | 72850 |
| CSP_n20_I70_E70_R20_010 | 72490.67 | 00:25 | 109.12 | 72250 | 72690 |

Fonte: Elaborada pelo autor (2023)

Tabela 43 – Resultados obtidos do SLO+BRKGA entre as instâncias CSP_n10_I40_E70_R06_003 e CSP_n15_I40_E70_R06_009.

| Instância | Valor | SLO+BRKGA | | | |
|-------------------------|----------|---------------|---------------|--------|--------|
| | | Tempo (mm:ss) | Desvio Padrão | Mínimo | Máximo |
| CSP_n10_I40_E70_R06_003 | 13292.67 | 00:27 | 6.80 | 13290 | 13310 |
| CSP_n10_I40_E70_R06_007 | 13314.00 | 00:27 | 8.00 | 13310 | 13330 |
| CSP_n10_I40_E70_R06_008 | 13450.00 | 00:27 | 0.00 | 13450 | 13450 |
| CSP_n10_I40_E70_R10_003 | 13750.67 | 00:28 | 8.14 | 13730 | 13770 |
| CSP_n10_I40_E70_R10_005 | 13888.00 | 00:28 | 6.00 | 13870 | 13890 |
| CSP_n10_I40_E70_R10_009 | 13871.33 | 00:28 | 4.99 | 13870 | 13890 |
| CSP_n10_I40_E70_R14_002 | 14408.67 | 00:29 | 10.24 | 14390 | 14430 |
| CSP_n10_I40_E70_R14_006 | 14407.33 | 00:29 | 6.80 | 14390 | 14410 |
| CSP_n10_I40_E70_R14_008 | 14533.33 | 00:29 | 7.45 | 14530 | 14550 |
| CSP_n10_I70_E40_R00_002 | 12390.00 | 00:25 | 0.00 | 12390 | 12390 |
| CSP_n10_I70_E40_R00_005 | 12390.00 | 00:25 | 0.00 | 12390 | 12390 |
| CSP_n10_I70_E40_R00_008 | 12390.00 | 00:25 | 0.00 | 12390 | 12390 |
| CSP_n10_I70_E40_R10_001 | 13694.67 | 00:28 | 11.18 | 13670 | 13710 |
| CSP_n10_I70_E40_R10_004 | 13694.67 | 00:28 | 8.46 | 13690 | 13710 |
| CSP_n10_I70_E40_R10_010 | 13787.33 | 00:28 | 11.23 | 13770 | 13810 |
| CSP_n10_I70_E40_R20_001 | 14930.67 | 00:31 | 14.13 | 14910 | 14950 |
| CSP_n10_I70_E40_R20_006 | 14958.00 | 00:31 | 27.62 | 14910 | 14990 |
| CSP_n10_I70_E40_R20_010 | 14972.00 | 00:31 | 14.92 | 14950 | 14990 |
| CSP_n10_I70_E70_R02_003 | 15830.00 | 00:30 | 0.00 | 15830 | 15830 |
| CSP_n10_I70_E70_R02_006 | 15877.33 | 00:29 | 9.64 | 15870 | 15890 |
| CSP_n10_I70_E70_R02_009 | 15850.67 | 00:30 | 3.59 | 15850 | 15870 |
| CSP_n10_I70_E70_R10_002 | 17040.00 | 00:32 | 16.93 | 17010 | 17070 |
| CSP_n10_I70_E70_R10_007 | 16914.00 | 00:32 | 14.97 | 16890 | 16950 |
| CSP_n10_I70_E70_R10_010 | 16868.00 | 00:32 | 18.15 | 16830 | 16910 |
| CSP_n10_I70_E70_R20_002 | 18302.00 | 00:35 | 13.27 | 18290 | 18330 |
| CSP_n10_I70_E70_R20_006 | 18296.00 | 00:35 | 14.74 | 18270 | 18330 |
| CSP_n10_I70_E70_R20_009 | 18266.67 | 00:35 | 13.74 | 18250 | 18290 |
| CSP_n15_I40_E70_R06_002 | 29772.67 | 00:59 | 14.36 | 29750 | 29790 |
| CSP_n15_I40_E70_R06_007 | 30032.00 | 00:59 | 14.92 | 30010 | 30050 |
| CSP_n15_I40_E70_R06_009 | 29913.33 | 00:59 | 20.06 | 29870 | 29950 |

