

Optimization of Nesting Systems in Shipbuilding: A Review

Sari Wanda Rulita¹, Gunawan¹ and Muzhoffar Dimas Angga Fakhri¹

Received: 24 April 2024 / Accepted: 14 October 2024

© Harbin Engineering University and Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract

This review article provides a comprehensive analysis of nesting optimization algorithms in the shipbuilding industry, emphasizing their role in improving material utilization, minimizing waste, and enhancing production efficiency. The shipbuilding process involves the complex cutting and arrangement of steel plates, making the optimization of these operations vital for cost-effectiveness and sustainability. Nesting algorithms are broadly classified into four categories: exact, heuristic, metaheuristic, and hybrid. Exact algorithms ensure optimal solutions but are computationally demanding. In contrast, heuristic algorithms deliver quicker results using practical rules, although they may not consistently achieve optimal outcomes. Metaheuristic algorithms combine multiple heuristics to effectively explore solution spaces, striking a balance between solution quality and computational efficiency. Hybrid algorithms integrate the strengths of different approaches to further enhance performance. This review systematically assesses these algorithms using criteria such as material dimensions, part geometry, component layout, and computational efficiency. The findings highlight the significant potential of advanced nesting techniques to improve material utilization, reduce production costs, and promote sustainable practices in shipbuilding. By adopting suitable nesting solutions, shipbuilders can achieve greater efficiency, optimized resource management, and superior overall performance. Future research directions should focus on integrating machine learning and real-time adaptability to further enhance nesting algorithms, paving the way for smarter, more sustainable manufacturing practices in the shipbuilding industry.

Keywords Cutting plate; Nesting algorithms; Nesting optimization; Shipbuilding efficiency; Algorithmic optimization

1 Introduction

The shipbuilding industry is widely regarded as a highly promising sector owing to its significant contributions to the global economy (Kamola-Cieslik, 2021). The industry supports numerous other industries by providing essential maritime transportation while driving economic growth to meet the rising demand for new ship constructions. This demand is propelled by global trade expansion, advance-

ments in shipping technology, and the pursuit of more efficient and environmentally friendly vessels (Smith et al., 2022). Notably, leading shipbuilding nations such as China, South Korea, and Japan depend heavily on this industry for economic stability and growth. China, for example, has expanded its shipbuilding capacity to become the world's largest shipbuilder, a growth fueled by government policies and significant investments in shipyard infrastructure (Ko and Shinoda, 2021). South Korea, renowned for its advanced shipbuilding technology, maintains a competitive edge by specializing in high-value vessels such as LNG carriers and offshore platforms (Hong et al., 2024). Japan, with its rich maritime history, continues to lead in eco-friendly ship designs and the automation of ship production processes (Japan Ship Technology Research Association, 2020; Ko and Shinoda, 2021). Given the critical role of ships as the primary mode of transportation across waterways, their significance cannot be overstated. Ships are essential for transporting goods and people worldwide, supporting international trade and significantly influencing the global economy (Nor and Nazery, 2008). Consequently, the production of these vessels requires careful attention to ensure they meet stringent standards of efficiency, safety, and environmental sustainability. This entails the adoption of advanced technologies, the implementation of rigorous safety protocols, and strict compliance with international

Article Highlights

- Nesting algorithms are categorized into exact, heuristic, metaheuristic, and hybrid, each offering unique solutions for material efficiency and waste reduction in shipbuilding.
- Advanced cutting technologies like oxy-fuel, plasma cutting, and algorithms such as GA, SA, and PSO enhance accuracy and production efficiency.
- The study employed PRISMA methodology, analyzing 35 recent studies (2018–2024) to identify best practices in nesting algorithms.
- Focused on material and cost efficiency, this study encourages integrating AI and machine learning to improve precision and scalability.

✉ Gunawan
gunawan_kapal@eng.ui.ac.id

¹ Department of Mechanical Engineering, Universitas Indonesia, Kampus UI, Depok 16424, Indonesia

environmental regulations (Alblas and Pruijn, 2024). The importance of the shipbuilding industry is further highlighted by its substantial impact on global trade and economic development (Baso et al., 2020; Sutrisno et al., 2024). The industry supports an extensive network of ancillary sectors, creates employment opportunities, and fosters innovation in maritime transportation. Its ability to adapt to changing market demands and technological advancements is crucial for maintaining and enhancing the efficiency and sustainability of global maritime operations.

In ship production, key activities related to plate cutting include preparation procedures and the steel plate cutting process, collectively referred to as the fabrication process (Mustafa et al., 2022). These tasks involve a series of precise steps and the use of advanced cutting technologies, such as oxy-fuel and plasma cutting, which demand high accuracy (Saiyara, 2024). Oxy-fuel cutting, preferred for its ability to handle thick steel plates, employs the combustion of oxygen and fuel gases to cut through steel. In contrast, plasma cutting uses an accelerated jet of hot plasma to slice through metal, providing greater precision and speed, particularly for thinner plates. The importance of these cutting processes cannot be overstated, as they are fundamental to preparing and arranging plate parts for smooth assembly while preserving the ship's structural integrity. Errors during this stage can lead to significant issues during assembly, potentially compromising the vessel's quality and safety (Okubo and Mitsuyuki, 2022). Given the critical role of plate cutting, selecting the appropriate technologies and processes is essential for optimizing steel plate utilization and reducing waste. This optimization process, known as nesting, involves strategically arranging cutting patterns on steel sheets to maximize material use and minimize waste. It requires careful consideration of factors such as part shapes, sheet dimensions, and design specifications. Advanced nesting software and algorithms are instrumental in generating optimal cutting patterns, enabling effective and efficient operations (Son et al., 2020). As the shipbuilding industry evolves, ongoing advancements and refinements in nesting techniques continue to improve the precision and efficiency of plate cutting, highlighting their critical role in enhancing the quality, efficiency, and sustainability of modern shipbuilding practices. The fabrication process and plate cutting optimization rely on specific design variables, which guide the selection of appropriate nesting methods. Nesting algorithms are broadly categorized into exact, heuristic, metaheuristic, and hybrid types, each providing effective solutions for optimizing nesting in shipbuilding. A thorough understanding of the characteristics and applications of each algorithm is crucial to achieving the best possible outcomes in nesting optimization.

Although nesting is critical in the shipbuilding industry, there is a notable lack of comprehensive studies that systematically evaluate and compare the various algorithms

used for nesting optimization. Existing literature often emphasizes individual algorithms without offering a holistic perspective on their comparative effectiveness across diverse shipbuilding contexts. This research addresses this gap by conducting a systematic review of prominent nesting algorithms, focusing on their application within the shipbuilding industry. The primary objective is to thoroughly evaluate these algorithms to identify the most effective methods for optimizing material utilization and minimizing waste in ship production. This study is particularly timely, given the growing pressures on the shipbuilding industry to improve efficiency, reduce costs, and adhere to increasingly stringent environmental regulations. This research aims to investigate the current state of nesting algorithms to identify the most efficient solutions available today. Additionally, it seeks to compare these algorithms based on material utilization, computational efficiency, and their applicability to various shipbuilding contexts. Understanding these factors is essential for shipbuilders, as it will enable them to select the most suitable algorithms for their specific operational needs, ensuring more sustainable and cost-effective production processes. Furthermore, the research explores the criteria for evaluating each algorithm's suitability, focusing on key factors that influence the performance of these algorithms. By examining these factors, the study aims to identify areas where existing algorithms can be improved to meet the evolving demands of the shipbuilding industry. This exploration is crucial not only for advancing current nesting technologies but also for guiding future research and development efforts in this field. Ultimately, this paper seeks to provide valuable insights that will help shipbuilders achieve greater efficiency and sustainability in their production processes, thereby strengthening the industry's ability to adapt to global market demands and technological advancements.

2 Fabrication process and plate cutting optimization

The fabrication process is a crucial phase in shipbuilding. It follows the design stage and includes the basic design, technical project, and detailed project of the intended product (ILO, 2023). This phase is critical as it transitions into the shipbuilding stage, where the actual construction of the ship's structure occurs. The shipbuilding phase is notably complex, involving several intricate processes, such as cutting steel plates and profiles, constructing panels, assembling modular cross-sectional blocks, building superblocks, and, ultimately, assembling the ship hull (Okumoto et al., 2009; Mustafa et al., 2022). Each of these steps requires precise coordination and meticulous execution to ensure the integrity and performance of the final vessel. The success of the shipbuilding phase heavily depends on

the efficiency and accuracy of the plate cutting process, which begins with cutting steel according to the design specifications (Basuki et al., 2012; Firmansyah et al., 2021). The cutting process is conducted using advanced machinery, such as cutting machines, lasers, or plasma cutters, which shape the steel plates to the required dimensions and forms for the ship's construction. The precision of these cuts is critical, as any inaccuracies can result in significant issues during subsequent stages of assembly. Additionally, the plate cutting process plays a key role in the overall efficiency of shipbuilding and material utilization (Rigo and Caprace, 2011; Perez-Martinez and Fernandez, 2023). Effective plate cutting minimizes waste and optimizes material use, making it both cost-effective and environmentally friendly. The quality of the cut plates directly influences the ease of assembly and the structural integrity of the ship. As a result, continuous advancements and innovations in cutting technologies and techniques are pursued to improve accuracy, reduce waste, and enhance overall efficiency in shipbuilding.

Plate cutting with nesting is a technique used to arrange and sequence cutting patterns from larger to smaller shapes or vice versa, with the goal of minimizing waste and optimizing material utilization, fabrication, and production (Oliveira and Gordo, 2018). This method strategically places smaller shapes within a larger material sheet to minimize unused material, thereby enhancing efficiency and reducing costs. The nesting process is often optimized using various approaches or algorithms, where the algorithm's flow is determined by design variables, constraints, and an objective function. These algorithms can range from simple heuristic methods to more complex metaheuristic techniques, such as genetic algorithms (GAs), simulated annealing (SA), or particle swarm optimization (PSO). The effectiveness of the nesting process largely depends on the algorithm's ability to manage the complexities of the material layout, taking into account factors such as the geometric shapes of the parts, material properties, and cutting constraints. Additionally, the objective function in these algorithms typically aims to maximize material utilization or the material utilization ratio while adhering to specific fabrication requirements and ensuring the quality of the cut parts. The material utilization rate, a key performance metric in this context (Xu and Yang, 2022), is defined by Equation (1).

$$\text{Material Utilization Ratio} = \frac{A_{fp} + A_{rm}}{\sum A} \quad (1)$$

where A_{fp} is the area of finished products, A_{rm} is the area of residual material, and $\sum A$ is the total area of raw material used. This equation measures material usage efficiency, indicating the proportion of raw material effectively used for the finished products and residual materials. A higher material utilization ratio signifies better optimization and less waste. This comprehensive approach to nesting improves

material efficiency and supports sustainable manufacturing practices by minimizing waste and reducing energy consumption during the cutting process. Similarly, the design variables for optimizing nesting in plate cutting include parameters with predefined constraints, such as plate dimensions and cutting patterns. The goal is to maximize nesting efficiency, which is quantified by the total yield rate, as shown in Equation (2), as proposed by Hamada et al. (2019).

$$\text{Total yield rate (\%)} = \frac{\sum_{x=1}^N AP_x}{\sum_{y=1}^M AM_y} \times 100 \quad (2)$$

Here, AP_x represents the total area of the parts (P_x), while AM_y denotes the area of the base material (M_y). N refers to the number of ship parts, and M indicates the number of base materials used. The yield rate measures nesting efficiency by minimizing the area of base material used during the fabrication process.

Nesting optimization design variables are closely related to the packing problem, which can be classified into two types: strip packing and bin packing. Strip packing involves arranging various strip patterns or shapes of different lengths onto a plane or base material with a fixed width (Martello et al., 2003), as illustrated in Figure 1. The objective of this process is to maximize area utilization by minimizing unused space. The patterns or shapes are strategically positioned to reduce the number of strips required and improve material efficiency (Liao et al., 2016). This is particularly important in the manufacturing industry, where raw materials are often costly, and efficient material usage can result in significant cost savings. Additionally, this method helps reduce material waste, supporting sustainability and eco-friendly practices.

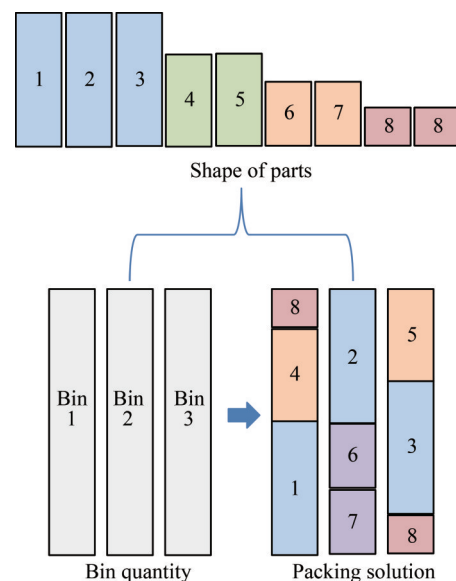


Figure 1 Strip packing

Conversely, the bin packing problem in nesting optimization focuses on fitting parts of varying sizes and space requirements into a limited number of available base materials or bins (Han et al., 2007), as illustrated in Figure 2. The goal is to minimize the area of base material used and maximize space efficiency (Hartono et al., 2022; González-San-Martín et al., 2023). In this context, parts must be arranged in a way that optimizes base material usage, minimizes wasted space, and maximizes the capacity of the available bins. This approach is essential in logistics and storage, where efficient space utilization can lower operational costs and improve productivity. Overall, both strip packing and bin packing play crucial roles in nesting optimization, helping industries enhance material and space efficiency, reduce costs, and promote more sustainable manufacturing practices.

