Matheus Knack da Silva

Algoritmo Híbrido GRASP com VNS para o Problema de Alocação de Berços Contínuos

ALEGRE - ES

2025

Matheus Knack da Silva

Algoritmo Híbrido GRASP com VNS para o Problema de Alocação de Berços Contínuos

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Espírito Santo

Centro de Ciências Exatas, Naturais e da Saúde

Departamento de Computação

Orientador: Prof. Dr. Edmar Hell Kampke

ALEGRE - ES

2025

Matheus Knack da Silva

Algoritmo Híbrido GRASP com VNS para o Problema de Alocação de Berços Contínuos

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 20 de Março de 2025.

COMISSÃO EXAMINADORA

Prof. Dr. Edmar Hell Kampke
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Dayan de Castro Bissoli
Universidade Federal do Espírito Santo

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

Agradecimentos

Primeiramente, agradeço a Deus por me dar forças para nunca desistir e por permitir que eu chegasse até aqui. Foi graças a Ele e ao apoio incondicional dos meus pais que tive a oportunidade de concluir o curso e finalizar este Trabalho de Conclusão de Curso (TCC).

Expresso minha profunda gratidão aos meus pais e à minha irmã, que estiveram ao meu lado desde o início dessa jornada, oferecendo apoio e incentivo inabaláveis. Meu agradecimento especial também à minha noiva, que me acompanhou desde o começo do curso até sua conclusão, sempre me motivando nos momentos desafiadores.

Agradeço especialmente ao meu orientador, Edmar, por toda a orientação, paciência e dedicação durante o desenvolvimento e a escrita deste trabalho. Também estendo minha gratidão ao professor Geraldo por fornecer os materiais e as instâncias do problema abordado nesta monografia, contribuindo significativamente para a pesquisa.

Sou grato pelas amizades construídas ao longo dessa caminhada, algumas das quais levarei para toda a vida. Os momentos de companheirismo e descontração foram essenciais para tornar essa jornada mais leve e memorável.

Por fim, meu sincero agradecimento a todas as pessoas que, diretamente ou indiretamente, contribuíram para a minha formação e para a conclusão deste curso.

Resumo

O Problema de Alocação de Berços Contínuos (PAB-CH) é um desafio de otimização presente na gestão de terminais portuários, cujo objetivo é minimizar o tempo total de serviço dos navios. Este trabalho propõe uma abordagem baseada na meta-heurística híbrida GRASP+VNS para resolver o PAB-CH. O método utiliza o GRASP para a construção iterativa de soluções e o VNS como mecanismo de diversificação, explorando diferentes estruturas de vizinhança. Os resultados obtidos demonstram que a abordagem proposta é capaz de gerar soluções de boa qualidade em tempos computacionais reduzidos, sendo considerada uma alternativa viável para a solução do PAB-CH.

Palavras-chave: Problema de Alocação de Berços Contínuos, GRASP, VNS, Meta-heurísticas, Otimização Combinatória.

Abstract

The Continuous Berth Allocation Problem (CBAP) is an optimization challenge in port terminal management, aiming to minimize the total service time of ships. This work proposes an approach based on the hybrid GRASP+VNS metaheuristic to solve the CBAP. The method employs GRASP for the iterative construction of solutions and VNS as an diversification mechanism, exploring different neighborhood structures. The results demonstrate that the proposed approach is capable of generating high-quality solutions within reduced computational times, making it a viable alternative for solving the CBAP.

Keywords: Continuous Berth Allocation Problem, GRASP, VNS, Metaheuristics, Combinatorial Optimization.

Lista de Figuras

Figura 1 – Abordagens do PAB 13
Figura 2 – Representação bidimensional do PAB-CH 14

Figura 3 – Exemplo do movimento Troca Navios 21
Figura 4 – Exemplo do movimento Remove e Realoca Navio 22
Figura 5 – Variáveis referentes ao tempo 24
Figura 6 – Variáveis referentes a atracação dos navios 24

Lista de Tabelas

Tabela 1 – Calibragem dos Parâmetros 27
Tabela 2 – Resultados Obtidos pelo GRASP+VNS 28
Tabela 3 – Comparação com trabalhos da literatura 29
Tabela 4 – Ambientes de testes 29

Lista de Algoritmos

1 Algoritmo GRASP proposto por Feo e Resende (1995) 18

2 Algoritmo VNS adaptado de Mladenović e Hansen (1997) 20

Lista de Abreviaturas e Siglas

GRASP	*Greedy Randomized Adaptive Search Procedure*
LC	Lista de Candidatos
LRC	Lista Restrita de Candidatos
PAB	Problema de Alocação de Berços
PAB-CH	Problema de Alocação de Berços Contínuos
PAB-D	Problema de Alocação de Berços Discreto
UFES	Universidade Federal do Espírito Santo
VNS	*Variable Neighborhood Search*

Sumário

1 Introdução **11**
1.1 O Problema e sua Importância 11
1.2 Objetivos 12
1.2.1 Objetivo Geral 12
1.2.2 Objetivos Específicos 12

2 Revisão de Literatura **13**
2.1 PAB 13
2.2 GRASP 15
2.3 VNS 16

3 Metodologia **18**
3.1 GRASP 18
3.1.1 Construção da solução inicial 19
3.1.2 Busca local 19
3.1.2.1 VNS 19
3.1.2.2 Troca Navios 21
3.1.2.3 Remove e Realoca Navio 21
3.1.2.4 Perturbação 22
3.2 Modelo Matemático do Problema 22

4 Resultados **26**
4.1 Instâncias para o PAB-CH 26
4.2 Parâmetros para o GRASP e VNS 26
4.3 Ambiente de testes 27
4.4 Resultados computacionais 27

5 Conclusões **31**

Referências **32**

1 Introdução

A necessidade de reduzir o tempo de atendimento dos navios nos portos torna-se cada vez mais urgente à medida que o mercado busca maior eficiência e agilidade. Neste contexto, a pesquisa em otimização desempenha um papel crucial ao buscar métodos que não apenas diminuam o tempo de serviço, mas também aumentem a produtividade das operações portuárias.

Nos terminais portuários, os berços são locais onde os navios ficam atracados para serem atendidos, ou seja, onde são realizadas as operações de embarque ou desembarque. Um terminal portuário pode conter diversos berços de tamanhos distintos e com diferentes tempos de atendimento para cada tipo de navio.

Atualmente há um aumento da demanda por berços nos terminais de contêineres. Esse aumento está intrinsecamente ligado ao crescimento do comércio global e ao aumento do tamanho e da frequência dos navios. Como mencionado por Vis e Koster (2003), o tempo de espera dos navios para serem alocados em berços e o tempo de movimentação dos contêineres são fatores cruciais que impactam diretamente nos custos operacionais dos terminais.