Fonte: Elaborada pelo autor (2023)

Tabela 44 – Resultados obtidos do SLO+BRKGA entre as instâncias CSP_n15_I40_E70_R12_001 e CSP_n20_I40_E70_R10_010.

| Instância | Valor | SLO+BRKGA | | | |
|-------------------------|----------|---------------|---------------|--------|--------|
| | | Tempo (mm:ss) | Desvio Padrão | Mínimo | Máximo |
| CSP_n15_I40_E70_R12_001 | 31511.33 | 01:03 | 19.28 | 31470 | 31550 |
| CSP_n15_I40_E70_R12_004 | 31531.33 | 01:03 | 23.06 | 31490 | 31570 |
| CSP_n15_I40_E70_R12_008 | 31888.00 | 01:03 | 23.30 | 31850 | 31930 |
| CSP_n15_I40_E70_R18_003 | 33065.33 | 01:06 | 25.13 | 33010 | 33130 |
| CSP_n15_I40_E70_R18_007 | 33115.33 | 01:07 | 40.64 | 33030 | 33210 |
| CSP_n15_I40_E70_R18_010 | 33147.33 | 01:06 | 28.63 | 33090 | 33190 |
| CSP_n15_I70_E40_R00_003 | 27990.00 | 00:55 | 0.00 | 27990 | 27990 |
| CSP_n15_I70_E40_R00_004 | 27990.00 | 00:55 | 0.00 | 27990 | 27990 |
| CSP_n15_I70_E40_R00_009 | 27990.00 | 00:55 | 0.00 | 27990 | 27990 |
| CSP_n15_I70_E40_R12_003 | 31420.00 | 01:04 | 16.93 | 31390 | 31450 |
| CSP_n15_I70_E40_R12_004 | 31371.33 | 01:04 | 23.06 | 31310 | 31410 |
| CSP_n15_I70_E40_R12_009 | 31436.00 | 01:04 | 20.10 | 31410 | 31470 |
| CSP_n15_I70_E40_R18_001 | 32876.00 | 01:07 | 19.43 | 32830 | 32910 |
| CSP_n15_I70_E40_R18_004 | 32846.00 | 01:07 | 26.53 | 32810 | 32890 |
| CSP_n15_I70_E40_R18_010 | 32870.67 | 01:07 | 15.04 | 32850 | 32890 |
| CSP_n15_I70_E70_R04_002 | 36250.67 | 01:07 | 17.50 | 36230 | 36290 |
| CSP_n15_I70_E70_R04_004 | 36260.67 | 01:07 | 29.09 | 36210 | 36310 |
| CSP_n15_I70_E70_R04_008 | 36228.67 | 01:07 | 18.57 | 36190 | 36250 |
| CSP_n15_I70_E70_R12_002 | 38693.33 | 01:14 | 29.25 | 38630 | 38730 |
| CSP_n15_I70_E70_R12_005 | 38692.67 | 01:14 | 37.85 | 38610 | 38750 |
| CSP_n15_I70_E70_R12_009 | 38923.33 | 01:14 | 34.38 | 38830 | 38970 |
| CSP_n15_I70_E70_R20_003 | 40717.33 | 01:18 | 37.41 | 40650 | 40810 |
| CSP_n15_I70_E70_R20_006 | 40759.33 | 01:18 | 29.09 | 40710 | 40850 |
| CSP_n15_I70_E70_R20_008 | 40855.33 | 01:18 | 47.31 | 40770 | 40930 |
| CSP_n20_I40_E70_R00_001 | 49610.00 | 01:36 | 0.00 | 49610 | 49610 |
| CSP_n20_I40_E70_R00_004 | 49650.00 | 01:36 | 0.00 | 49650 | 49650 |
| CSP_n20_I40_E70_R00_009 | 49670.00 | 01:37 | 0.00 | 49670 | 49670 |
| CSP_n20_I40_E70_R10_001 | 54633.33 | 01:50 | 34.29 | 54550 | 54690 |
| CSP_n20_I40_E70_R10_004 | 54647.33 | 01:50 | 24.62 | 54610 | 54710 |
| CSP_n20_I40_E70_R10_010 | 54680.67 | 01:51 | 34.92 | 54590 | 54750 |