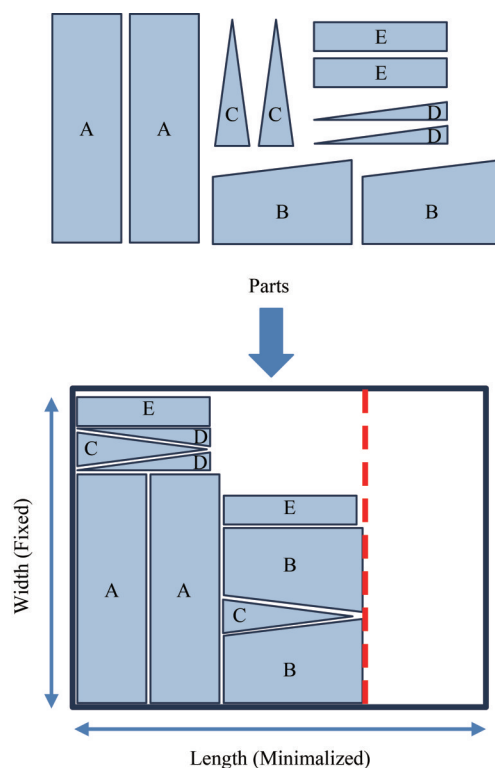


Figure 2 Bin packing

3 Classification of nesting algorithms

Nesting algorithms are a crucial component of nesting software, particularly for strip packing and bin packing design variables. They play a pivotal role in optimizing the arrangement of parts within a specified area, such as a sheet of metal or other base material designated for cutting. The primary goal of nesting algorithms is to minimize waste, maximize space utilization, and reduce production time and costs (Goodman et al., 1994). Implementing these

algorithms offers several advantages, including improved part placement, reduced waste through efficient component measurement, better material utilization, and minimized scrap (Lodi, 1999; Lodi et al., 2002; Martínez-Sykora et al., 2017).

Canellidis et al. (2013) demonstrated that strategically placing parts to maximize space utilization can significantly improve productivity and cost-effectiveness. Efficient arrangements show that nesting optimization can reduce material usage, lower production expenses, and enhance overall efficiency in the production process (Timmerman, 2013). Additionally, nesting algorithms streamline the component layout, saving time and effort by automating the optimal placement process, which further enhances efficiency. Properly organizing components contributes to higher productivity by enabling faster cutting and production processes.

Nesting algorithms also offer customization and flexibility to meet specific needs, accommodating different types of parts and constraints while adapting to various manufacturing scenarios. Overall, these algorithms help maximize material utilization, reduce waste, and enhance the overall yield from the base material. As illustrated in Figure 3, these algorithms are categorized into several types, each providing distinct approaches to achieve efficiency and effectiveness in the manufacturing process.

Figure 3 illustrates the classification of nesting algorithms into four main categories: exact algorithms, heuristic algorithms, metaheuristic algorithms, and hybrid algorithms. Exact algorithms, including linear programming (LP) and branch and bound, use mathematical approaches to find optimal solutions. Heuristic algorithms, such as first-fit, best-fit, and next-fit, apply practical approaches to achieve good results. Metaheuristic algorithms, such as GAs, SA, and PSO, are inspired by natural or social processes to identify optimal solutions. Hybrid algorithms combine various methods, such as GA+SA, GA+PSO, and ant colony optimization (ACO)+GA, to enhance algorithm performance. Further explanations of each category and algorithm are provided in the following subsections.

3.1 Exact algorithm

Exact algorithms guarantee the identification of the optimal solution by exploring all possible solutions (Jones, 2014). While these algorithms provide optimal solutions, they typically have high computational costs, particularly for large-scale problems. Two examples of exact algorithms are LP and branch and bound. LP involves the formulation of a problem as a set of linear equations and inequalities (Vielma et al., 2008; Huang et al., 2021). The objective is to optimize a linear function while satisfying the given linear constraints. LP models generally consist of decision variables, an objective function, and constraints. Common techniques for solving LP problems include the Simplex

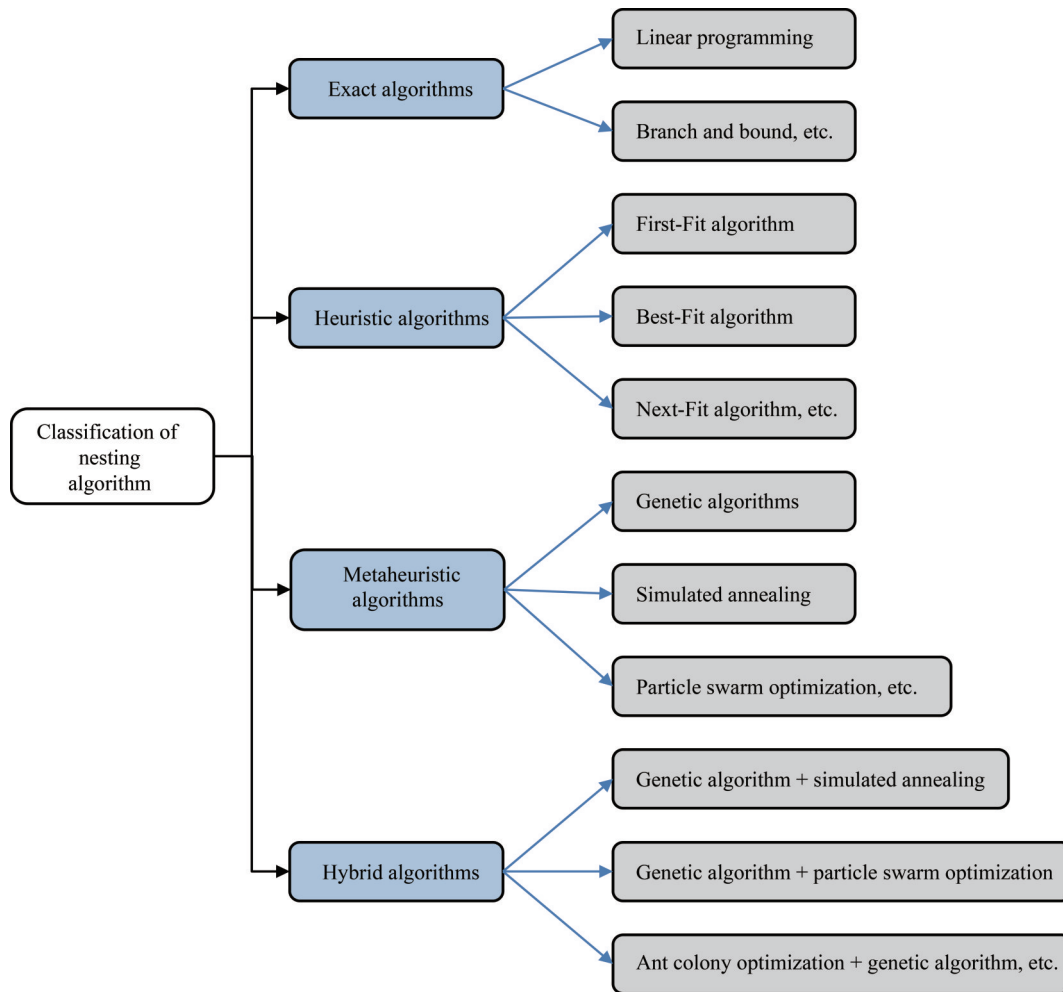


Figure 3 Nesting algorithm

method and Interior-Point methods. For example, in tackling the irregular strip packing problem, Cherri et al. (2016) developed mixed-integer linear programming models to handle piece non-overlapping constraints. They achieved this by either applying direct trigonometry to formulate the constraints or decomposing pieces into convex parts to simplify the geometric handling of non-overlapping constraints.

Branch and bound is a method used to find optimal solutions for combinatorial problems by breaking them into smaller subproblems and systematically evaluating potential solutions, as illustrated in Figure 4. The algorithm involves two key steps: “branching”, where the problem is divided into subproblems, and “bounding”, where upper and lower bounds are calculated for these subproblems. If the bound of a sub-problem indicates that it cannot yield a better solution than the best one found so far, it is discarded (Alvarez-Valdes et al., 2013). In their study, Alvarez-Valdes et al. developed a branch-and-bound algorithm for the irregular strip packing problem. This approach involved creating more efficient formulations and exploring various branching strategies, lower bounds, and variable-fixing procedures to improve computational performance.

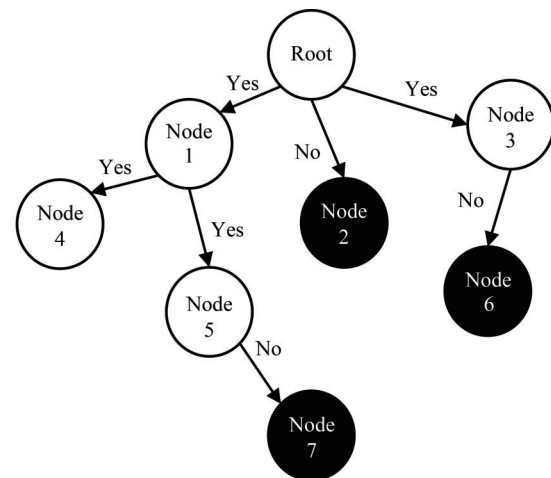


Figure 4 Branch-and-bound algorithm

3.2 Heuristic algorithm

Heuristic algorithms use practical rules or “heuristics” to find reasonably good solutions more quickly than exact

algorithms (Muriyatmoko et al., 2024). While these algorithms do not guarantee optimal solutions, they often provide satisfactory results within a reasonable time frame. Examples of heuristic algorithms include the first-fit, best-fit, and next-fit algorithms. The first-fit algorithm assigns an item to the first available bin that can accommodate it. As described by Yue (1991), the first-fit decreasing (FFD) algorithm first sorts items in descending order of size and then places each item into the first bin with sufficient remaining space. Although this method is simple and fast, it does not always yield the optimal solution. The effectiveness of the FFD algorithm has been extensively studied, with proven bounds on its performance in relation to the optimal number of bins required.

Pospelov et al. (2023) applied the best-fit algorithm for optimizing resource allocation, which begins with a sorted list of resources or bins and items to allocate. Each item is placed into the bin that will have the least leftover capacity after the placement, ensuring it fits. If no existing bin can accommodate the item, a new bin is created. While this approach is more computationally demanding than the first-fit algorithm, it generally leads to better resource utilization by minimizing gaps, making future item placements more efficient.

The next-fit algorithm, used in bin packing problems, starts with an empty bin and a counter for the number of bins, initially set to 1 (Wang et al., 2023). The algorithm iterates through the list of items, placing each item in the current bin if it fits. If an item does not fit into the current bin, that bin is closed, a new bin is initiated, and the item is placed in it. This process continues until all items are packed, with the final bin count illustrated in Figure 5. For example, with items sized [4, 8, 1, 4, 2] and bins with a capacity of 10, start with Bin 1 and place Item 4 into Bin 1, making it [4]. Item 8 does not fit, so start Bin 2 and place Item 8, making it [8]. Add Item 1 to Bin 2, making it [8, 1]. Item 4 does not fit, so start Bin 3 and place Item 4, making it [4]. Add Item 2 to Bin 3, making it [4, 2]. Finally, add Item 1 to Bin 3, making it [4, 2, 1]. The result is three bins: Bin 1 = [4], Bin 2 = [8, 1], Bin 3 = [4, 2, 1].

3.3 Metaheuristic algorithm

Metaheuristic algorithms are advanced optimization techniques designed to find near-optimal solutions for a broad range of problems. These algorithms generally combine multiple heuristics and incorporate randomization to explore the solution space. Examples of metaheuristic algorithms include GAs, SA, and PSO.

GAs are based on the principles of biological evolution, utilizing processes such as selection, crossover, and mutation to evolve solutions over multiple generations. According to Lourenço et al. (2001), GAs begin with an initial population of potential solutions represented by chromosomes. These chromosomes are selected based on their fit-

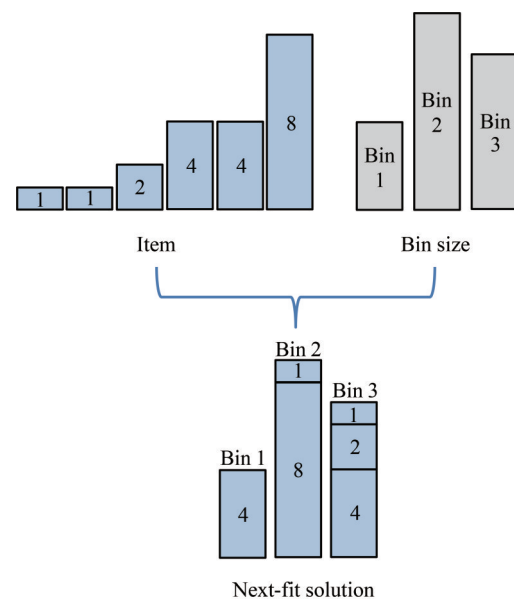


Figure 5 Next-fit algorithm

ness, undergo crossover to combine parts of two solutions, and are subject to mutation to introduce variability. This process continues until a stopping criterion is met, such as reaching a maximum number of generations or achieving an acceptable fitness level (Selow et al., 2007).

SA is an optimization method inspired by the cooling process in metallurgy, where solutions are explored randomly with a decreasing probability of accepting worse solutions over time. As explained by Ingber (1993), the SA algorithm starts with an initial solution and temperature. At each iteration, a new solution is generated by making a small random change to the current solution. If the new solution is better, it is accepted; if it is worse, it may still be accepted with a probability that decreases as the temperature lowers. The temperature gradually decreases according to a cooling schedule until it reaches a predefined minimum, at which point the algorithm stops (Gomes and Oliveira, 2006).

PSO mimics the social behavior of bird flocks or fish schools, where each “particle” in the swarm adjusts its position based on both individual and collective experiences. According to Kennedy and Eberhart (1995), PSO begins with a swarm of randomly initialized particles, each representing a potential solution. Each particle updates its position by considering its own previous best position and the best positions found by its neighbors, with the aim of converging toward the optimal solution identified by the swarm (Kareem et al., 2022). This iterative process continues until a stopping criterion, such as a maximum number of iterations or convergence to a stable solution, is reached.