Além disso, conforme observado por Imai, Nishimura e Papadimitriou (2003), a alocação eficiente de berços é fundamental para a competitividade dos terminais de contêineres, pois afeta diretamente a produtividade e a receita gerada. Entretanto, o aumento da demanda por berços pode levar a períodos de espera mais longos para os navios e atrasos no manuseio, resultando em custos adicionais para os responsáveis do terminal portuário, como penalidades por não cumprimento de prazos.

Portanto, essa interseção entre o aumento da demanda por berços nos portos, os custos associados ao tempo de atendimento e a pesquisa em otimização, destaca a importância de abordagens inovadoras e eficientes para resolver o Problema de Alocação de Berços (PAB).

1.1 O Problema e sua Importância

Em uma versão do PAB, é possível que apenas um navio seja atendido por vez em cada berço. Contudo, o navio pode ocupar o espaço não alocado pelos berços vizinhos ao berço de atendimento. Dessa forma, navios grandes podem ocupar mais de uma posição, permitindo que navios menores compartilhem o mesmo berço. Assim, se sempre que um navio chegar ao porto, o berço que oferece o menor tempo de atendimento estiver disponível, o tempo de serviço do terminal portuário será reduzido ao mínimo possível.

Segundo Mauri et al. (2016), a versão do Problema de Alocação de Berços Contínuos, que neste trabalho será denominado por PAB-CH, permite uma melhor representação da realidade, principalmente quando alguns berços de atracação possuem diferentes equipamentos, capazes de operar tipos específicos de carga, como exemplo, navios e contêineres de diversos tamanhos.

O PAB-CH é NP-difícil, como mencionado por Mauri et al. (2016) e Lujan et al. (2021). Isso significa que embora nunca tenha sido provado a não existência de um algoritmo eficiente, o mesmo ainda não foi encontrado. Nesse caso, é comum usar heurísticas e métodos de aproximação, cujo objetivo é encontrar uma solução aproximada para o problema com baixo tempo de processamento.

Métodos como o *Variable Neighborhood Search* (VNS) e *Greedy Randomized Adaptive Search Procedure* (GRASP) oferecem abordagens heurísticas eficazes para lidar com instâncias de tamanho realista do problema. Além disso, não foi encontrado nenhum trabalho na literatura que oferecia a combinação dessas meta-heurísticas para a resolução do PAB-CH. O intuito desta combinação é encontrar soluções do PAB-CH que minimizem o tempo total de serviço, que inclui o tempo de espera e atendimento dos navios nos terminais.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como finalidade aplicar a meta-heurística GRASP, com adição do VNS na realização da busca local, para resolução do PAB-CH.

1.2.2 Objetivos Específicos

a) Estudar os dados necessários para resolver o PAB-CH.

b) Implementar a meta-heurística GRASP com VNS para resolver o PAB-CH.

c) Realizar testes de validação e calibragem de parâmetros no algoritmo desenvolvido.

d) Efetuar correções no algoritmo, se forem necessárias.

e) Executar a aplicação desenvolvida utilizando as instâncias propostas na literatura.

f) Avaliar os resultados obtidos e compará-los com os apresentados na literatura.

2 Revisão de Literatura

Nesta seção, será abordado o Problema de Alocação de Berços (PAB), incluindo sua definição e principais abordagens. Além disso, serão apresentados trabalhos da literatura que trataram desse problema, bem como estudos que implementaram as meta-heurísticas GRASP e VNS.

2.1 PAB

O Problema de Alocação de Berços (PAB) consiste em alocar navios em determinados berços ao longo do cais, de forma a satisfazer restrições como: a profundidade da água, distância máxima para a alocação do navio, o tempo de alocação permitido (que se refere a limitações de alguns portos que possuem um tempo máximo para atendimento, pois estão sujeitos a variações na maré) e o tempo de movimentação do navio.

Esses fatores permitem o PAB ser modelado por diferentes abordagens, demonstradas na Figura 1. Uma delas é o Problema de Alocação de Berços Discreto (PAB-D), onde o cais é dividido em berços, e cada navio só pode ser atendido em um único berço por vez. Nessa abordagem, o comprimento dos navios não é considerado e nenhum berço interfere no outro. Já no Problema de Alocação de Berços Contínuo (PAB-C), não há divisões no cais, permitindo que os navios atraquem em qualquer posição do cais. Em outra versão do PAB-C, denominado neste trabalho como PAB-CH, ocorre a hibridização entre o PAB-D e o PAB-C. No PAB-CH o cais é dividido em berços, mas navios grandes podem ocupar mais de um berço, enquanto navios pequenos podem compartilhar o espaço não ocupado em seu berço. O foco deste trabalho foi o PAB-CH.

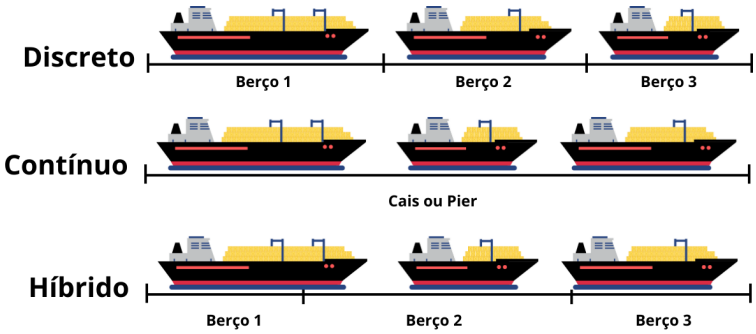

Figura 1 – Abordagens do PAB

Além disso, se for levada em conta a chegada de navios, o problema pode ser abordado como estático ou dinâmico (IMAI; NISHIMURA; PAPADIMITRIOU, 2001). O caso estático pressupõe que todos os navios já estão no porto e prontos para serem atendidos,

já o caso dinâmico pressupõe que os navios podem chegar a qualquer momento no porto para serem atendidos. Neste trabalho foi abordado o PAB-CH na sua versão dinâmica.

Cordeau et al. (2005) e Mauri et al. (2016) mostraram que o PAB pode ser representado em um espaço bidimensional, onde os navios são considerados como retângulos, conforme apresentado na Figura 2. Uma das dimensões do retângulo representa o comprimento do navio com uma margem de segurança, e a outra dimensão representa o tempo de atendimento. Portanto, o PAB pode ser visto como o problema de encaixar esses retângulos no espaço bidimensional sem sobreposições, satisfazendo as janelas de tempo e as restrições de disponibilidade.