Fonte: Elaborada pelo autor (2023)

Tabela 45 – Resultados obtidos do SLO+BRKGA entre as instâncias CSP_n20_I40_E70_R16_001 e CSP_n20_I70_E70_R14_010.

| Instância | Valor | SLO+BRKGA | | | |
|-------------------------|----------|---------------|---------------|--------|--------|
| | | Tempo (mm:ss) | Desvio Padrão | Mínimo | Máximo |
| CSP_n20_I40_E70_R16_001 | 57484.67 | 01:56 | 63.02 | 57370 | 57590 |
| CSP_n20_I40_E70_R16_007 | 57461.33 | 01:55 | 49.71 | 57350 | 57550 |
| CSP_n20_I40_E70_R16_009 | 57524.00 | 01:56 | 55.89 | 57410 | 57630 |
| CSP_n20_I70_E40_R00_002 | 49670.00 | 01:39 | 0.00 | 49670 | 49670 |
| CSP_n20_I70_E40_R00_006 | 49650.00 | 01:38 | 0.00 | 49650 | 49650 |
| CSP_n20_I70_E40_R00_008 | 49690.00 | 01:39 | 0.00 | 49690 | 49690 |
| CSP_n20_I70_E40_R08_002 | 53538.67 | 01:49 | 29.07 | 53470 | 53590 |
| CSP_n20_I70_E40_R08_004 | 53609.33 | 01:49 | 23.37 | 53570 | 53650 |
| CSP_n20_I70_E40_R08_009 | 53495.33 | 01:49 | 20.61 | 53450 | 53530 |
| CSP_n20_I70_E40_R14_003 | 56252.67 | 01:56 | 36.78 | 56170 | 56330 |
| CSP_n20_I70_E40_R14_004 | 56351.33 | 01:55 | 40.97 | 56290 | 56410 |
| CSP_n20_I70_E40_R14_009 | 56289.33 | 01:55 | 38.12 | 56190 | 56350 |
| CSP_n20_I70_E70_R02_001 | 62953.33 | 01:58 | 17.95 | 62890 | 62970 |
| CSP_n20_I70_E70_R02_004 | 63209.33 | 01:58 | 43.35 | 63110 | 63290 |
| CSP_n20_I70_E70_R02_010 | 63161.33 | 01:57 | 23.49 | 63110 | 63210 |
| CSP_n20_I70_E70_R08_002 | 66206.00 | 02:05 | 33.63 | 66150 | 66270 |
| CSP_n20_I70_E70_R08_007 | 66212.00 | 02:05 | 36.64 | 66130 | 66290 |
| CSP_n20_I70_E70_R08_008 | 66293.33 | 02:04 | 58.90 | 66130 | 66410 |
| CSP_n20_I70_E70_R14_002 | 69319.33 | 02:10 | 60.16 | 69150 | 69410 |
| CSP_n20_I70_E70_R14_004 | 69440.00 | 02:10 | 48.65 | 69310 | 69530 |
| CSP_n20_I70_E70_R14_010 | 69316.00 | 02:11 | 77.83 | 69190 | 69490 |

Fonte: Elaborada pelo autor (2023)