3.4 Hybrid algorithm

Hybrid algorithms combine two or more algorithms from

different categories to leverage their respective strengths, aiming to improve solution quality and computational efficiency (Ting et al., 2015; Azevedo et al., 2024). These algorithms are particularly useful in various applications, as they can address the limitations of one algorithm by utilizing the strengths of another. Examples of hybrid algorithms include combinations such as GA with SA, GA with PSO, and ACO with GA, among others. The GA and SA combination capitalizes on GA's ability to explore global solutions and SA's strength in local solution exploitation (Elhaddad, 2012; Bettemir and Sonmez, 2015). Similarly, the GA+PSO hybrid enhances solution search by combining GA global exploration capabilities with PSO's ability to intensify the search in promising areas (Ru and Jianhua, 2008). ACO+GA combines ACO (Lee et al., 2008; Ashari et al., 2016) with GA, where ACO uses pheromone trails to guide the search, while GA generates solution variations, improving efficiency in complex optimization problems. Each of the aforementioned categories and algorithms has unique characteristics and applications. The selection of the most suitable algorithm depends on the specific nature and requirements of the problem at hand.

4 Development of nesting algorithms in shipbuilding

Nesting algorithms play a crucial role in shipbuilding by optimizing plate cutting, reducing waste, and improving material utilization. They efficiently arrange cutting patterns, enhancing the cutting process by strategically organizing patterns from large to small. These algorithms also enable the accurate determination of material needs, support supply management, and align with production schedules (Xie et al., 2007; Xu, 2016).

Various algorithms, including exact, heuristic, metaheuristic, and hybrid methods outlined in the previous section, have been developed to enhance nesting efficiency. These advanced techniques help reduce computation time and material waste while improving material utilization. Assessing their performance under different conditions is crucial to ensure smooth integration into production processes. By choosing the most effective nesting solutions, shipbuilders can optimize material usage, cut costs, and streamline production timelines, thereby promoting sustainability and minimizing waste through optimal nesting strategies.

Nesting optimization in shipbuilding is a critical process that greatly influences material utilization, production efficiency, and overall cost-effectiveness (Egeblad et al., 2007). A comprehensive and precise consideration of nesting solutions ensures the optimal use of materials, reducing waste and minimizing production costs. In shipbuilding, where materials such as steel plates are extensively used (Suzuki et al., 2004; Yu-ichi, 2007; Uemori et al., 2012),

efficient nesting results in significant savings and improved production efficiency. Additionally, it plays a crucial role in achieving accurate cuts and fits, which are essential for the structural integrity and performance of the ship.

Nesting algorithms have been classified into four categories in the previous section: exact algorithms, heuristic algorithms, metaheuristic algorithms, and hybrid algorithms. Each of these approaches has its strengths and weaknesses, and understanding them is essential for selecting the most suitable method for a specific application. Exact algorithms provide precise solutions by exploring all possible configurations, though they are often computationally intensive and may not be practical for large-scale problems due to lengthy processing times (Hifi, 2001). Heuristic algorithms offer approximate solutions by employing rules of thumb or strategies that reduce the search space. While faster than exact algorithms, they may not always find the best possible solution (Hu et al., 2015). Metaheuristic algorithms are advanced techniques that enhance the effectiveness of heuristic algorithms in exploring the solution space (Lee et al., 2023). These include methods such as GAs, SA, PSO, and others. Metaheuristic algorithms balance solution quality and computation time, making them well-suited for complex nesting problems. Hybrid algorithms integrate elements from various algorithmic approaches to leverage the strengths of each method while minimizing their weaknesses (Fujita et al., 1993; Wu et al., 2003; Xu et al., 2017). For example, a hybrid algorithm might use a heuristic method to quickly find a good solution and then refine it using an exact algorithm.

To aid in the categorization of various identification criteria for optimizing nesting algorithms in shipbuilding, Sari et al. (2024) identified 10 essential criteria for evaluating these algorithms. Each criterion plays a crucial role in determining the effectiveness and practicality of the nesting solution, as illustrated in Figure 6 and detailed in Table 1 below.

By integrating the criteria in Table 1 into the evaluation of nesting algorithms, shipbuilders can ensure that the selected solution meets both technical requirements and the practical constraints of the shipbuilding industry. This comprehensive approach fosters more efficient production processes, better resource management, and ultimately, higher-quality, cost-effective shipbuilding operations. Additionally, a further review is conducted using a systematic review method, as illustrated in Figure 7. The systematic review method provides advantages such as comprehensive coverage of existing literature, rigorous evaluation of research quality, and structured synthesis of findings (Gray, 2019; Williams et al., 2021), offering a robust foundation for making informed decisions about nesting algorithms.

The research methodology for this study began with the formulation of a clear research plan that outlined the objectives and scope of the literature review and analysis. The

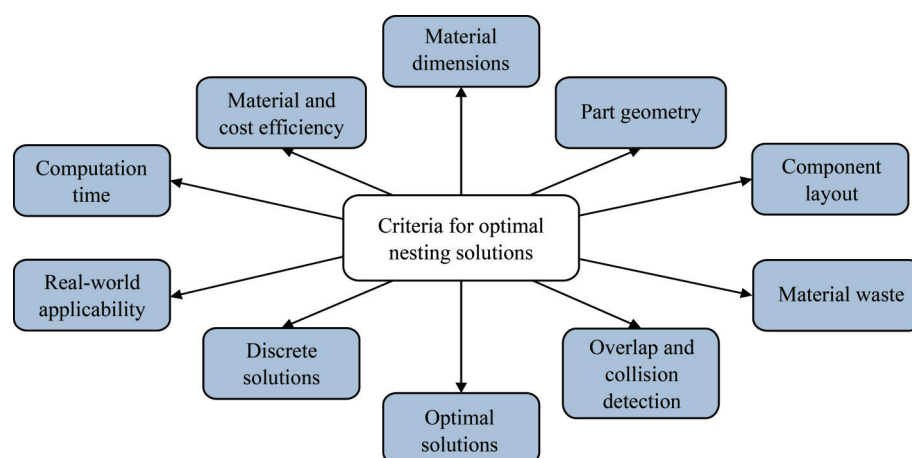


Figure 6 Criteria for optimal nesting solutions in shipbuilding

Table 1 Explanation of criteria for optimal nesting solutions

No.	Consideration criteria	Explanation
1	Material dimensions	The cutting pattern is heavily influenced by material dimensions, such as length, width, and thickness, to minimize waste.
2	Part geometry	Considering part shapes is crucial after determining the base material size, particularly for components with irregular or complex geometries.
3	Component layout	Efficiently arranging parts within a material sheet to minimize gaps and maximize material utilization.
4	Material waste	Optimizing part placement to minimize material waste during the cutting process.
5	Overlap and collision detection	The ability to detect overlaps and collisions to prevent material waste and machine damage.
6	Optimal solutions	Using integer programming techniques to generate optimal and practical nesting solutions for shipbuilding processes.
7	Discrete solutions	Ensuring that the solution is feasible and can be directly implemented in manufacturing or shipbuilding production.
8	Real-world applicability	The algorithm should be practical and applicable to real-world scenarios, taking into account ease of implementation and compatibility with existing manufacturing processes.
9	Computation time	The algorithm should deliver results within a reasonable time frame, accounting for the tight schedules in shipbuilding projects.
10	Material and cost efficiency	The algorithm should optimize material utilization and minimize production costs while maintaining quality, thereby enhancing cost-effectiveness in production.

literature search was conducted using several academic databases, including IEEE Xplore, ScienceDirect, Springer-Link, and Google Scholar, with a focus on publications from 2018 to 2024 to ensure the inclusion of the most recent advancements in nesting algorithms. Relevant search terms were defined using Boolean operators (“and/or”) to combine keywords such as “nesting algorithms”, “shipbuilding”, “plate cutting optimization”, and “material utilization”. This comprehensive search strategy played a crucial role in identifying all potentially relevant studies. The selection process involved both automated and manual filtering stages. First, duplicate records were removed, followed by

the exclusion of articles that did not meet the predefined inclusion criteria. Studies were excluded if they did not focus on nesting algorithms relevant to shipbuilding, were published before 2018, or were not written in English. These criteria were strictly applied to ensure that only studies directly related to the research objectives were included in the final review. The parameters for data collection were defined, encompassing the types of algorithms reviewed, the context of their usage, key performance indicators evaluated, and specific criteria for assessing nesting algorithms. The quality of the collected data was carefully assessed to ensure its validity and reliability. The gathered information

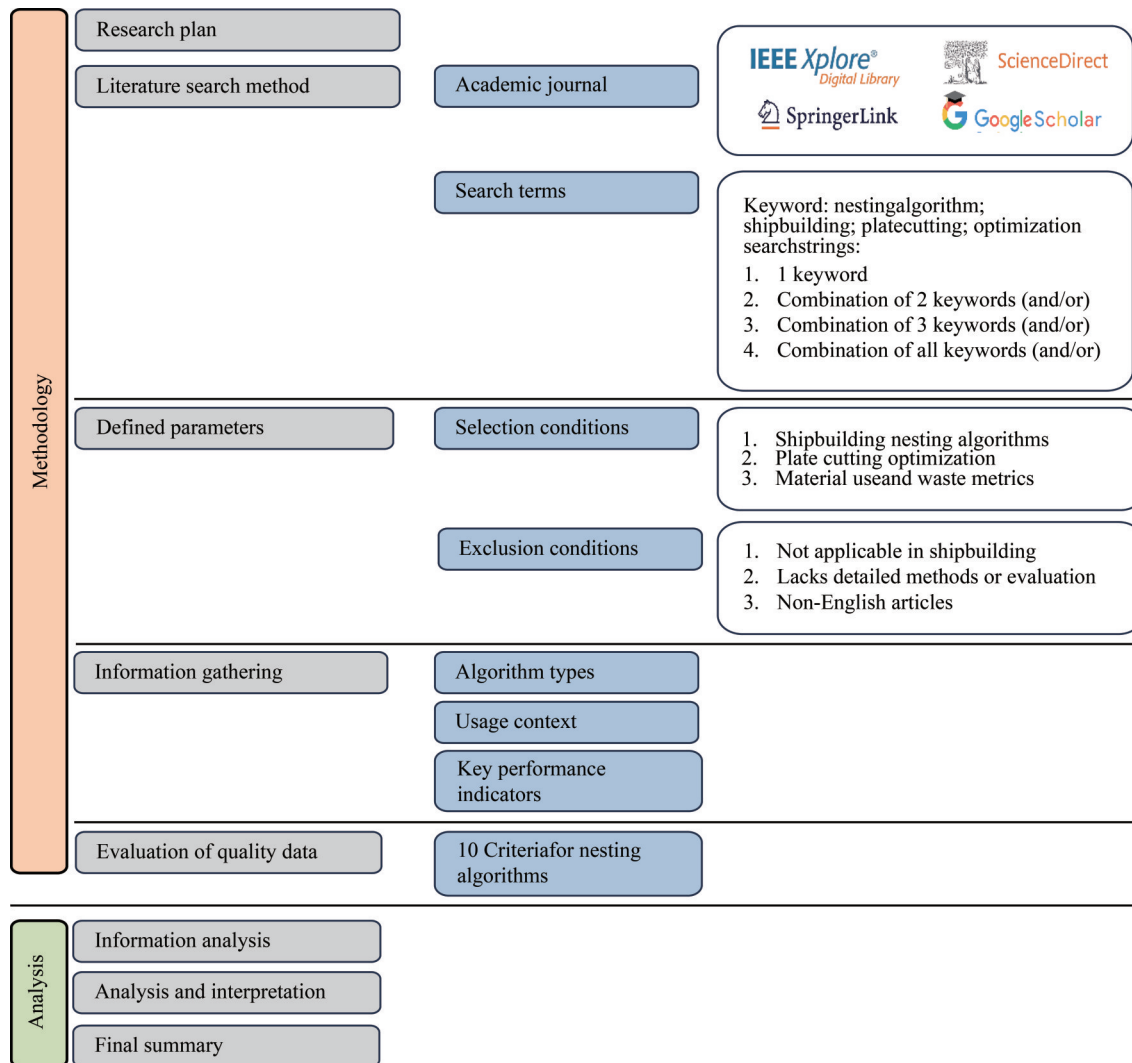


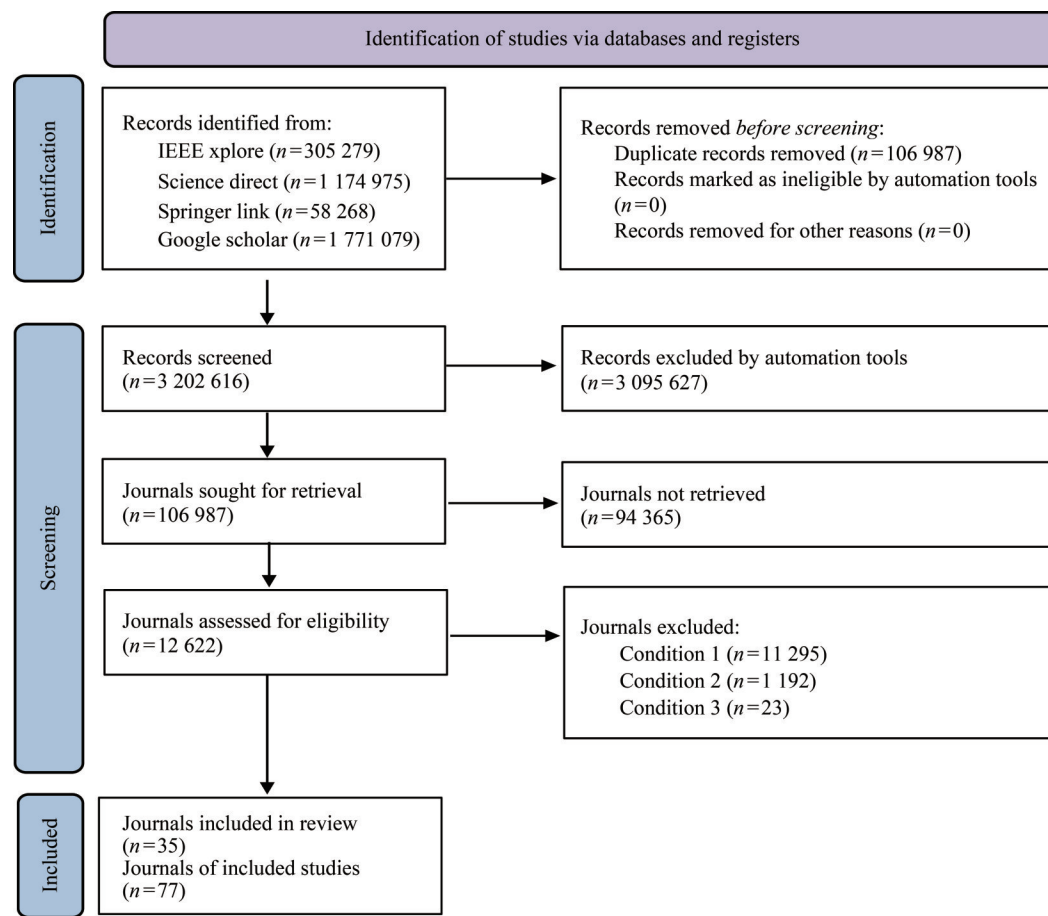
Figure 7 Methodology of the review

was then analyzed to identify trends and patterns, followed by the interpretation of the results to extract meaningful insights into the researched topic.