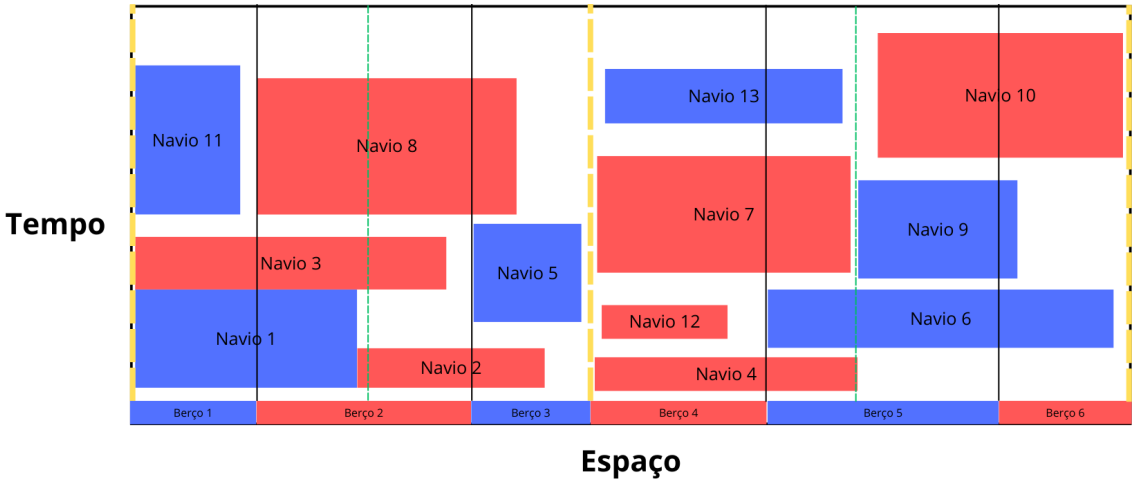

Figura 2 – Representação bidimensional do PAB-CH

A Figura 2 é uma representação do PAB-CH. Cada retângulo representa um navio, onde os retângulos vermelhos representam navios alocados ao berço vermelho mais próximo, e os azuis, ao berço azul mais próximo. As linhas amarelas indicam descontinuidades no cais, as linhas pretas delimitam os diferentes berços e as linhas verdes tracejadas marcam os limites à esquerda e à direita de cada berço. É possível notar ver que um navio pode usar parte dos berços vizinhos, desde que não haja sobreposições nem violações das descontinuidades.

O artigo de Oliveira, Mauri e Lorena (2012a) descreveu o PAB como dinâmico, mas modelado no caso discreto, e propôs uma nova alternativa para resolvê-lo. A alternativa proposta baseia-se na aplicação do método *Clustering Search* (CS), utilizando o *Simulated Annealing* (SA) para a geração de soluções. O CS é um método iterativo que divide o espaço de busca em *clusters*, e é composto por uma meta-heurística para a geração de soluções, um processo de agrupamento e uma heurística de busca local. Os resultados computacionais foram comparados com resultados encontrados na literatura.

Posteriormente, Oliveira, Mauri e Lorena (2012b) propuseram usar novamente o CS em conjunto com o SA, mas agora para resolver o PAB-CH. Os resultados experimentais

indicaram que a abordagem baseada em CS gerou soluções de qualidade em tempos computacionais reduzidos, superando as soluções obtidas por outros trabalhos apresentados na literatura.

Uma obra muito importante na literatura para o PAB foi produzida por Mauri et al. (2016), em que os autores usaram a heurística *Adaptive Large Neighborhood Search* (ALNS) para resolver tanto o PAB-D quanto o PAB-CH, com base no princípio de destruição e reconstrução de soluções. Os experimentos computacionais demonstraram a superioridade relativa do algoritmo ALNS, encontrando todas as soluções ótimas para o PAB-D e novas melhores soluções para o PAB-CH. Além disso, foi utilizado um novo conjunto de instâncias com características contínuas, baseadas nas que foram propostas por Ruiz, Batista e Vega (2012). Por fim, não foi encontrado trabalho na literatura, posterior ao trabalho de Mauri et al. (2016), que tenha abordado o PAB-CH e obtido resultados melhores.

2.2 GRASP

Neves e Ochi (2006) implementaram um método iterativo com memória adaptativa utilizando conceitos da heurística GRASP na solução de um problema de roteamento de veículos com múltiplos depósitos. Os resultados experimentais deste estudo demonstram a eficiência da versão adaptativa em comparação com o GRASP padrão. Similarmente, Mestria, Ochi e Martins (2009) aplicaram heurísticas híbridas, que incorporam memória adaptativa à versão tradicional do GRASP, para resolver o Problema do Caixeiro Viajante com Grupamentos (PCVG), uma variação do Problema do Caixeiro Viajante (PCV). Os resultados computacionais mostram que as versões adaptativas do algoritmo superam a versão tradicional do GRASP.

Kampke (2010) aplicou o GRASP ao problema de programação de tarefas em máquinas paralelas. O trabalho abordou o sequenciamento de tarefas em máquinas paralelas, onde os tempos de preparação dependem da sequência e do número de recursos utilizados. Devido à complexidade combinatória do problema, foi inicialmente proposto um algoritmo baseado na meta-heurística GRASP, no qual o parâmetro de aleatoriedade na fase de construção é auto-ajustado conforme as soluções previamente encontradas (GRASP Reativo). Os resultados obtidos são comparados com os melhores resultados disponíveis na literatura, demonstrando que o uso do GRASP melhorou, em média, 9,14% os resultados da literatura, comprovando sua viabilidade na solução de problemas práticos na indústria.

Barbosa e Souza (2011) abordaram o problema de programação de cursos universitários com base em currículos, utilizando uma meta-heurística híbrida. A solução inicial é gerada por GRASP, e o refinamento é realizado pela meta-heurística *Iterated Local Search* (ILS) com relaxamento. O objetivo é alocar períodos e salas para todas as aulas,

evitando conflitos entre disciplinas do mesmo currículo. O problema é baseado em aspectos e instâncias da segunda edição do *International Timetabling Competition* (ITC), realizado em 2007. Os resultados computacionais foram comparados com a literatura, destacando a influência positiva do procedimento de relaxamento no desempenho do algoritmo híbrido proposto.

2.3 VNS

Reis e Cunha (2011) abordaram o problema da distribuição física, envolvendo o agrupamento e a alocação de entregas a uma frota de veículos, com o objetivo de minimizar o custo total de frete. O problema surge em situações práticas onde os pontos de entrega estão próximos, tornando as distâncias entre paradas irrelevantes para o cálculo do custo total. Esse problema foi modelado como um problema de *bin-packing* unidimensional com *bins* de tamanho variável (*Variable Sized Bin-Packing*), onde veículos de diferentes capacidades e custos são utilizados para alocar um conjunto de cargas de forma a minimizar o custo total. O trabalho propõe uma heurística baseada no VNS para resolver esse problema. Experimentos computacionais com instâncias derivadas de *benchmarks* da literatura mostram que o algoritmo proposto alcançou a solução ótima na maioria dos testes e, nos demais, obteve soluções próximas ao limitante inferior. A heurística baseada em VNS se destacou por sua rapidez e, em problemas de grande porte, foi a única a encontrar soluções, ficando menos de 1% acima dos limitantes inferiores.