APÊNDICE B – ARQUIVOS DO EXEMPLO DO *FRAMEWORK*

B.1 ARQUIVO COM AS CONFIGURAÇÕES

```

m = 4;
n = 2;
d = [ 90.0, 100.0, 100.0, 100.0, 100.0];
dd = [
    [ 10.0, 10.0, 10.0, 20.0, 20.0],
    [ 20.0, 20.0, 20.0, 10.0, 10.0],
    [ 20.0, 20.0, 20.0, 10.0, 10.0],
    [ 10.0, 10.0, 10.0, 20.0, 20.0],
    [ 10.0, 10.0, 10.0, 20.0, 20.0]
];
AC = [
    [ 1, 0, 0, 3],
    [ 4, 3, 0, 1]
];
DC = [
    [ 3, 0, 3, 0],
    [ 4, 2, 2, 0]
];

```

B.2 ARQUIVO COM A SOLUÇÃO

```

1 ( 1, 2) (-1, 0) VY [00]
2 (-1, 0) ( 3, 1) YV [10]
3 ( 0, 3) ( 1, 2) VV [10]
3 ( 1, 1) ( 0, 2) VV [10]
1 ( 1, 3) (-1, 0) VY [10]
2 (-1, 0) ( 1, 1) YV [10]

```

ANEXO A – DADOS DA INSTÂNCIA PARA CSP

A.1 I

```

/*****
*
* Type: Instance for Container Sequencing Problem
*
* Authors: Frank Meisel, Matthias Wichmann
* Address: Martin-Luther-University, Halle, Germany;
*          Technical University Braunschweig, Germany
*
* Paper: Container Sequencing for Quay Cranes with Internal Reshuffles
* Journal: working paper
*
*****/

m = 10; // number of stacks in considered bay
n = 10; // number of tiers in considered bay

// duration of a loaded move of type i in S={'VV','VB','VY','YV','BV'}
d = [ 90.0, 100.0, 100.0, 100.0, 100.0];

// dij duration of an empty move between a loaded move of type i and a loaded move of type
j, i, j in S
dd = [ // VV VB VY YV BV
/*VV*/ [ 10.0, 10.0, 10.0, 20.0, 20.0],
/*VB*/ [ 20.0, 20.0, 20.0, 10.0, 10.0],
/*VY*/ [ 20.0, 20.0, 20.0, 10.0, 10.0],
/*BV*/ [ 10.0, 10.0, 10.0, 20.0, 20.0],
/*YV*/ [ 10.0, 10.0, 10.0, 20.0, 20.0]
];

```

// arrival configuration (incoming master bay plan) - 0 empty, 1 import, 3 reshuffle, 4 fixed

```
AC = [  
    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], // tier i = 10  
    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], // tier i = 9  
    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], // tier i = 8  
    [ 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0], // tier i = 7  
    [ 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0], // tier i = 6  
    [ 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0], // tier i = 5  
    [ 1, 4, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0], // tier i = 4  
    [ 1, 4, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0], // tier i = 3  
    [ 1, 4, 1, 1, 1, 1, 4, 1, 1, 1, 0, 0], // tier i = 2  
    [ 4, 4, 1, 4, 4, 4, 4, 1, 1, 1, 0, 0] // tier i = 1  
];
```

// departure configuration (outgoing master bay plan) - 0 empty, 2 export, 3 reshuffle, 4 fixed

```
DC = [  
    [ 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0], // tier i = 10  
    [ 2, 0, 0, 0, 0, 0, 2, 2, 0, 0, 2, 0], // tier i = 9  
    [ 2, 2, 0, 0, 0, 0, 2, 2, 0, 0, 2, 0], // tier i = 8  
    [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0], // tier i = 7  
    [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0], // tier i = 6  
    [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0], // tier i = 5  
    [ 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0], // tier i = 4  
    [ 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0], // tier i = 3  
    [ 2, 4, 2, 2, 2, 2, 4, 2, 2, 2, 2, 0], // tier i = 2  
    [ 4, 4, 2, 4, 4, 4, 4, 2, 2, 2, 0, 0] // tier i = 1  
];  
// end of file
```