The study followed a structured and transparent review process in accordance with PRISMA guidelines (Page et al., 2020; 2021), with a flow diagram provided to overview the systematic review, as illustrated in Figure 8. The diagram details the stages of identification, screening, and inclusion, starting with 320, 261 records identified from the search and concluding with 35 studies included in the final review. This structured approach ensured that the research was based on a robust foundation, offering reliable and relevant insights into the effectiveness of various nesting algorithms in shipbuilding. Through a systematic review of the literature, the study identified the most effective algorithms for this context, evaluated their performance based on established criteria, and ensured that the solutions derived are applicable in real-world shipbuilding scenarios. To assess the effectiveness of various nesting methods, the study eval-

uated their alignment with 10 optimality criteria: material dimensions, part geometry, component layout, material waste reduction, overlap and collision detection capabilities, achievement of optimal solutions, real-world applicability, computation time efficiency, and cost-effective material usage. The results presented in Table 2 compare the different methods, providing valuable insights into their ability to enhance efficiency, minimize waste, and facilitate practical implementation within shipbuilding processes.

The analysis of Table 2 reveals various nesting methods employed in shipbuilding, focusing on their alignment with 10 optimality criteria. The method by Hamada et al. (2019), which is based on branch and bound, achieves 100% alignment, demonstrating strong efficiency in both strip and bin arrangements. Meanwhile, the approach by Calabrese et al. (2022), employing custom genetic nesting, achieves 90% alignment by combining elements of strip and bin packing. In contrast, the methods by Rakotonirainy and Vuuren (2020) and Liu and Chang (2023), which focus more on

**Figure 8** Flow diagram of the research study filtering process**Table 2** Comparison criteria of various nesting methods in shipbuilding

No.	Author	Optimization method	Design variable		Criteria for optimal nesting solution									
			Strip packing problem	Bin packing problem	1	2	3	4	5	6	7	8	9	10
1	Mundim et al. (2018)	Heuristic (H4NP)	✓	-	✓	✓	-	-	✓	✓	✓	✓	✓	✓
2	Wang et al. (2018)	Customized branch-and-bound	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	Cherri et al. (2018)	Mixed-integer quadratically constrained programming	-	✓	-	✓	✓	✓	✓	✓	✓	✓	-	✓
4	Gomez and Terashima (2018)	Hyper-heuristic	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	Grange et al. (2018)	Mixed-integer linear programming	-	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
6	Hamada et al. (2019)	Branch and bound	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	Djilali et al. (2019)	Jostle heuristic	-	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓
8	Kierkosz and Łuczak (2019)	One-pass heuristic	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9	Zhang et al. (2019)	Tabu search	✓	-	✓	✓	-	✓	✓	✓	-	✓	✓	✓
10	Fekete et al. (2019)	Split packing	-	✓	✓	-	-	-	✓	✓	-	✓	✓	-

Table 2 Comparison criteria of various nesting methods in shipbuilding (continuous)

No.	Author	Optimization method	Design variable		Criteria for optimal nesting solution									
			Strip packing problem	Bin packing problem	1	2	3	4	5	6	7	8	9	10
11	Arnaout et al. (2020)	Ant colony optimization (ACO)	-	√	-	-	√	√	√	√	√	√	√	-
12	Guo et al. (2020)	Geometric similarity future searching & fuzzy matching	-	√	√	√	√	√	√	√	-	√	√	√
13	Rakotonirainy and Vuuren (2020)	Hybrid simulated annealing	√	-	√	-	√	√	√	√	-	√	√	√
14	Martinez-Sykora et al. (2021)	Greedy and genetic	-	√	√	√	√	√	√	√	-	√	√	√
15	Li et al. (2021)	Hybrid adaptive genetic algorithm (HAGA)	-	√	√		√	√	√	√	-	√		√
16	Fang et al. (2021)	Sequence transfer-based particle swarm optimization (ST-PSO)	-	√	√	√	√	√	√	√	√	√	√	√
17	Umetani and Murakami (2022)	Coordinate Descent Heuristics	√	-	√	√	√	√	√	√	√	√	√	√
18	Xu and Yang (2022)	Improved genetic	√	-	-	-	√	√	√	√	√	√	√	√
19	Gunbeyaz et al. (2022)	Discrete event simulation (DES)	√	-	√	-	√	√	√	√	-	√	√	√
20	Calabrese et al. (2022)	Custom genetic nesting	√	√	√	-	√	√	√	√	√	√	-	-
21	Gardeyn and Wauters (2022)	Heuristic of combined ruin and recreate	-	√	√	√	√	√	√	√	√	√	√	√
22	Fernandez et al. (2022)	Voxel-based metaheuristic	√	-	√	√	√	√	√	√	√	√	√	√
23	Zhang et al. (2022)	Iteratively doubling local search	-	√	√	√	√	√	√	√	-	√	√	√
24	Fang et al. (2023b)	Deep reinforcement learning	-	√	-	√	√	√	√	√	-	√	-	√
25	Liu and Chang (2023)	Hybrid BonSA	√		-	-	-	-	-	-	-	√	√	√
26	Ko and Hsieh (2023)	Three heuristic	-	√	-	√	-	-	-	√	√	√	√	√
27	Fang et al. (2023a)	Hybrid reinforcement learning	-	√	√	√	√	√	√	√	-	√	-	√
28	Na and Yang (2023)	Deep neural network-based classification and clustering	√	-	√	√	-	-	-	√	-	√	√	√
29	Abdou et al. (2023)	Reinforcement learning and GNNs	-	√	√	√	√	√	√	√	-	√	√	√
30	Liu et al. (2023)	Segmented genetic and RL	-	√	√	√	√	√	√	√	-	√	√	√
31	Zhang et al. (2023)	Recursive dynamic programming	-	√	√	-	√	-	√	√	√	√	√	√
32	Wang et al. (2024)	Maximum residual rectangle genetic (MRGG)	-	√	√	-	√	√	√	√	-	√	√	√
33	Lallier et al. (2024)	Convolutional neural network (CNN) and graph neural network (GNN)	√	-	√	√	√	√	√	√	√	√	√	√

strip packing with hybrid SA and hybrid BonSA, respectively, achieve 70% and 60% alignment. Bin packing methods generally show higher alignment (80%–100%), while hybrid methods, such as those used by Hamada and Calabrese, demonstrate potential by combining the strengths of both approaches. This analysis offers a thorough evaluation of nesting methods, highlighting their effectiveness in

reducing waste and improving efficiency in shipbuilding processes. A deeper examination of each research algorithm was then conducted, focusing on its efficiency, iteration rate, and final nesting optimization results, as detailed in Table 3.

Table 3 presents the efficiency of various nesting methods employed in the shipbuilding industry. It covers key

Table 3 Efficiency of various nesting methods in shipbuilding

Number of Ref.	Explanation of method	Iteration time and rate	Optimization results
1	Iterative algorithm with different placement rules for 2D nesting within limited-size containers.	1 000 iterations for maximization problems and 10 iterations for minimization problems.	Best solutions for limited and unconstrained placement problems with an iteration rate of 1 000 (for some instances).
2	Handles irregular shape nesting using reverse convex quadratic constraints.	Approximately 287 seconds for four-polygon instances and 17058 seconds for five-polygon instances.	Solved nesting problems with up to five polygons, reducing gaps to 0.2%.
3	Addresses the irregular strip packing problem by permitting continuous rotations.	Some instances reach the time limit, such as the “Blaze2” instance, which took 15 120 seconds.	An improvement was observed in the “Shapes4” instance, with the length reduced from 23.02 to 19.92.
4	Integrates NSGA-II, SPEA2, and GDE3 to address bi-objective 2D bin packing problems.	An iteration rate of 1 000 iterations per trial.	Improved the utilization ratio by 12% and reduced the number of bins used by 15%.
5	Examines algorithms for bin packing with overlapping items, including the best fusion and grouping genetic algorithm.	Execution times: greedy < 0.1 seconds, overload-and-remove: 1 second, standard GA: 7 seconds.	The best fusion and grouping GA provides optimal or near-optimal solutions with an accuracy of 85%.
6	Integrates bin-packing and strip-packing problems using a branch-and-bound approach to achieve optimal nesting solutions.	GA: 23 hours, rule-based: 11 min, manual: 2 hours	The system improved yield rates by up to 5% compared with manual methods and saved time (GA: 64.67%, rule-based: 66.78%, manual: 65.91%), particularly for complex parts.
7	Subdivides irregular shapes into regular sub-shapes and then applies a heuristic for optimal placement.	50 iterations.	Increased material utilization by 25% and reduced production costs by 10%.
8	Solves nesting problems using the no-fit polygon concept combined with various fitting functions.	Iteration rate of up to 48 iterations in 14.48 seconds.	Achieved a filling rate of up to 94% in single knapsack problems, highlighting its efficiency in material utilization.
9	Optimizes the cutting sequence as both an open and constrained traveling salesman problem.	Maximum iterations: 100.	Efficiency increased by 21.62%, while cutter lifting times were reduced to 26.
10	Recursively splits a set of circles to achieve optimal worst-case packing density.	Total worst-case runtime: $O(n^2)$.	Visual examples of packing results in various container shapes, achieving a utilization rate of 53.90%.
11	Solves multi-level warehouse layout problems using the ACO algorithm.	Number of iterations: 9580.	ACO outperforms genetic algorithms and exact methods in large-size problem instances.
12	Utilizes Freeman chain code for 2D free-form shape layout, incorporating fuzzy matching.	Computation time overhead: 102.84 seconds.	Optimized layout with a utilization rate of 70.86%.
13	Integrates simulated annealing with a heuristic construction algorithm for strip packing.	Limited to 60 seconds for large instances and 5000 iterations for smaller ones.	Reduced packing height by an average of 1% compared with the original IA algorithm.

Table 3 Efficiency of various nesting methods in shipbuilding (continuous)

Number of Ref.	Explanation of method	Iteration time and rate	Optimization results
14	The greedy method generates a placement sequence, while the genetic algorithm optimizes the sequence.	Various computation times range from 0.343 to 4.303 seconds across different case studies, with 30 generations per iteration.	Reduced computation time by 30% and improved material utilization by 20%.
15	Solves two-dimensional rectangular packing problems.	The average calculation time for the J2 instance was 23.3 seconds.	Achieved full packing layouts with a utilization rate of 99.92%.
16	Uses piece-matching and sequencing strategies for irregular packing.	The algorithm runs 20 times per case.	The algorithm demonstrates higher efficiency in both space utilization (80%–83% vs. 74%–78%) and computation time (20–25 s vs. 25–35 s) compared with the traditional algorithm across various samples.
17	Uses corner detection to reduce search space in raster models.	Approximately 1.26×10^5 iterations on average for high-resolution instances.	Achieved a 15% reduction in container length, maximizing layout density while maintaining reasonable computation time.
18	Applies real number encoding for steel plate cutting optimization.	Convergence occurs after ~100 generations, with further improvement after ~20 000 generations.	Detailed nesting results show a utilization rate of 92.73%.
19	Models the secondary cutting zone of a ship recycling yard and compares various cutting technologies.	Multiple simulation runs for 10 repetitions.	Increased productivity by 60% and reduced operational costs by 40%.
20	Estimates the number of printable components on a given printer plan without iterative processes.	Non-iterative algorithm.	Achieved a 30% increase in machine volume utilization, with cost and time savings of 25% and 20%, respectively.
21	Integrates ruin and recreate with a goal-driven method for 2D variable-sized bin packing with guillotine constraints.	Computation time is 600 seconds, using 8 threads for the experiments.	In benchmark tests, it outperforms state-of-the-art algorithms in solution quality by up to 15% compared with existing methods.
22	Uses voxel-based representation with ILP formulation and metaheuristics for 3D irregular packing.	Execution time ranges from 2 to 3 hours.	Achieved up to 90% efficiency in solving three-dimensional irregular packing problems within practical computation times.
23	Uses a waste least first decreasing strategy, bottom-left method, and random local search for irregular bin packing.	The iteratively doubling strategy avoids excessive time on a single bin placement ($1000/n$).	Achieved a 20% reduction in bin usage and a 15% improvement in bin utilization efficiency.
24	Employs pointer networks and model-free reinforcement learning for 1D cutting stock problems.	Computation time is less than 1 second.	Detailed cutting stock schemes with a utilization rate of 94.56% (Instance S1).
25	Combines genetic algorithm and linear programming for 2D irregular packing.	A maximum of 30 generations is used.	Increased utilization rates by up to 5.89%.
26	Proposes three heuristic algorithms for stacking irregularly shaped stone pieces.	Cycle times: Algorithm 1: 478.52 seconds, Algorithm 2: 488.12 seconds, Algorithm 3: 483.12 seconds.	Detailed nesting results with utilization rate up to 41.28%.
27	Integrates Monte Carlo learning, Q-learning, and Sarsa-learning with the Bottom-Left positioning strategy.	Total number of episodes: 300.	Achieved the best utilization rates on various benchmarks, e.g., Dighe1 (87.31%).
28	Uses deep neural networks for classification and pairwise clustering to optimize the nesting of ship parts.	Pairing times: Sheet 1: 0.40 seconds, Sheet 2: 0.51 seconds, Sheet 3: 0.82 seconds, Sheet 4: 0.63 seconds, Sheet 5: 2.59 seconds.	Reduced scrap rate by 11.0%, pairing time by 44.1%, and arrangement time by 47.5%.