Sakiyama (2014) resolveu o Problema de Escalonamento de Motoristas (PEM), que visa criar uma escala de trabalho para motoristas cobrindo uma frota de veículos com o menor custo, atendendo às restrições das leis trabalhistas, os acordos sindicais e as normas da empresa. Esse processo é crucial no planejamento operacional de empresas de transporte coletivo, devido ao alto custo associado aos motoristas. O trabalho de Sakiyama (2014) propõe uma abordagem envolvendo dois procedimentos de busca local, o Procedimento de Corte de Recombinações (PCR) e o *k-swap*, combinados com a meta-heurística VNS. Foram utilizadas instâncias reais, com mais de 2300 viagens, e instâncias aleatórias, derivadas das reais, para validação. Os experimentos demonstraram a eficiência da meta-heurística VNS em grandes instâncias, com resultados superiores aos obtidos por trabalhos que usaram apenas PCR e *k-swap*.

Fernandes et al. (2020) abordou a resolução do Problema Integrado de Carregamento e Roteamento de Veículos com Frota Heterogênea e Janela de Tempo (PICRVFHJT), um tema recorrente na otimização combinatória. O objetivo é minimizar os custos de transporte, considerando o carregamento de itens e restrições como capacidade de peso e área dos veículos, bem como o intervalo de tempo para atendimento dos clientes. É apresentado um modelo matemático para o problema. O estudo propõe um algoritmo

baseado na meta-heurística VNS, que utiliza o método *Variable Neighborhood Descent* (VND) para a busca local. Foram criadas 35 instâncias para testes. As instâncias foram resolvidas inicialmente pelo modelo matemático usando o *software* CPLEX e, em seguida, pelo método VNS-VND. Os resultados demonstram um bom desempenho do método, encontrando soluções ótimas e em tempos computacionais rápidos, aumentando a eficiência e aplicabilidade do método para situações reais.

Como dito anteriormente, as metodologias GRASP e VNS demonstram um bom desempenho na resolução de outros problemas na área de otimização combinatória, sendo capazes de lidar com instâncias de tamanho realista de forma eficiente. No entanto, não foi encontrado nenhum trabalho na literatura que tenha explorado a hibridização desses métodos para resolver o PAB-CH. Dessa forma, é justificada a combinação dessas metaheurísticas com o objetivo de aproveitar as vantagens individuais de cada uma, buscando um equilíbrio entre diversificação e intensificação na busca por soluções de qualidade para o PAB-CH.

3 Metodologia

Nesta seção é apresentada a implementação da meta-heurística GRASP, combinada com a meta-heurística VNS, usada como estratégia de busca local, para resolver o PAB-CH. Além disso, é apresentado o modelo matemático que formaliza o problema, destacando suas variáveis, definições, função objetivo e restrições.

3.1 GRASP

A meta-heurística GRASP foi proposta por Feo e Resende (1995), e já se provou amplamente eficaz em outros problemas, conforme pode ser observado na Seção 2.2. O Algoritmo 1 apresenta o pseudocódigo genérico da meta-heurística GRASP.

Algoritmo 1: Algoritmo GRASP proposto por Feo e Resende (1995)

Entrada: Max_{Iter}, α
Saída: Solução S^{Melhor}

1 **início**
2 $\quad FO^{Melhor} \leftarrow \infty$;
3 $\quad$ **para** $j \leftarrow 1$ **até** Max_{Iter} **faça**
4 $\quad\quad S^{Inicial} \leftarrow GeraSolucaoInicial(\alpha)$;
5 $\quad\quad S^{Atual} \leftarrow BuscaLocal(S^{Inicial})$;
6 $\quad\quad$ **se** $f(S^{Atual}) < FO^{Melhor}$ **então**
7 $\quad\quad\quad S^{Melhor} \leftarrow S^{Atual}$;
8 $\quad\quad\quad FO^{Melhor} \leftarrow f(S^{Atual})$;
9 $\quad\quad$ **fim se**
10 $\quad$ **fim para**
11 **fim**

O algoritmo GRASP consiste basicamente em duas fases: a construção da solução inicial (linha 4) e o refinamento da solução (linha 5), comumente conhecido como busca local.

No Algoritmo 1 existem dois parâmetros: o número máximo de iterações (Max_{Iter}) e o fator de aleatoriedade (α). Analisando minuciosamente o algoritmo, podemos notar que, enquanto não atingir o Max_{Iter}, ele constrói repetidamente uma solução $S^{Inicial}$ (linha 4), que é melhorada por um procedimento de busca local (linha 5). A melhor solução encontrada até o momento é armazenada (linhas 6 a 9).

3.1.1 Construção da solução inicial

A solução é construída iterativamente combinando estratégias gulosa e aleatória. Para isso são usadas duas listas: a Lista de Candidatos (LC) e a Lista Restrita de Candidatos (LRC). Entre LC e LRC, o fator de aleatoriedade (α) atua como parâmetro de mediação. A seguir, são apresentadas as definições desses conceitos:

- Lista de Candidatos (LC): É uma lista que contém todos os elementos (ou candidatos) que podem ser inseridos na solução do problema. A LC é sempre mantida ordenada com base em um critério, geralmente relacionado ao impacto que cada candidato terá na qualidade da solução em construção.
- Lista Restrita de Candidatos (LRC): É uma sublista formada pelos elementos mais bem avaliados da LC, conforme o critério de ordenação utilizado. Em outras palavras, a LRC é composta pelos primeiros elementos da LC.
- α: É um fator de ponderação que determina o tamanho da LRC e, consequentemente, quais candidatos serão considerados na seleção.

Neste trabalho, primeiramente uma função ordena os navios em ordem crescente com base na hora de chegada, inserindo-os de forma ordenada em uma lista L. Iterativamente, o primeiro navio (navio x) da lista L é retirado. A partir disso é gerado uma lista de berços ordenada crescentemente pelo tempo de atendimento do navio x, formando a LC. Em seguida, uma sublista contendo os $\alpha\%$ melhores berços é definida, constituindo a LRC. Por fim, um berço é escolhido aleatoriamente dentro da LRC para o navio x ser alocado. Esse processo se repete até que a solução inicial do problema contenha todos os navios alocados em berços.

3.1.2 Busca local

A estratégia de busca local utilizada neste trabalho foi a meta-heurística VNS.

3.1.2.1 VNS

A meta-heurística VNS foi desenvolvida por Mladenović e Hansen (1997), que propuseram essa técnica para demonstrar que era possível encontrar soluções eficientes por meio de mudanças sistemáticas de vizinhança. O Algoritmo 2 apresenta o pseudocódigo da meta-heurística VNS, com os seguintes parâmetros:

- Solução inicial ($S_{inicial}$).
- Quantidades diferentes de gerar vizinhos (Q_{viz}).

- Numero de iterações (N_{int}).