Table 3 Efficiency of various nesting methods in shipbuilding (continuous)

Number of Ref.	Explanation of method	Iteration time and rate	Optimization results
29	Applies GNNs to estimate geometrical compatibility indices for 2D and 3D nesting problems.	Time reduction ranges from 30% to 48% compared with traditional methods.	Improved material utilization by up to 18%.
30	Optimizes CNC nesting laser cutting paths for ship hull components.	Maximum iterations: 1000, run independently 30 times per test.	Improved material utilization rates by up to 5.89%, averaging 4.02% better than previous methods.
31	Addresses the constrained 2D guillotine cutting problem with defects using normal and raster points.	Performance is evaluated with a time limit of 150 seconds per instance.	Achieved a 95% increase in cutting efficiency compared with previous methods.
32	Combines heuristic and intelligent optimization techniques for ship nesting.	Generates up to a maximum of 200 iterations.	Improved material utilization in ship nesting from 91.5% to 97.5%.
33	Combines CNNs and GNNs to estimate 2D nesting efficiency.	Consistent 5-minute processing time for all tasks in the dataset.	Achieved with a standard deviation of 10.9% and a utilization rate of 79.20%.

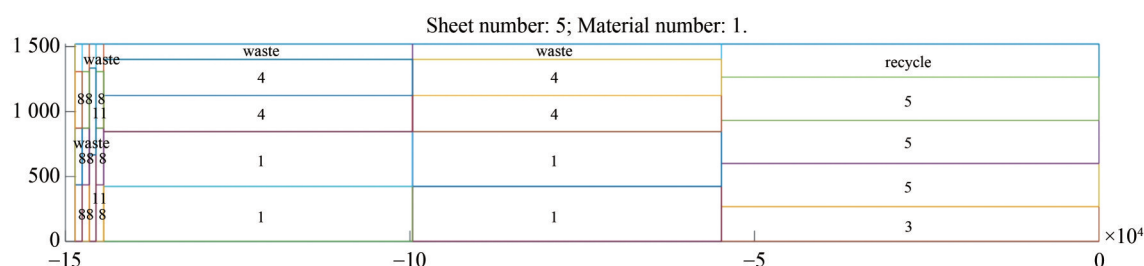
aspects such as iteration time, iteration rate, and optimization results achieved by each method. The columns in the table include the reference number of the method, a brief description of the nesting approach, the iteration time and rate used, and the optimization outcomes, such as improvements in efficiency, material utilization rates, and reductions in cost or time. The methods listed differ in their approaches to solving nesting problems, including the use of iterative algorithms, reverse convex quadratic constraints, and the integration of multiple evolutionary algorithms. Each method is evaluated based on its effectiveness in enhancing material utilization and reducing production time and costs. This analysis builds upon the 10 evaluation criteria discussed in Table 2, with methods selected based on the highest compatibility with each design variable.

To identify the two best algorithms in each packing category (strip packing, bin packing, and hybrid packing), the data presented in Table 3 were analyzed. This analysis highlights that the research by Xu and Yang (2022) and Wang et al. (2024) are the top choices for strip packing. Xu and Yang (2022) utilize an improved GA with real number encoding, achieving an impressive material utilization rate of approximately 92.73%, as illustrated in Figure 9. Convergence occurs after around 100 generations, with fur-

ther improvements in the solution observed after approximately 20000 generations. Wang et al. (2024) use the maximum residual rectangle genetic (MRRG) algorithm, which reaches a material utilization rate of 97.5%, showing a 4%–5% improvement over the previous maximum residual rectangle (MRR) algorithm, as illustrated in Figures 10 and 11. This indicates a significant increase in material utilization efficiency.

In the bin packing category, the papers by Li et al. (2021) and (Hamada et al., 2019) stand out as the best. Li et al. (2021) introduce the hybrid adaptive genetic algorithm (HAGA), which demonstrates a high material utilization rate, reaching 100% in some instances, as shown in Table 4 and Figure 12. This algorithm performs 150 continuous iterations without significant changes, emphasizing its effectiveness in maximizing material usage. Hamada et al. (2019) utilize a branch-and-bound method with a rule-based approach that requires only 11 minutes of computation time. This method achieves a good optimization result compared with manual methods, with a utilization rate of 66.78%, as illustrated in Figure 13.

For hybrid packing, the top choices are Wang et al. (2024) and Hamada et al. (2019). Wang et al. (2024) with the MRRG algorithm, achieves an impressive material uti-

**Figure 9** Nesting result based on rectangular packing method (Xu and Yang, 2022)

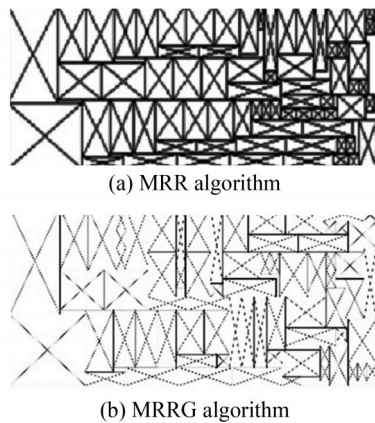


Figure 10 Nesting results (Wang et al., 2024)

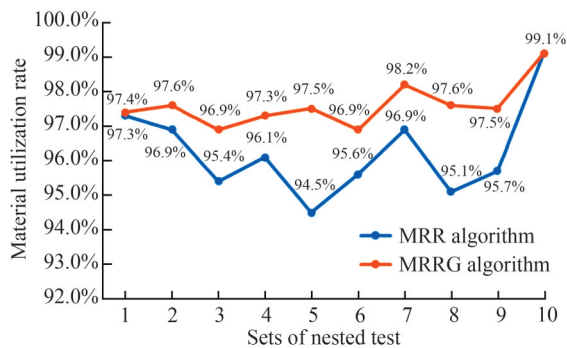


Figure 11 Material utilization comparison between MRR and MRRG algorithms (Wang et al., 2024)

lization rate of 97.5%. This algorithm combines the maximum residual strategy with GAs, leading to significant improvements in material efficiency. Hamada et al. (2019)

employ a brand-and-bound algorithm, with Figure 14 illustrating the nesting system applied to a ship's engine room. The system-generated plan achieved a 5% higher yield rate than the manually arranged plan, owing to more efficient part combinations using the same base material. This demonstrates the system's potential to improve efficiency and outcomes, particularly for complex shapes. Moreover, the system is significantly time-efficient, generating plans in 40 minutes compared with 2 hours by an expert engineer, demonstrating its superiority over manual arrangements.

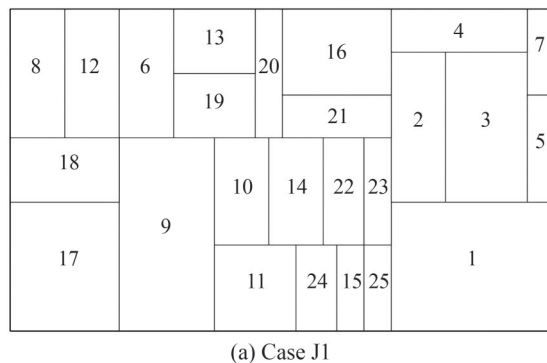
In conclusion, for strip packing, the studies by Xu and Yang (2022) and Wang et al. (2024) are the top performers, offering exceptional material utilization rates and efficient algorithms. For bin packing, the works by Li et al. (2021) and Hamada et al. (2019) provide efficient and rapid solutions. In hybrid packing, the combined methods of Wang et al. (2024) and Hamada et al. (2019) deliver excellent results in optimizing material utilization and operational efficiency in shipbuilding. This analysis emphasizes the importance of selecting the appropriate algorithm based on the packing type and specific industry requirements to achieve optimal material and operational efficiency.

5 Discussion

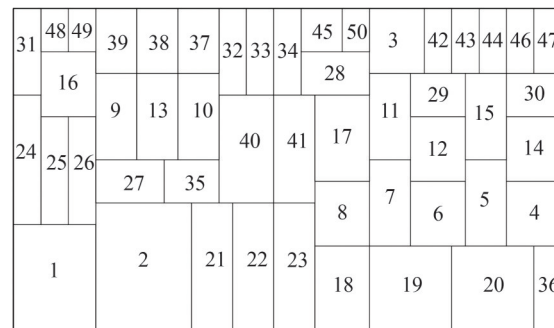
The analysis conducted using the systematic review methodology in the previous section identifies several algorithms highly suitable for nesting optimization criteria. However, based on these results, a continuous review can be performed to address deficiencies, potential future devel-

Table 4 Comparison of Efficiency and Speed: Filling Rate vs. Calculation Time for J Instances (Li et al., 2021)

Instances	N	Sheet	SGA		AGA		AGA		HAGA	
			F (%)	Time (s)	F (%)	Time (s)	F (%)	Time (s)	F (%)	Time (s)
J1	25	40×15	100	10.71	100	8.37	100	7.69	100	5.59
J2	50	40×15	99.65	27.18	99.875	24.53	99.90	25.11	99.917	23.3
Average			99.825	19.95	99.94	16.45	99.95	16.4	99.96	14.45



(a) Case J1



(b) Case J2

Figure 12 Nesting layout of instances (Li et al., 2021)

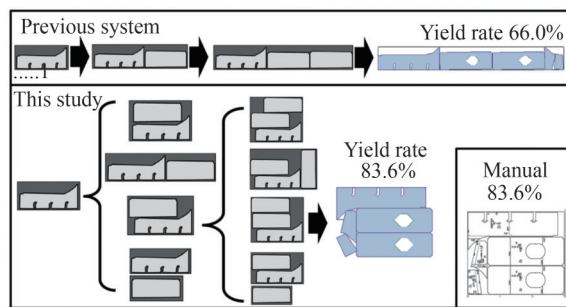


Figure 13 Effectiveness rate of branch-and-bound algorithm (Hamada et al., 2019)

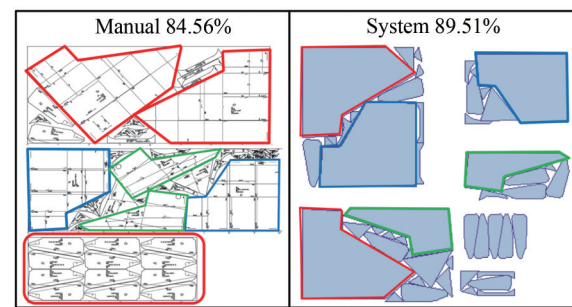


Figure 14 Comparison of nesting layout: manual arrangement by expert engineer versus nesting system (Hamada et al., 2019)

opments, and suggestions for improvement for each algorithm presented in the papers. A comparison of the disad-

vantages, potential developments, and future improvement suggestions for each paper is presented in Table 5.

Table 5 Comparison of disadvantages, potential developments, and suggestions for optimization methods

Number of Ref.	Disadvantages of algorithm	Potential developments	Suggestions for improving
1	Limited capability to handle heterogeneous pieces, requiring a high number of iterations to achieve better solutions, resulting in significant computational expense.	Develop automatic placement rule selection using hyper-heuristics and apply it to open-dimension problems with innovative placement strategies.	Implement automatic selection using hyper-heuristics, develop new placement rules, and enhance overall algorithm efficiency.
2	High iteration count with potential for uncovered polygon parts to penetrate boundaries.	Create tighter approximations using smaller circle libraries and implement a customized BB-based solver.	Expand the circle library and introduce more circles to improve overlap approximation and improve accuracy and performance.
3	Computationally intensive with imprecise piece representation, leading to inefficiencies.	Enhance piece representation, reduce computational time, and develop efficient algorithms tailored for non-convex pieces.	Apply symmetry-breaking constraints, develop stronger lower bounds, and implement tighter constraints to enhance computational efficiency.
4	Complex algorithms with scalability issues, high computational costs, and challenges in generalization to diverse scenarios.	Incorporate additional objectives, explore diverse features, and investigate alternative mechanisms for action determination.	Experiment with various MOEAs, explore innovative population creation methods, and research learning metrics to improve decision-making processes.
5	Unbounded approximation factor with high complexity, resulting in poor performance in worst-case scenarios for certain algorithms.	Conduct worst-case analysis, establish robust lower bounds, develop efficient cutting techniques, and design specialized heuristics.	Integrate decantation post-treatment, develop specialized algorithms, and combine heuristics with genetic algorithms to achieve superior performance.
6	GA-based approaches exhibit inefficiency; single-point search methods are suboptimal, and the BL method offers limited placement options.	Address combinatorial explosion challenges, improve packing methodologies, and maintain multiple intermediate proposals for better optimization.	Expand the search area, incorporate branch-and-bound methods, and improve one-point search strategies to retain multiple optimal plans.
7	Addressing the complexity of irregular shapes, ensuring non-overlapping constraints, precise confinement, and efficient pivoting strategies.	Develop nesting software incorporating the algorithm to cater to diverse industrial applications.	Apply placement optimization rules, evaluate established placement strategies, and develop models to effectively subdivide irregular shapes.
8	One-pass placement finalizes results but faces significant computation time challenges for very large instances.	Investigate alternative element sequences, design new fitting functions, and adapt the approach for non-rectangular shapes and one-dimensional fixed problems.	Incorporate parallel computation techniques, explore novel fitting functions, and design methods tailored for non-rectangular shapes.

Table 5 Comparison of disadvantages, potential developments, and suggestions for optimization methods (continuous)

Number of Ref.	Disadvantages of algorithm	Potential developments	Suggestions for improving
9	Limited integration of process constraints, with traditional algorithms proving ineffective in certain scenarios.	Fully integrate process constraints, including geometric and cutting condition considerations, for enhanced applicability.	Ensure efficient cutting sequences, enhance heuristic approaches, and investigate diverse search pathways for better outcomes.
10	Ineffective handling of acute triangles, challenges with specific container shapes, and potential failure of split packing for certain instances.	Extend the split packing approach to three-dimensional problems, create online implementations, and apply the method to various shapes to improve worst-case performance scenarios.	Develop strategies to handle acute triangles, refine greedy splitting techniques, introduce weighted generalization, and expand algorithms to accommodate new shape categories.
11	B&B approaches are time-intensive for large problems, often unable to solve all instances within acceptable timeframes, requiring the incorporation of heuristics.	Compare the algorithm performance with advanced methods such as particle swarm optimization and whale optimization algorithms.	Extend research using advanced algorithms and enhance performance by integrating sophisticated heuristics and optimization techniques.
12	Lacks sequential optimization; not suited for other 2D layout problems; increases computation time without enhancing layout results.	Expand the method to address a broader range of 2D layout problems; improve the longest common subsequence approach; enhance the handling of highly complex shapes.	Minimize time overhead by optimizing shape placement sequences and developing methods for handling complex geometries.
13	Performance depends on benchmark instances; prolonged execution due to greedy selection and local improvement strategies.	Improve search efficiency by integrating heuristic and metaheuristic techniques; optimize parameters and leverage machine learning to adapt to varying problem characteristics.	Replace randomized improvement with simulated annealing; omit the local improvement stage in favor of a multi-start strategy; adjust heuristic construction for denser packing.
14	Prioritizes speed over accuracy, leading to suboptimal placements and high computation times in certain cases.	Achieve a balance between accuracy and computation time by refining semi-discrete representations.	Integrate anti-aliasing effects, introduce minimum gap constraints, and refine semi-discrete representations for improved accuracy.
15	Requires managing multiple parameters, with potential for early evolution stagnation.	Investigate optimal parameter settings and mitigate sudden probability drops during evolutionary processes.	Standardize parameter settings, develop continuous evolution strategies, and balance the consideration of individual fitness and population dynamics.
16	Dependent on high-quality case samples; limited rotation angle flexibility; features a complex, time-intensive positioning algorithm.	Gather additional case studies and samples; incorporate deep learning and global transfer learning to enhance adaptability.	Leverage insights from prior tasks, incorporate both local and global information, and optimize positioning strategies based on evaluation metrics.
17	High computational cost for raster representations; line search becomes time-consuming with high-resolution rasters.	Lower computational costs by refining corner detection methods and narrowing the search space.	Implement fast local search techniques, guided local search with adaptive penalty weights, and double scanline representations for efficient solutions.
18	Reduced global search capability, making it less effective for small-scale problems.	Improve generalizability by applying the method to diverse cutting processes and accommodating different raw material dimensions.	Explore underutilized solution regions, enhance global search capabilities, and optimize strategies for cutting standard rectangular parts.
19	Lack of transparency, flexibility, and coverage in ship recycling tools.	Support decision-making, optimize productivity, consider CAPEX and OPEX, and conduct case studies for new technologies, such as plasma cutters.	Increase productivity through optimization, utilize discrete event simulation, cover all activities, and provide flexibility for process modifications.
20	Limited to the <i>XY</i> plane, not integrated into commercial systems, and does not account for qualitative aspects.	Integrate the <i>Z</i> -axis for 3D nesting, handle qualitative aspects, and improve integration with commercial 3D printing systems.	Introduce technological parameters, incorporate iterative processes, and develop flexible integration for various software environments.

Table 5 Comparison of disadvantages, potential developments, and suggestions for optimization methods (continuous)

Number of Ref.	Disadvantages of algorithm	Potential developments	Suggestions for improving
21	Local optima stagnation, with the radical ruin procedure sometimes being counterproductive, leads to diminishing returns with additional threads.	Explore the 2BP variant without the guillotine constraint, address real-world limitations, and apply a goal-driven approach to other problem types.	Implement sophisticated diversification, fine-tune parameters, and refine multithreaded implementations.
22	High computational effort is required, with solving the ILP model feasible only for small instances.	Investigate metaheuristic approaches and develop comprehensive benchmark datasets and instance generators.	Apply metaheuristic techniques, combine constructive algorithms with local search, and explore voxel representations for different resolutions.
23	Overlap minimization is time-consuming; initial position generation may be inefficient, and the warm-start strategy is not always effective.	Enhance overlap minimization, improve initial positioning, create adaptive warm-start strategies, and refine iterative strategies using ML or adaptive algorithms.	Incorporate advanced heuristics, use hybrid approaches, leverage parallel processing for larger instances, and implement dynamic adjustment mechanisms.
24	Poor universality, susceptibility to local optima traps, and high computational costs.	Develop machine learning and deep learning algorithms, utilizing reinforcement learning for sequential optimization.	Integrate diverse optimization techniques, utilize advanced learning models, and implement improved exploration mechanisms.
25	Serial cutting constraints limit the optimal solution, and TSP conversion affects cutting path optimization.	Allow free laser head movement and use an RL-based Segmented Genetic Algorithm (RLSGA).	Use part-cutting constraints, apply RLSGA for global solutions, and enhance laser head movement flexibility.
26	Inefficiency in handling dynamic and varied stone shapes, requiring continuous real-time decision-making.	Improve stability and utilization by developing advanced heuristic algorithms, enhancing real-time decision-making, and integrating sophisticated vision systems and feedback mechanisms.	Enhance vision system accuracy, improve the real-time feedback loop, develop algorithms for handling varied shapes, and incorporate machine learning for predicting the best stacking order.
27	Low efficiency and scalability, with a high skill requirement and inconsistent solution quality.	Develop efficient network solutions, optimize parameters and training sets, explore transfer mechanisms, and apply them to 2D nesting problems.	Construct efficient network solutions, optimize parameters, improve generalization, and enhance residual material utilization.
28	High complexity, long computational time, manual adjustments, and ineffective clustering.	Reduce computational time, handle diverse shapes, and incorporate sophisticated deep learning models with dynamic arrangement orders.	Utilize DL-based clustering and geometric data models, improve feature alignment, and expand the training dataset.
29	The framework did not use nesting environments during training, resulting in inefficiencies in time and material waste reduction for irregular parts.	Design realistic nesting environments for training and develop smart nesting algorithms for layout problems.	Design a realistic nesting environment and replace metaheuristics with a learning-based approach for the CVRP problem.
30	Insufficient resources, long solution times for large problems, heuristics trapped in local optima, and machine learning techniques dependent on the quality of the training set.	Integrate advanced optimization techniques such as metaheuristics, reinforcement learning, and deep learning, and explore applications of genetic algorithm-linear programming (GA-LP).	Integrate the LP phase within GA dynamically, explore advanced optimization techniques, and improve algorithm performance.
31	Inefficiency in reducing points and increased time required to find optimal solutions with defects.	Incorporate recent discrete methods, such as “meet in the middle”, to enhance upper bound values and improve efficiency.	Utilize recent discrete methods, explore techniques for refining upper bound values, and reduce computational times.
32	The use of a single heuristic strategy lacks universality, limiting the ability to find optimal solutions.	Convert irregular shapes into rectangles or triangles to optimize material utilization.	Apply multiple heuristic strategies, improve genetic algorithms, and enhance solution diversity.
33	Insufficient task information extraction and a low number of edges affect performance, with dependency on the time parameter.	Utilize edge embedding, integrate structural encoding innovations, and consider the use of a master node.	Enhance global performance through node and edge embeddings, structural encoding innovations, and increased variability in dataset generation.

To simplify the evaluation in Table 5, the majority of shortcomings of the proposed algorithm are categorized based on design variables: bin packing, strip packing, and hybrid packing. This categorization provides a structured approach to identifying and addressing common issues frequently encountered during algorithm development. By focusing on these recurring problems, future enhancements can avoid similar pitfalls, leading to more robust and efficient solutions. Additionally, the categorization helps identify clear patterns for future development by examining papers with well-defined and structured deficiencies. This method simplifies the evaluation process and ensures that the development is guided by a thorough understanding of past shortcomings. By adopting this approach, developers can prioritize improvements that address the most critical issues, ensuring that future algorithms are both innovative and practical. The ultimate goal is to establish a more efficient development cycle that builds on previous advancements, leading to increasingly sophisticated and effective algorithms.

Gunbeyaz et al. (2022) focused on improving productivity in ship recycling through discrete event simulation. However, their nesting optimization algorithm has several limitations, including a lack of transparency in the current tools, inadequate coverage of all activities within ship recycling yards, and limited flexibility in process modification. Future developments should aim to support decision-making for equipment usage, develop user-friendly frameworks to optimize yard productivity and consider both CAPEX and OPEX for new technologies. Additionally, more case studies are needed to identify risks associated with technologies such as plasma cutters. Improvements should focus on enhancing productivity through comprehensive optimization and flexible process adjustments.

Hamada et al. (2019) presented an automatic nesting system that employs the branch-and-bound method. However, the GA-based approach in this study suffers from inefficiency in computation time, a suboptimal one-point search method, and limited arrangement options with the BL method. Future improvements should address the combinatorial explosion in methods such as BL and GA, enhance the search for arrangement positions, and retain multiple intermediate proposals. The algorithm requires a broader search area for better arrangement plans and should apply the branch-and-bound method to decompose complex problems into more manageable subproblems.

Xu and Yang (2022) proposed a steel plate cutting optimization algorithm based on a genetic algorithm with real number coding. The main drawback of this algorithm is its reduced global search capability. Future research should focus on improving the model's generalizability and applicability to various cutting processes, as well as managing packing solutions for different raw material sizes. Enhancing the algorithm will involve exploring underutilized

regions of the solution space to improve the global search capability.

Fang et al. (2023a) developed a hybrid reinforcement learning algorithm to address 2D irregular packing problems. However, the algorithm has several shortcomings, including poor universality, susceptibility to local optimum traps, and high computational costs. Future development should focus on leveraging machine learning and deep learning algorithms for sequential packing optimization, utilizing reinforcement learning to model sequential decision problems as Markov decision processes. To enhance the algorithm, it is essential to integrate diverse optimization techniques, utilize advanced learning models such as deep reinforcement learning, and implement improved exploration mechanisms to avoid local optima.

Research by Calabrese et al. (2022) developed a nesting algorithm aimed at optimizing part placement in additive manufacturing. However, this algorithm operates only in the XY plane, limiting its applicability to three-dimensional nesting, and does not consider the qualitative aspects of printed components. Future development should integrate the Z-axis, address qualitative factors such as surface finish and structural integrity, and enhance compatibility with commercial 3D printing systems for advanced processes such as selective laser melting. Additionally, improvements should introduce technological parameters specific to additive manufacturing, incorporate iterative processes to ensure accuracy and develop flexible integration strategies for various software environments.

The optimization algorithms discussed can be categorized into strip packing, bin packing, and hybrid packing. Strip packing, as demonstrated by Hamada et al. (2019), aims to minimize material waste by arranging parts in a strip shape but faces challenges such as computational inefficiency and suboptimal placements. The aim of bin packing, explored by Xu and Yang (2022), is to maximize space utilization in containers, but the method struggles with maintaining the global nature of the search solution and managing small-scale problems. Hybrid packing, as highlighted by Fang et al. (2023a), combines elements of both approaches to address irregular packing problems but encounters issues such as local optimum traps and high computational costs.

Future development directions for packing algorithms should focus on addressing combinatorial explosions in strip packing by enhancing search methods and preserving intermediate proposals. Additionally, improving the generalizability of bin packing models for various cutting processes is essential, along with advancing machine learning techniques for hybrid packing. Suggested improvements include broadening the search area in strip packing, exploring less characterized solution spaces in bin packing, and integrating diverse optimization methods in hybrid packing.

Implementing these algorithms across various software

environments can enhance their flexibility and usability, making them more accessible to a broader range of users, including non-experts. Developing user-friendly interfaces and integration tools that allow seamless application in different industrial settings would significantly increase adoption. Ensuring that these algorithms are easily modifiable and adaptable to specific user needs will further improve their practicality and effectiveness. In summary, the primary directions for the future development of these packing algorithms are to address combinatorial explosions in strip packing by refining search methods and preserving intermediate proposals and to enhance the generalizability of bin packing models for various cutting processes. Advancing machine learning techniques for hybrid packing is also crucial. Expanding the search area in strip packing, exploring less explored solution spaces in bin packing, and incorporating diverse optimization strategies in hybrid packing are key steps forward. Additionally, integrating these algorithms into various software environments will enhance their flexibility and usability, leading to more robust and adaptable solutions for a range of packing problems. By embracing these advancements, we can achieve more efficient, reliable, and scalable packing solutions that better meet the needs of modern industrial applications.