Algoritmo 2: Algoritmo VNS adaptado de Mladenović e Hansen (1997)

```
Entrada: Solução S_inicial, Q_viz, N_int
Saída: Solução S^Melhor
1  início
2      S_melhor ← S_inicial;
3      S_atual ← S_inicial;
4      q ← 1;
5      v ← 1;
6      enquanto q ≤ N_int faça
7          S_atual ← geraVizinho(S_atual, v);
8          se S_atual < S_melhor então
9              S_melhor ← S_atual;
10             v ← 1;
11         fim se
12         senão
13             S_atual ← S_melhor;
14             se v < Q_viz então
15                 v ← v + 1;
16             fim se
17             senão
18                 v ← 1;
19             fim se
20         fim se
21         q ← q + 1
22     fim enquanto
23 fim
```

O Algoritmo 2 começa com uma solução inicial ($S_{inicial}$), gerada na fase construtiva do GRASP, e com os valores de Q_{viz} e N_{int} definidos. A cada iteração, o algoritmo verifica se o número de iterações (N_{int}) foi alcançado. Caso contrário, o algoritmo executa a função $geraVizinho$, que envia a solução atual (S_{atual}) e v. O valor de v identifica o método de vizinhança a ser utilizado. A solução atual (S_{atual}) é então atualizada com uma solução vizinha, gerada por diferentes métodos de troca, com o objetivo de melhorar a solução. Se S_{atual} for superior à melhor solução encontrada até o momento (S_{melhor}), esta é atualizada com a nova melhor solução, e o algoritmo reinicia a partir dela. Caso contrário, S_{atual} é descartada, e o algoritmo continua a partir da melhor solução encontrada até o momento, mas usando outro método de troca.

No algoritmo VNS implementado neste trabalho, caso todos os Q_{viz} diferentes métodos de troca tenham sido usados sem melhorar S_{melhor}, o valor de v é reiniciado com o primeiro movimento de troca.

Neste trabalho foram implementados três movimentos de troca ($Q_{viz} = 3$) para geração de soluções vizinhas pelo VNS. A seguir são descritos os três movimentos:

3.1.2.2 Troca Navios

A primeira abordagem para a geração de vizinhos consiste na seleção aleatória de dois berços (y_1 e y_2) dentre todos os existentes no problema. Em seguida, é verificado se o berço y_1 selecionado possui ao menos um navio alocado. Caso o berço escolhido não contenha nenhum navio, um novo berço é selecionado aleatoriamente, repetindo esse processo até que seja encontrado um berço ocupado.

Uma vez identificado o berço y_1 com navios, um dos navios ali alocados (x_1) é selecionado aleatoriamente. Posteriormente, um berço y_2 é escolhido, podendo ser o mesmo berço que y_1 ou um diferente. No entanto, o navio selecionado para esse segundo berço (x_2) deve ser distinto de x_1. A seguir, os navios x_1 e x_2 são trocados de posição. Dessa forma, essa abordagem pode resultar em diferentes configurações de alocação, contribuindo para a diversificação na busca por soluções.

A Figura 3 ilustra o movimento de Troca de Navios, no qual os berços 2 e 3 são selecionados como y_1 e y_2, respectivamente. Nessa troca, os navios 3 e 7, correspondentes a x_1 e x_2, têm suas posições invertidas, de modo que o navio 7 passa a ocupar o berço 2, enquanto o navio 3 é realocado para o berço 3.

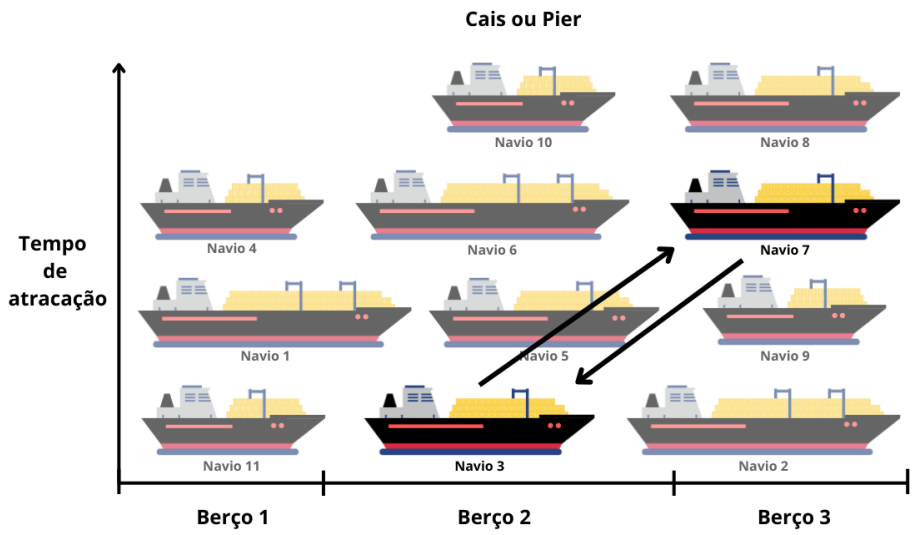

Figura 3 – Exemplo do movimento Troca Navios

3.1.2.3 Remove e Realoca Navio

A segunda estratégia de geração de vizinhos também envolve a seleção aleatória de dois berços, y_1 e y_2, sendo um deles (y_1) destinado à remoção de um navio e o outro (y_2) à realocação do mesmo. Em seguida, um navio é escolhido aleatoriamente dentre aqueles

presentes em y_1. Após isso, é verificado se o berço y_2 possui as condições necessárias para atender o navio em questão. Caso essa condição seja satisfeita, o navio é transferido para o berço y_2, sendo alocado após o último navio já alocado nesse berço.

Esse procedimento visa explorar diferentes possibilidades de alocação, buscando melhorar a distribuição dos navios entre os berços, reduzindo possíveis congestionamentos e otimizando o aproveitamento dos recursos disponíveis.

A Figura 4 ilustra o movimento de Remoção e Realocação de Navio. Nesse exemplo, o berço 2 é selecionado como y_1, de onde um navio será removido, enquanto o berço 3 é escolhido como y_2 para realocação. O navio 3 selecionado é transferido para o berço 3, desde que as condições necessárias para a realocação sejam atendidas.