6 Conclusions

The use of various nesting algorithms in shipbuilding has significantly improved material utilization, reduced waste, and enhanced overall production efficiency. These algorithms can be classified into exact, heuristic, meta-heuristic, and hybrid categories, with each offering distinct benefits for nesting optimization. Exact algorithms provide precise solutions but are often hindered by high computational times. Heuristic algorithms deliver faster, more practical solutions, although they may not always achieve the optimal result. Metaheuristic algorithms offer a balanced approach, optimizing both solution quality and computational efficiency, making them particularly suitable for complex nesting challenges. Hybrid algorithms combine the strengths of different approaches, offering robust and efficient solutions. A systematic review of these algorithms highlights their performance across various criteria, emphasizing their practical applicability in real-world shipbuilding scenarios. This comprehensive evaluation underscores the importance of selecting the most effective algorithm to optimize material usage, reduce costs, and improve production timelines, ultimately promoting sustainable and efficient shipbuilding practices. Moreover, ongoing research suggests potential advancements, such as integrating AI, machine learning, and real-time adaptability, to further enhance the efficiency and effectiveness of nesting solutions. Overall, nesting algo-

rithms are essential tools for enhancing productivity and sustainability in shipbuilding, with broad applications across various manufacturing sectors.

Acknowledgement The authors would like to thank the Maritime Technology and Resources Study Program, Department of Mechanical Engineering, Universitas Indonesia, for supporting this research and the Indonesia Endowment Funds for Education or Lembaga Pengelola Dana Pendidikan (LPDP) from the Ministry of Finance for the funding support towards this article review under scholarship contract number 0000559/TRP/M/19/lpdp2023.

Competing interest The authors have no competing interests to declare that are relevant to the content of this article.

References

- Abdou K, Mohammed O, Eskandar G, Ibrahim A, Matt PA, Huber MF (2023) Smart nesting: estimating geometrical compatibility in the nesting problem using graph neural networks. *Journal of Intelligent Manufacturing* 35(2024): 2811-2827. <https://doi.org/10.1007/s10845-023-02179-0>
- Alblas G, Puijn J (2024) Are current shipbuilding cost estimation methods ready for a sustainable future? A literature review of cost estimation methods and challenges. *International Shipbuilding Progress* 71(1): 3-28. <https://doi.org/10.3233/isp-230009>
- Alvarez-Valdes R, Martinez A, Tamarit JM (2013) A branch and bound algorithm for cutting and packing irregularly shaped pieces, *International Journal of Production Economics* 145(2): 463-477. <https://doi.org/10.1016/j.ijpe.2013.04.007>
- Arnaout JP, ElKhouri C, Karayaz G (2020) Solving the multiple level warehouse layout problem using ant colony optimization. *Operational Research* 20(1): 473-490. <https://doi.org/10.1007/s12351-017-0334-5>
- Ashari IA, Muslim MA, Alamsyah A (2016) Comparison performance of genetic algorithm and ant colony optimization in course scheduling optimizing. *Scientific Journal of Informatics* 3(2): 149-158. <https://doi.org/10.15294/sji.v3i2.7911>
- Azevedo BF, Rocha AMAC, Pereira AI (2024) Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Machine Learning* 113(7): 4055-4097. <https://doi.org/10.1007/s10994-023-06467-x>
- Baso S, Musrina M, Anggriani ADE (2020) Strategy for improving the competitiveness of shipyards in the eastern part of indonesia. *Kapal: Jurnal Ilmu Pengetahuan dan Teknologi Kelautan* 17(2): 74-85. <https://doi.org/10.14710/kapal.v17i2.29448>
- Basuki M, Manfaat D, Nugroho S, Dinariyana AAB (2012) Improvement of the process of new business of ship building industry. *Journal of Economics, Business, and Accountancy Ventura* 15(2): 187-204. <https://doi.org/10.14414/jebav.v15i2.74>
- Bettemir ÖH, Sonmez R (2015) Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling. *Journal of Management in Engineering* 31(5): 1-8. [https://doi.org/10.1061/\(asce\)me.1943-5479.0000323](https://doi.org/10.1061/(asce)me.1943-5479.0000323)
- Calabrese M, Primo T, Del Prete A, Filitti G (2022) Nesting algorithm for optimization part placement in additive manufacturing. *International Journal of Advanced Manufacturing Technology* 119(7-8): 4613-4634. <https://doi.org/10.1007/s00170-021-08130-y>

- Canellidis V, Giannatsis J, Dedoussis V (2013) Efficient parts nesting schemes for improving stereolithography utilization. *CAD Computer Aided Design* 45(5): 875-886. <https://doi.org/10.1016/j.cad.2012.12.002>
- Cherri LH, Mundim LR, Andretta M, Toledo FMB, Oliveira JF, Carravilla MA (2016) Robust mixed-integer linear programming models for the irregular strip packing problem. *European Journal of Operational Research* 253(3): 570-583. <https://doi.org/10.1016/j.ejor.2016.03.009>
- Cherri LH, Cherri AC, Soler EM (2018) Mixed integer quadratically-constrained programming model to solve the irregular strip packing problem with continuous rotations. *Journal of Global Optimization* 72(1): 89-107. <https://doi.org/10.1007/s10898-018-0638-x>
- Dijlali A, Fouad M, Kouloughli S (2019) Optimizing the placement of irregular shapes 2D Nesting. 4th International Conference on Power Electronics and their Applications, ICPEA, 1-5. <https://doi.org/10.1109/ICPEA1.2019.8911153>
- Egeblad J, Nielsen BK, Odgaard A (2007) Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research* 183(3): 1249-1266. <https://doi.org/10.1016/j.ejor.2005.11.063>
- Elhaddad YR (2012) Combined simulated annealing and genetic algorithm to solve optimization problems. *World Academy of Science, Engineering and Technology* 6(8): 1508-1510. <https://doi.org/10.5281/zenodo.1057697>
- Fang J, Rao Y, Liu P, Zhao X (2021) Sequence transfer-based particle swarm optimization algorithm for irregular packing problems. *IEEE Access* 9: 131223-131235. <https://doi.org/10.1109/ACCESS.2021.3114331>
- Fang J, Rao Y, Zhao X, Du B (2023a) A hybrid reinforcement learning algorithm for 2D irregular packing problems. *Mathematics* 11(2): 327. <https://doi.org/10.3390/math11020327>
- Fang J, Rao Y, Luo Q, Xu J (2023b) Solving one-dimensional cutting stock problems with the deep reinforcement learning. *Mathematics*, 11(4): 1028. <https://doi.org/10.3390/math11041028>
- Fekete SP, Morr S, Scheffer C (2019) Split packing: Algorithms for packing circles with optimal worst-case density. *Discrete and Computational Geometry* 61(3): 562-594. <https://doi.org/10.1007/s00454-018-0020-2>
- Fernandez LC, Bennel JA, Martinez-Sykora A (2022) Voxel-based solution approaches to the 3D irregular packing problem. *Operations Research* 71(4): 1021-1439. <https://doi.org/10.1287/opre.2022.2260>
- Firmansyah MR, Asri S, Fachrudin F, Mustafa W, Clausthaldi FR (2021) Identifying and formulating information system of ship production process for an Indonesian small shipyard. *International Journal of Metacentre* 1(1): 12-23. <http://ijm-nasp.unhas.ac.id/index.php/ijm/article/view/5>
- Fujita K, Akagi S, Hirokawa N (1993) Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm. *ASME 1993 Design Technical Conferences* 19, 477-484. <https://doi.org/10.1115/DETC1993-0337>
- Gardeyn J, Wauters T (2022) A goal-driven ruin and recreate heuristic for the 2D variable-sized bin packing problem with guillotine constraints. *European Journal of Operational Research* 301(2): 432-444. <https://doi.org/10.1016/j.ejor.2021.11.031>
- Gomes AM, Oliveira JF (2006) Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research* 171(3): 811-829. <https://doi.org/10.1016/j.ejor.2004.09.008>
- Gomez JC, Terashima-Marín H (2018) Evolutionary hyper-heuristics for tackling bi-objective 2D bin packing problems. *Genetic Programming and Evolvable Machines* 19(1-2): 151-181. <https://doi.org/10.1007/s10710-017-9301-4>
- González-San-Martín J, Cruz-Reyes L, Gómez-Santillán C, Fraire H, Rangel-Valdez N, Dorronsoro B, Quiroz-Castellanos M (2023) Comparative study of heuristics for the one-dimensional bin packing problem. *Studies in Computational Intelligence* 1096: 293-305. https://doi.org/10.1007/978-3-031-28999-6_19
- Goodman ED, Tetelbaum AY, Kureichik VM (1994) A genetic algorithm approach to compaction, bin packing, and nesting problems. (August). Available from https://www.researchgate.net/publication/2776320_A_Genetic_Algorithm_Approach_to_Compaction_Bin_Packing_and_Nesting_Problems[Accessed on Jul. 05, 2024]
- Grange A, Kacem I, Martin S (2018) Algorithms for the bin packing problem with overlapping items. *Computers & Industrial Engineering* 115: 331-341. <https://doi.org/10.1016/j.cie.2017.10.015>
- Gray A (2019) Body as voice: Restorative dance/movement psychotherapy with survivors of relational trauma. *The Routledge International Handbook of Embodied Perspectives in Psychotherapy: Approaches from Dance Movement and Body Psychotherapies*, 147-160. <https://doi.org/10.4324/9781315159416>
- Gunbeyaz SA, Kurt RE, Turan O (2022) Investigation of different cutting technologies in a ship recycling yard with simulation approach. *Ships and Offshore Structures* 17(3): 564-576. <https://doi.org/10.1080/17445302.2020.1846916>
- Guo B, Hu J, Wu F, Peng Q (2020) Automatic layout of 2D free-form shapes based on geometric similarity feature searching and fuzzy matching. *Journal of Manufacturing Systems* 56: 37-49. <https://doi.org/10.1016/j.jmsy.2020.04.019>
- Hamada K, Ikeda Y, Tokumoto H, Hase S (2019) Development of automatic nesting system for shipbuilding using the branch-and-bound method. *Journal of Marine Science and Technology (Japan)* 24(2): 398-409. <https://doi.org/10.1007/s00773-018-0559-x>
- Han X, Iwama K, Ye D, Zhang G (2007) Strip packing vs. bin packing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4508 LNCS(10231060): 358-367. https://doi.org/10.1007/978-3-540-72870-2_34
- Hartono N, Ismail AH, Zeybek S, Caterino M, Jiang K, Sahin M, Natalia C (2022) The 1-D bin packing problem optimisation using bees algorithm. *Journal Industrial Services* 7(2): 323. <https://doi.org/10.36055/jiss.v7i2.14387>
- Hifi M (2001) Exact algorithms for large-scale unconstrained two and three staged cutting problems. *Computational Optimization and Applications* 18(1): 63-88. <https://doi.org/10.1023/A:1008743711658>
- Hong PC, Park YS, Hwang DW, Sepehr MJ (2024) A growth theory perspective on the competitive landscape of shipbuilding: a comparative study of Japan, Korea, and China. *Maritime Economics & Logistics*, 26: 462-489. <https://doi.org/10.1057/s41278-023-00279-5>
- Hu S, Liu T, Wang S, Kao Y, Sun X (2015) A hybrid heuristic algorithm for ship block construction space scheduling problem. *Discrete Dynamics in Nature and Society* 7: 1-6. <https://doi.org/10.1155/2015/841637>
- Huang L, Chen X, Huo W, Wang J, Zhang F, Bai B, Shi L (2021) Branch and bound in mixed integer linear programming problems: A survey of techniques and trends. *Discrete Optimization (November)*: 1-29. <https://doi.org/10.48550/arXiv.2111.06257>
- ILO (2023) Sectoral skills priorities for the shipbuilding industry: Skills demand and supply insights based on the ILO Skills for

- Trade and Economic Diversification (STED) methodology Skills for prosperity Indonesia. Available from <https://www.ilo.org/publications/sectoral-skills-priorities-shipbuilding-industry> [Accessed on Jul. 05, 2024]
- Ingber L (1993) Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling* 18(11): 29-57. [https://doi.org/10.1016/0895-7177\(93\)90204-C](https://doi.org/10.1016/0895-7177(93)90204-C)
- Japan Ship Technology Research Association (2020) Roadmap to zero emission from international shipping. Shipping Zero Emission Project (March): 1-136. Available from <https://www.mlit.go.jp/common/001354314.pdf> [Accessed on Jul. 05, 2024]
- Jones DR (2014) A fully general, exact algorithm for nesting irregular shapes. *Journal of Global Optimization* 59(2-3): 367-404. <https://doi.org/10.1007/s10898-013-0129-z>
- Kamola-Cieslik M (2021) Changes in the global shipbuilding industry on the examples of selected states worldwide in the 21st Century. *European Research Studies Journal XXIV (Issue 2B)*: 98-112. <https://doi.org/10.35808/ersj/2205>
- Kareem SW, Ali KWH, Askar S, Xoshaba FS, Roojwan H (2022) Metaheuristic algorithms in optimization and its application: a review. *Journal on Advanced Research in Electrical Engineering* 6(1): 7-12. <https://doi.org/10.12962/jaree.v6i1.216>
- Kennedy J, Eberhart R (1995) Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, 105-111. https://doi.org/10.1007/978-3-031-17922-8_4
- Kierkosz I, Łuczak M (2019) A one-pass heuristic for nesting problems. *Operations Research and Decisions* 29(1): 37-60. <https://doi.org/10.5277/ord190103>
- Ko MC, Hsieh SJ (2023) Design and evaluation of algorithms for stacking irregular 3D objects using an automated material handling system. *International Journal of Advanced Manufacturing Technology* 126(5-6): 1951-1964. <https://doi.org/10.1007/s00170-023-11248-w>
- Ko S, Shinoda T (2021) The development and challenges of China's shipbuilding industry. 7th International Conference on Ship and Offshore Technology, ICSOT Indonesia, 25-34. <https://doi.org/10.3940/rina.icsotindonesia.2021.04>
- Lallier C, Blin G, Pinaud B, Vézard L (2024) Graph neural network comparison for 2D nesting efficiency estimation. *Journal of Intelligent Manufacturing* 35(2): 859-873. <https://doi.org/10.1007/s10845-023-02084-6>
- Lee J, Kim BI, Kim SH (2023) A matheuristic algorithm for block assignment problem in long-term production planning in the shipbuilding industry. *Computers and Industrial Engineering* 184 (February): 109603. <https://doi.org/10.1016/j.cie.2023.109603>
- Lee ZJ, Su SF, Chuang CC, Liu KH (2008) Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. *Applied Soft Computing Journal* 8(1): 55-78. <https://doi.org/10.1016/j.asoc.2006.10.012>
- Li YB, Sang HB, Xiong X, Li YR (2021) An improved adaptive genetic algorithm for two-dimensional rectangular packing problem. *Applied Sciences (Switzerland)* 11(1): 1-20. <https://doi.org/10.3390/app11010413>
- Liao X, Guo S, Ou C (2016) An physical force-driven packing optimization design method in strip packing problems. *International Forum on Energy, Environment and Sustainable Development (IFEESD)*, 483-487. <https://doi.org/10.2991/ifeesd-16.2016.86>
- Liu C, Si Z, Hua J, Jia N (2023) Optimizing two-dimensional irregular packing: A hybrid approach of genetic algorithm and linear programming. *Applied Sciences* 13(22): 12474. <https://doi.org/10.3390/app132212474>
- Liu X, Chang D (2023) An improved method for optimizing CNC laser cutting paths for ship hull components with thicknesses up to 24 mm. *Journal of Marine Science and Engineering* 11(3): 652. <https://doi.org/10.3390/jmse11030652>
- Lodi A (1999) Algorithms for two-dimensional bin packing and assignment problems. *Doktorarbeit, DEIS, Universita di Bologna*. Available from http://www.dei.unipd.it/~fisch/ricop/tesi/tesi_dottorato_Lodi_1999.pdf [Accessed on Jul. 05, 2024]
- Lodi A, Martello S, Monaci M (2002) Two-dimensional packing problems: A survey. *European Journal of Operational Research* 141(2): 241-252. [https://doi.org/10.1016/S0377-2217\(02\)00123-6](https://doi.org/10.1016/S0377-2217(02)00123-6)
- Lourenço HR, Martin O, Stützle T (2001) A Beginner's introduction to iterated local search. *Proceeding of the 4th Metaheuristics International Conference* 1-6. Available from https://econ.upf.edu/~ramalhin/PDFfiles/2001_MIC_FILS.pdf [Accessed on Jul. 05, 2024]
- Martello S, Monaci M, Vigo D (2003) An exact approach to the strip-packing problem. *INFORMS Journal on Computing* 15(3): 310-319. <https://doi.org/10.1287/ijoc.15.3.310.16082>
- Martinez-Martinez G, Sanchez-Romero JL, Jimeno-Morenilla A, Mora-Mora H (2021) An improved nesting algorithm for irregular patterns. (October). <https://doi.org/10.20944/preprints2021.10.0334.v1>
- Martinez-Sykora A, Alvarez-Valdes R, Bennell JA, Ruiz R, Tamarit JM (2017) Matheuristics for the irregular bin packing problem with free rotations. *European Journal of Operational Research* 258(2): 440-455. <https://doi.org/10.1016/j.ejor.2016.09.043>
- Mundim LR, Andretta M, Carravilla MA, Oliveira JF (2018) A general heuristic for two-dimensional nesting problems with limited-size containers. *International Journal of Production Research* 56(1-2): 709-732. <https://doi.org/10.1080/00207543.2017.1394598>
- Muriyatmoko D, Djunaidy A, Muklason A (2024) Heuristics and metaheuristics for solving capacitated vehicle routing problem: An algorithm comparison. *Procedia Computer Science* 234: 494-501. <https://doi.org/10.1016/j.procs.2024.03.032>
- Mustafa W, Asri S, L FF, Rizal M, Clausthaldi FR, Rijal A, Jafar J, Putra M (2022) Characteristics of SPOB ship hull block design based on the plate supply requirements. *International Journal of Metacentre* 2(2): 1-8. <http://ijm-nasp.unhas.ac.id/index.php/ijm/article/view/18>
- Na GY, Yang J (2024) Two-dimensional polygon classification and pairwise clustering for pairing in ship parts nesting. *Journal of Intelligent Manufacturing* 35(7): 3169-3184. <https://doi.org/10.1007/s10845-023-02196-z>
- Nor DSARM, Nazery K (2008) An overview of the shipbuilding industry in malaysia. 5th Asia Maritime Conference 1-12. Available from https://www.researchgate.net/publication/320182999_An_Overview_of_the_Shipbuilding_Industry_A_Literature_Review [Accessed on Jul. 05, 2024]
- Okubo Y, Mitsuyuki T (2022) Ship production planning using shipbuilding system modeling and discrete time process simulation. *Journal of Marine Science and Engineering* 10(2): 176. <https://doi.org/10.3390/jmse10020176>
- Okumoto Y, Takeda Y, Mano M, Okada T (2009) Design of ship hull structures: A practical guide for engineers, Springer Berlin Heidelberg, 1-578. <https://doi.org/10.1007/978-3-540-88445-3>
- Oliveira A, Gordo JM (2018) Implementation of new production processes in panel's line. *Maritime Transportation and Harvesting of Sea Resources* 2(October 2017): 763-773. Available from https://www.researchgate.net/publication/320307455_Implementation_of_new_production_processes_in_panel%27s_line [Accessed on

- Jul. 05, 2024]
- Page MJ, McKenzie J, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, Shamseer L, Tetzlaff JM, Akl AEA, Brennan SE, Chou R, Glanville J, Grimshaw JM, Hróbjartsson A, Lalu MM, Li T, Loder EW, Mayo-Wilson E, McDonald S, McGuinness LA, Stewart LA, Thomas J, Tricco AC, Welch VA, Whiting P, Moher D (2020) The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. p. 1. <https://doi.org/10.1136/bmj.n71>
- Page MJ, McKenzie J, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, Shamseer L, Tetzlaff JM, Akl AEA, Brennan SE, Chou R, Glanville J, Grimshaw JM, Hróbjartsson A, Lalu MM, Li T, Loder EW, Mayo-Wilson E, McDonald S, McGuinness LA, Stewart LA, Thomas J, Tricco AC, Welch VA, Whiting P, Moher D (2021) PRISMA 2020 Checklist. *The BMJ*: 2020-2021. <https://doi.org/10.1136/bmj.n71>
- Perez-Martinez J, Fernandez PR (2023) Material and production optimization of the ship design process by introducing CADs from early design stages. *Journal of Marine Science and Engineering* 11(1): 233. <https://doi.org/10.3390/jmse11010233>
- Pospelov KN, Vatamaniuk IV, Lundaeva KA, Gintciak AM (2023) Heuristic approach to planning complex multi-stage production systems. *International Journal of Technology* 14(8): 1790-1799. <https://doi.org/10.14716/ijtech.v14i8.6833>
- Rakotonirainy RG, Vuuren JHV (2020) Improved metaheuristics for the two-dimensional strip packing problem. *Applied Soft Computing Journal* 92: 106268. <https://doi.org/10.1016/j.asoc.2020.106268>
- Rigo P, Caprace JD (2011) Optimization of ship structures. *Marine Technology and Engineering* 2(October): 925-944. <https://api.semanticscholar.org/CorpusID:202597610>
- Ru N, Jianhua Y (2008) A GA and particle swarm optimization based hybrid algorithm. 2008 IEEE Congress on Evolutionary Computation, CEC 2008: 1047-1050. <https://doi.org/10.1109/CEC.2008.4630925>
- Saiyara, N (2024) Material of construction, steel material preparation & plate cutting for ship hull. <https://doi.org/10.6084/m9.figshare.26049247>
- Sari WR, Gunawan, Muzhoffer DAF (2024) Systematic review of optimization of nesting system in shipbuilding. 5th International Conference on Information Technology and Advanced Mechanical and Electrical Engineering (ICITAMEE) [Preprint]
- Selow R, Neves F, Lopes HS (2007) Genetic algorithms for the nesting problem in the packing industry. *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2007)* 1-6. Available from https://www.researchgate.net/publication/221616369_Genetic_Algorithms_for_the_Nesting_Problem_in_the_Packing_Industry [Accessed on Jul. 05, 2024]
- Smith K, Diaz R, Shen Y (2022) Development of a framework to support informed shipbuilding based on supply chain disruptions. *Procedia Computer Science* 200: 1093-1102. <https://doi.org/10.1016/j.procs.2022.01.309>
- Son S, Kim B, Ryu C, Hwang I, Jung CH, Shin JG (2020) Production automation system for three-dimensional template pieces used to evaluate shell plate completeness. *International Journal of Naval Architecture and Ocean Engineering* 12: 116-128. <https://doi.org/10.1016/j.ijnaoe.2019.08.001>
- Sutrisno S, Suef M, Ma'ruf B (2024) A conceptual model of agile manufacturing in the shipbuilding industry for improving shipyard performance. *International Journal of Agile Systems and Management* 17(2): 246-268. <https://doi.org/10.1504/IJASM.2024.138863>
- Suzuki S, Muraoka R, Obinata T, Endo S, Horita T, Omata K (2004) Steel products for shipbuilding. JFE Technical Report 2(2): 41-48. Available from <https://www.jfe-steel.co.jp/en/research/report/002/pdf/002-05.pdf> [Accessed on Jul. 05, 2024]
- Timmerman M (2013) Optimization methods for nesting problems. University West 49. Available from <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A631363&dsid=8735> [Accessed on Jul. 05, 2024]
- Ting TO, Yang XS, Cheng S, Huang K (2015) Hybrid metaheuristic algorithms: Past, present, and future. *Studies in Computational Intelligence* 585: 71-83. https://doi.org/10.1007/978-3-319-13826-8_4
- Uemori R, Fujioka M, Inoue T, Minagawa M, Ichikawa K, Shirahata H, Nose T (2012) Steels for marine transportation and construction. *Nippon Steel Technical Report* (101): 37-47. Available from https://www.nipponsteel.com/en/tech/report/nsc/pdf/NSTR101-08_tech_review-1-2.pdf [Accessed on Jul. 05, 2024]
- Umetani S, Murakami S (2022) Coordinate descent heuristics for the irregular strip packing problem of rasterized shapes. *European Journal of Operational Research* 303(3): 1009-1026. <https://doi.org/10.1016/j.ejor.2022.03.034>
- Vielma JP, Ahmed S, Nemhauser GL (2008) A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing* 20(3): 438-450. <https://doi.org/10.1287/ijoc.1070.0256>
- Wang A, Hanselman CL, Gounaris CE (2018) A customized branch-and-bound approach for irregular shape nesting. *Journal of Global Optimization* 71(4): 935-955. <https://doi.org/10.1007/s10898-018-0637-y>
- Wang Y, Zhang P, Gu Y, Zhang X, Jin C, Chen M (2024) Research of ship nesting based on maximum residual rectangle strategy integrated improved genetic algorithm. *Advances in Transdisciplinary Engineering* 46: 499-506. <https://doi.org/10.3233/ATDE231141>
- Wang Y, Han Y, Wang Y, Li J, Gao K, Liu Y (2023) An effective two-stage iterated greedy algorithm for distributed flowshop group scheduling problem with setup time. *Expert Systems with Applications* 233: 120909. <https://doi.org/10.1016/j.eswa.2023.120909>
- Williams RI, Clark LA, Clark WR, Raffo DM (2021) Re-examining systematic literature review in management research: Additional benefits and execution protocols. *European Management Journal* 39(4): 521-533. <https://doi.org/10.1016/j.emj.2020.09.007>
- Wu TH, Chen JF, Low C, Tang PT (2003) Nesting of two-dimensional parts in multiple plates using hybrid algorithm. *International Journal of Production Research* 41(16): 3883-3900. <https://doi.org/10.1080/0020754031000149239>
- Xie SQ, Wang GG, Liu Y (2007) Nesting of two-dimensional irregular parts: An integrated approach. *International Journal of Computer Integrated Manufacturing* 20(8): 741-756. <https://doi.org/10.1080/09511920600996401>
- Xu J, Yang W (2022) Multi-objective steel plate cutting optimization problem based on real number coding genetic algorithm. *Scientific Reports* 12(1): 1-20. <https://doi.org/10.1038/s41598-022-27100-2>
- Xu JJ, Wu XS, Liu HM, Zhang M (2017) An optimization algorithm based on no-fit polygon method and hybrid heuristic strategy for irregular nesting problem. *Chinese Control Conference, CCC*, 2858-2863. <https://doi.org/10.23919/ChiCC.2017.8027799>
- Xu YX (2016) An efficient heuristic approach for irregular cutting stock problem in ship building industry. *Mathematical Problems in Engineering* 8703782: 12. <https://doi.org/10.1155/2016/8703782>

- Yu-ichi K (2007) Status & prospects of shipbuilding steel and its weldability. *Transactions of JWRI* 36(1): 1-6. <https://doi.org/10.18910/9500>
- Yue M (1991) A simple proof of the inequality $FFD(L) \leq 11/9 OPT(L)+1$, $\forall L$ for the FFD bin-packing algorithm. *Acta Mathematicae Applicatae Sinica* 7(4): 321-331. <https://doi.org/10.1007/BF02009683>
- Zhang C, Han F, Zhang W (2019) A cutting sequence optimization method based on tabu search algorithm for complex parts machining. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 233(3): 745-755. <https://doi.org/10.1177/0954405417752527>
- Zhang H, Liu Q, Wei L, Zeng J, Leng J, Yan D (2022) An iteratively doubling local search for the two-dimensional irregular bin packing problem with limited rotations. *Computers and Operations Research* 137: 105550. <https://doi.org/10.1016/j.cor.2021.105550>
- Zhang H, Yao S, Liu Q, Wei L, Lin L, Leng J (2023) An exact approach for the constrained two- dimensional guillotine cutting problem with defects. *International Journal of Production Research* 61(9): 2986-3003. <https://doi.org/10.1080/00207543.2022.2074907>