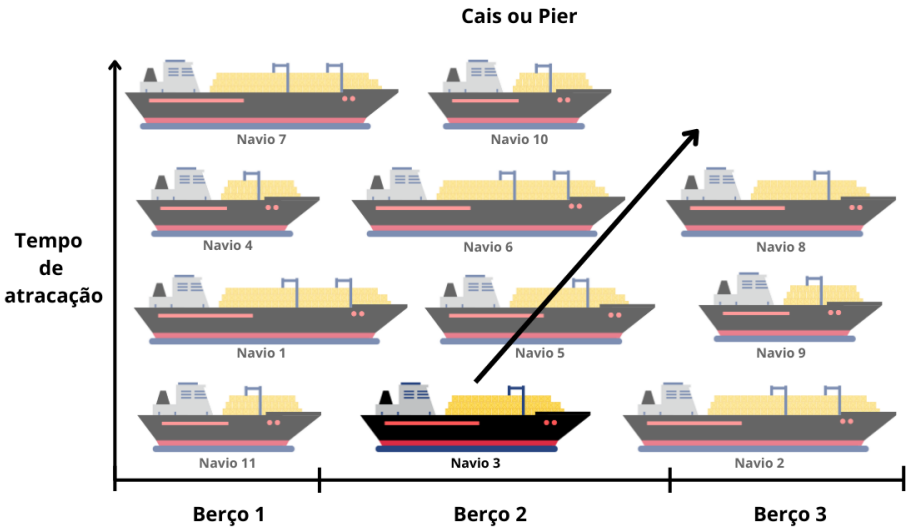

Figura 4 – Exemplo do movimento Remove e Realoca Navio

3.1.2.4 Perturbação

A terceira estratégia de geração de vizinhos é caracterizada por realizar uma perturbação na solução atual, com o objetivo de evitar que o algoritmo fique preso em mínimos locais. Para isso, essa abordagem executa o movimento Troca Navios uma quantidade η de vezes consecutivas.

Dessa forma, a solução é significativamente alterada, permitindo que o algoritmo explore regiões diferentes do espaço de busca e aumentando a probabilidade de encontrar soluções melhores.

3.2 Modelo Matemático do Problema

Nesta seção é apresentado o modelo matemático do PAB-CH, com as seguintes definições:

- N representa o conjunto de navios.
- K representa o conjunto de berços.
- i representa um navio.
- k representa um berço.
- t_i^k é o tempo de atendimento do navio i no berço k.
- T_i^k é o horário de atracação do navio i no berço k.
- z_i refere-se ao tamanho que o navio i possui.
- a_i refere-se ao tempo previsto de chegada do navio i.
- b_i refere-se ao tempo limite para atendimento do navio i.
- s^k representa o tempo de abertura do berço k.
- e^k representa tempo de fechamento do berço k.
- P_i representa a posição de atracação do navio i.
- L^k representa o berço a esquerda do berço k que pode ser usado.
- R^k representa o berço a direita de k que pode ser usado.
- P_k^s representa o limite da posição de atracação inicial do berço k.
- P_k^f representa o limite da posição de atracação final do berço k.

Dessa forma, a função objetivo do PAB-CH é descrito por uma função f que busca minimizar o tempo de atendimento de todos os navios alocados no cais em uma solução s. A função objetivo f é definida da seguinte forma:

$$\text{Minimizar } f(s) = \sum_{i \in N} (T_i^k - a_i + t_i^k) \text{ se } i \text{ atracou em } k \tag{3.1}$$

A Figura 5 apresenta uma representação das variáveis da função objetivo, enquanto que a Figura 6 demonstra visualmente as definições do modelo matemático.

Para que uma solução do PAB-CH seja considerada viável, deve-se considerar as seguintes restrições:

i. Todos os navios devem ser atendidos.

ii. Cada navio só pode ser atendido uma única vez, em um berço capaz de atendê-lo.

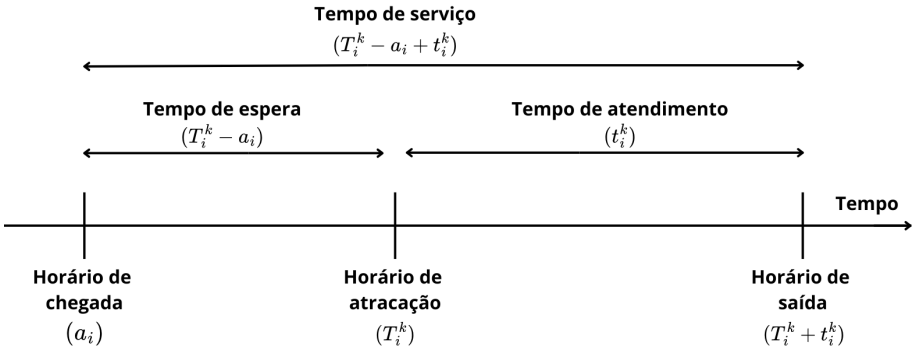

Figura 5 – Variáveis referentes ao tempo

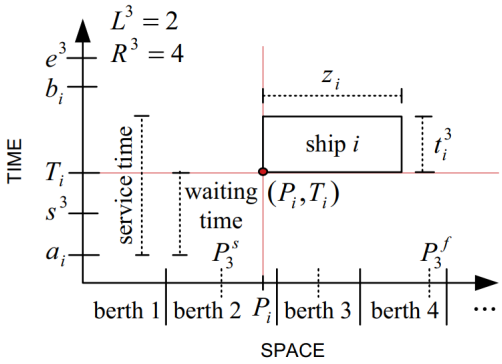

Figura 6 – Variáveis referentes a atracação dos navios

Fonte: Mauri et al. (2016).

iii. Cada navio só pode atracar após a sua chegada no porto $T_i^k \geq a_i$.

iv. O atendimento de cada navio deve terminar antes do seu tempo limite $(T_i^k + t_i^k \leq b_i)$.

v. Todos os navios devem ser atendidos no período de funcionamento dos berços, ou seja, $s^k \leq T_i^k$ e $T_i^k + t_i^k \leq e^k$, $\forall\, k \in K$ e i atracado em k.

vi. O tamanho dos berços, navios e "berços vizinhos" devem ser respeitados, ou seja, não pode haver sobreposições entre os navios atracados, e cada navio só pode usar parte do berço direito e parte do berço esquerdo, caso essas partes existam.

vii. As descontinuidades devem ser respeitadas.

A função objetivo (3.1) visa minimizar a soma do tempo de serviço de cada navio, que é influenciado pelo tempo de espera e pelo tempo de atendimento de cada navio no cais. As restrições (i) e (ii) garantem que cada navio seja atendido por um único berço, e que o berço seja adequado para o navio em questão. As restrições (iii), (iv) e (v) asseguram que todos os navios sejam atendidos apenas após sua chegada ao porto, dentro do período de funcionamento dos berços e respeitando seus respectivos limites de tempo. As restrições

(vi) e (vii) referem-se ao tamanho dos berços, estipulando que tanto as dimensões, quanto as descontinuidades, devem ser respeitadas.

4 Resultados

Nesta seção são apresentados e discutidos os resultados obtidos neste trabalho.

4.1 Instâncias para o PAB-CH

Para avaliar o desempenho do método proposto neste estudo foram usadas 30 instâncias distintas, cada uma composta por 60 navios e 13 berços. Esse conjunto corresponde a apenas um dos conjuntos de instâncias usadas por Mauri et al. (2016). O processo de geração dessas instâncias é aleatório, sendo descrito por Cordeau et al. (2005).

A seleção dessas instâncias se deve à facilidade de acesso, à dificuldade em obter dados reais dos portos para compor novas instâncias, e ao conhecimento prévio dos valores ótimos de referência.

4.2 Parâmetros para o GRASP e VNS

Os parâmetros empregados nos algoritmos foram α e q. O parâmetro α define o tamanho da LRC utilizada na etapa de construção de soluções do GRASP, enquanto o parâmetro q determina a quantidade de iterações do VNS.

Para o parâmetro q, os valores foram definidos com base na quantidade total de berços ($|K|$), na quantidade de navios ($|N|$) e no parâmetro β definido nessa seção. Dessa forma, o valor de q é dado por $q = \beta * |K| * |N|$.

O critério de parada do algoritmo GRASP (Max_{Iter}) foi substituído pela limitação do tempo de execução, sendo fixado um tempo máximo de 120 segundos por execução, assim como no algoritmo ALNS proposto por Mauri et al. (2016).

Os valores de α e β foram calibrados empiricamente por meio de experimentação. Para isso, foram realizadas execuções utilizando as instâncias i01, i15 e i30, sendo que cada instância foi executada cinco vezes para cada combinação dos conjuntos de valores de α e β.

Os testes foram conduzidos considerando o parâmetro α com os valores $\{0,2; 0,4; 0,6; 0,8\}$, enquanto β assumiu os valores $\{50; 100; 150; 200\}$. Os resultados obtidos são apresentados na Tabela 1.

Com base nessa análise, constatou-se que a configuração mais eficiente foi obtida com $\alpha = 0,2$ e $\beta = 50$, atingindo um valor médio de f igual a 1592. Esse resultado pode ser atribuído ao fato de que essa configuração considera os 20% melhores berços, priorizando

Tabela 1 – Calibragem dos Parâmetros

α	β	**Média total**
0,2	50	1592
	100	1598
	150	1597
	200	1610
0,4	50	1621
	100	1620
	150	1628
	200	1649
0,6	50	1626
	100	1637
	150	1654
	200	1648
0,8	50	1634
	100	1648
	150	1653
	200	1661

aqueles que minimizam o tempo de atendimento do navio. Além disso, a escolha de um valor reduzido para β permitiu ao GRASP realizar um maior número de iterações, ampliando a diversidade das soluções exploradas.

Diante da consistência e viabilidade dos resultados obtidos com esses valores, foram definidos como parâmetros finais do algoritmo GRASP+VNS os valores de $\alpha = 0,2$ e $q = 39000$.

O valor η, responsável por controlar o número de execuções da função Troca Navios durante a Perturbação, foi definido de forma escalável em relação à quantidade de berços ($|K|$) do problema, ou seja, $\eta = |K|$.

4.3 Ambiente de testes

Para realizar os experimentos computacionais foi utilizado um computador com processador AMD Ryzen 5 5600 de 3.5GHz com 6 núcleos e 12 threads, sistema operacional Windows 11 e 48GB DDR4 3200 MHz de memória RAM. Toda a implementação foi desenvolvida na linguagem de programação C++, e está disponível em: `https://github.com/mtknack/TCC`

4.4 Resultados computacionais

Após a parametrização dos valores de α e q foram realizados os testes finais nas 30 instâncias, sendo que cada instância foi submetida a um total de 10 execuções.

Em seguida, foram analisadas as soluções obtidas nessas execuções, sendo que todas as restrições foram atendidas em todas as execuções, ou seja, todas as soluções encontradas pelo método proposto são viáveis. A Tabela 2 sintetiza essas informações.

Tabela 2 – Resultados Obtidos pelo GRASP+VNS

Instância	f_{best}	f_{avg}	$\Delta(\%)$	ω(s)
i01	1678	1694,7	1,00	39,83
i02	1388	1404,1	1,16	84,11
i03	1269	1293,2	1,91	67,04
i04	1439	1454,8	1,10	8,92
i05	1352	1369,3	1,28	4,81
i06	1490	1517,0	1,81	83,76
i07	1373	1392,7	1,43	31,28
i08	1495	1515,1	1,34	41,63
i09	1729	1746,6	1,02	106,26
i10	1399	1409,1	0,72	44,26
i11	1644	1659,6	0,95	36,49
i12	1542	1577,7	2,32	4,75
i13	1504	1525,3	1,42	32,59
i14	1354	1366,9	0,95	63,23
i15	1413	1435,4	1,59	116,55
i16	1628	1649,8	1,34	27,06
i17	1342	1357,6	1,16	92,61
i18	1614	1639,2	1,56	98,87
i19	1647	1674,9	1,69	72,34
i20	1444	1477,4	2,31	38,18
i21	1516	1544,6	1,89	61,22
i22	1517	1552,3	2,33	111,34
i23	1464	1486,5	1,54	62,78
i24	1439	1450,5	0,80	77,69
i25	1608	1632,2	1,50	49,77
i26	1525	1549,9	1,63	87,46
i27	1421	1431,6	0,75	95,57
i28	1551	1580,0	1,87	87,40
i29	1387	1404,6	1,27	113,48
i30	1584	1602,0	1,14	91,71
Média	**1491,87**	**1513,2**	**1,43**	**64,43**

Na Tabela 2, a coluna f_{best} apresenta a melhor solução encontrada. A coluna f_{avg} representa a média dos valores obtidos ao longo das 10 execuções de cada instância. Já a coluna $\Delta(\%)$ expressa, em termos percentuais, o desvio da média em relação ao melhor valor encontrado. Por fim, a coluna ω(s), medida em segundos, refere-se ao tempo necessário para encontrar a solução f_{best}.

A análise da Tabela 2 revela que, no método proposto, o valor médio de Δ foi 1,43. No entanto, apenas em 3 das 30 instâncias o valor de Δ foi superior à duas unidades.

Ao examinar os valores de ω, percebe-se que, em média, as soluções são obtidas com aproximadamente metade do tempo limite de execução. Isso sugere que as melhores soluções são encontradas de forma eficiente, sem necessidade de utilizar todo o tempo disponível. Para uma análise mais detalhada, podemos agrupar as instâncias em três categorias: $\omega \leq 40$, $40 < \omega < 80$ e $\omega \geq 80$. Nesse contexto, constatou-se que apenas 12 instâncias demandaram mais de 80 segundos para alcançar sua melhor solução, das quais apenas 4 excederam os 100 segundos.

Por outro lado, no primeiro grupo, nove melhores soluções foram encontradas em menos de 40 segundos, enquanto outras nove exigiram entre 40 e 80 segundos. Em termos percentuais, 60% das soluções foram obtidas dentro dos primeiros 80 segundos, assim, reforçando a eficiência computacional do método proposto.

Na Tabela 3 é apresentada a comparação dos resultados obtidos neste trabalho com outros estudos disponíveis na literatura. Na primeira coluna da Tabela 3 são apresentados os métodos usados na comparação, sendo esses o ALNS proposto por Mauri et al. (2016), o CS proposto por Oliveira, Mauri e Lorena (2012b), e por fim, o método GRASP+VNS proposto neste trabalho.

Ainda na Tabela 3, a coluna $f_{best-avg}$ representa a média das melhores soluções (f_{best}). A coluna ω_{avg}(s) refere-se à média do tempo (ω(s)), em segundos, necessário para encontrar as melhores soluções. Por fim, a coluna Δ_{avg}(%) define a média de Δ(%).

Tabela 3 – Comparação com trabalhos da literatura

	$f_{best-avg}$	ω_{avg}(s)	Δ_{avg}(%)
ALNS	1414,27	73,03	0,82
CS	1412,13	73,89	1,17
GRAPS+VNS	1491,87	64,43	1,43

Na Tabela 4 é apresentada uma comparação do ambiente de teste deste trabalho com os ambientes de testes dos trabalhos de Mauri et al. (2016) e Oliveira, Mauri e Lorena (2012b). Essa comparação é importante para destacar o impacto que os diferentes ambientes de testes podem gerar nos resultados.

Tabela 4 – Ambientes de testes

	Processador	Frequência (GHz)	RAM (GB)
ALNS	Intel Core i5	2.67	4
CS	Intel core 2 duo	1.66	2
GRAPS+VNS	AMD Ryzen 5	3.5	48

A análise da Tabela 3 revela que, embora o método proposto não tenha alcançado resultados tão bons quanto os demais métodos, ele se destacou pelo menor tempo médio de execução para encontrar as melhores soluções. Além disso, a diferença média entre as

soluções obtidas pelo GRASP+VNS e os demais métodos foi de no máximo 79,74. Isso indica que apesar de não ser a melhor abordagem, o método proposto representa uma alternativa viável para a resolução do PAB-CH.

5 Conclusões

Este trabalho abordou o Problema de Alocação de Berços (PAB) utilizando um modelo adaptado da proposta de Cordeau et al. (2005). O problema foi tratado em sua forma contínua no que diz respeito às restrições espaciais, enquanto as restrições temporais foram consideradas de forma dinâmica. O principal objetivo foi minimizar o tempo total de serviço dos navios no porto, proporcionando uma alocação mais eficiente dos recursos disponíveis.

Para solucionar o PAB-CH foi aplicada a meta-heurística híbrida GRASP+VNS, combinando a flexibilidade do GRASP com a capacidade de diversificação do VNS, que explorou diferentes estruturas de vizinhança para refinar as soluções.

Os resultados obtidos demonstraram a eficiência da estratégia proposta. O VNS mostrou-se fundamental para aprimorar as soluções geradas pela heurística construtiva do GRASP, garantindo que todas as restrições do problema fossem atendidas. Além disso, o tempo computacional para encontrar as melhores soluções foi relativamente baixo, com média inferior a 65 segundos. É observado que os valores médios das soluções permaneceram próximos aos melhores resultados encontrados na literatura, demonstrando o bom desempenho do método proposto.

Como sugestões para trabalhos futuros, recomenda-se um ajuste mais refinado dos valores de β e η, explorando novos valores ou introduzindo parâmetros adicionais para a busca de vizinhança. Uma possível melhoria seria a implementação de estratégias de troca baseadas em critérios mais adequados, além de ajustar o momento em que os vizinhos são gerados ao longo da execução do algoritmo. Isso evitaria o descarte prematuro de soluções promissoras, que poderiam ser aprimoradas com trocas específicas.

Referências

BARBOSA, S. H. D.; SOUZA, S. R. de. Resolução do problema de programação de cursos universitários baseada em currículos via uma meta-heurística híbrida grasp-ils-relaxado. In: *Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional*. Rio de Janeiro, RJ, Brasil: SOBRAPO, 2011. v. 43, p. 1976–1987.

CORDEAU, J. F. et al. Models and tabu search heuristics for the berth-allocation problem. *Transportation science*, v. 39, n. 4, p. 526–538, 2005.

FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, v. 6, n. 2, p. 109–133, 1995.

FERNANDES, L. et al. Resolução do problema integrado de carregamento e roteamento de veículos com frota heterogênea e janela de tempo por meio da meta-heurística vns-vnd. In: *Anais do Encontro Nacional de Engenharia de Produção*. São José dos Campos, SP, Brasil: ABEPRO, 2020. v. 40.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, v. 35, n. 4, p. 401–417, 2001.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. Berth allocation with service priority. *Transportation Research Part B: Methodological*, v. 37, n. 5, p. 437–457, 2003.

KAMPKE, E. H. *Metaheurísticas para o problema de programação de tarefas em máquinas paralelas com tempos de preparação dependentes da sequência e de recursos*. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Viçosa, 2010.

LUJAN, E. et al. A fuzzy optimization model for the berth allocation problem and quay crane allocation problem (bap + qcap) with n quays. *Journal of Marine Science and Engineering*, v. 9, n. 2, p. 152, 2021.

MAURI, G. R. et al. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Computers Operations Research*, v. 70, p. 140–154, 2016.

MESTRIA, M.; OCHI, L. S.; MARTINS, S. L. Grasp com memória adaptativa para o problema do caixeiro viajante com grupamentos. In: *Anais do Encontro Nacional de Engenharia de Produção*. São José dos Campos, SP, Brasil: ABEPRO, 2009. v. 29.

MLADENOVIć, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, n. 11, p. 1097–1100, 1997.

NEVES, T. A.; OCHI, L. S. Grasp com memória adaptativa na solução de um problema de roteamento de veículos com múltiplas origens. In: *Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional*. Rio de Janeiro, RJ, Brasil: SOBRAPO, 2006. v. 38, p. 1323–1332.

OLIVEIRA, R. M.; MAURI, G. R.; LORENA, L. A. N. Clustering search for the berth allocation problem. *Expert Systems with Applications*, v. 39, n. 5, p. 5499–5505, 2012.

OLIVEIRA, R. M.; MAURI, G. R.; LORENA, L. A. N. Clustering search heuristic for solving a continuous berth allocation problem. In: HAO, J.-K.; MIDDENDORF, M. (Ed.). *European conference on evolutionary computation in combinatorial optimization*. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2012. p. 49–62.

REIS, J. V. A.; CUNHA, C. B. Uma nova heurística baseada na meta-heurística vns para o problema de agrupamento de entregas em veículos de uma frota heterogênea. *Transportes*, v. 19, n. 3, p. 75–84, 2011.

RUIZ, E. L.; BATISTA, B. M.; VEGA, J. M. M. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, v. 25, n. 6, p. 1132–1141, 2012.

SAKIYAMA, R. Z. *Avaliação da Meta-heurística VNS para um problema de planejamento operacional do transporte público*. Dissertação (Mestrado em Ciência da Computação) — Universidade Estadual de Maringá, 2014.

VIS, I. F. A.; KOSTER, R. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, v. 147, n. 1, p. 1–16, 2003.