August 14, 2022

SHIP DECK OBJECT PLACEMENT OPTIMIZATION USING A MANY-OBJECTIVE BILEVEL APPROACH

Noah J. Bagazinski *
Department of Mechanical Engineering
Mssachusetts Institute of Technology
Cambridge, Massachusetts, 02139
Email: noahbagz@mit.edu

Faez Ahmed
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts, 02139
Email: faez@mit.edu

ABSTRACT

The placement of objects on a ship is critical to many facets of the performance of a ship. Most notably, the mass distribution properties of objects in a ship affect the ship's stability, trim, and structural loading. Information gathered from object placement optimization can allow naval architects to further optimize the design of the whole ship by potentially reducing the structural weight of the vessel, and adjusting the shape of the hull or the general arrangements based on available space in the ship. This paper presents a novel, many-objective bin packing problem for object placement across multiple decks on a ship. This problem is also highly constrained to avoid object intersection and protrusion. The problem was optimized with the NSGA-II algorithm, utilizing a heuristic population initialization and by separating the objectives into a bilevel optimization scheme. The bilevel scheme decouples certain objectives and design variables from the rest of the problem and sequences the evaluation for the objectives in a two-stage process. The hypervolume of the final population measured the performance of the optimization test. The results indicate that sequencing the objectives with a bilevel scheme produces an 80.3% larger hypervolume than an all-in-one optimization for the same problem. The findings from this study provide a systematic way by combining concepts from many-objective optimization, bin packing heuristics, and bilevel optimization to sequence the optimization of many-objective, object placement problems.

*Address all correspondence to this author.

NOMENCLATURE

LOA	Length Overall of the Ship
BOA	Beam Overall of the Ship
N	Number of Objects Placed in the Ship
M	Number of Decks on the Ship
VCG	Vertical Center of Gravity
LCG	Longitudinal Center of Gravity
LCB	Longitudinal Center of Buoyancy
YCG	Transverse Center of Gravity
GM	Metacentric Height
BM_t	Height from VCB to the Metacenter
VCB	Vertical Center of Buoyancy
Δ	Ship Displacement
ρ	Density of Saltwater = 1025 kg/m^3

INTRODUCTION

Ships are highly complex and expensive systems and are often built as one-off designs or in limited production. The complexity in ship design stems from the need to balance many conflicting design requirements while finding potential designs in an immensely large design space. Therefore, ships require significant effort in design analysis to produce a product that is effective, efficient, and inexpensive to the customer. The early stages of ship design are the most critical in affecting much of the performance of the final ship in areas such as stability, structural strength, and hydrodynamics. Providing information about potential designs as early in the design process as possible gives naval architects better insight into delivering quality ship designs to their customers. One critical area in a ship's design is the mass distribution properties of objects placed in the hull. The mass properties affect the ship's stability, trim, and structural loading.

Information gathered from object placement optimization can allow naval architects to further optimize the design of the whole ship by potentially reducing the structural weight of the vessel, and adjusting the shape of hull or the general arrangements based on available space in the ship. Many-objective design optimization provides an optimal set of designs across several objectives that allow designers to choose a design that is best tailored to a customer's needs early in the design process. Many of these customer needs can be decoupled, where the optimization of one objective does not necessarily affect the optimization of another. For example, structural strength and hull drag are not directly tied to each other, so the shape of the hull and the structure could be designed simultaneously or sequentially. Another type of optimization, bilevel algorithms, provide an opportunity in ship design, as they split the optimization problem into two or more distinct components, allowing decoupled objectives to be evaluated sequentially.

The following sections detail a process of placing objects across multiple ship decks to optimize the mass properties of the entire ship using a bilevel optimization scheme. The first objective seeks to minimize the net vertical center of gravity of all the objects so, increasing the stability of the ship. The second objective seeks to minimize differences in the total footprint area of objects placed in each deck, so that each deck is equally utilized. The third and fourth objectives seek to position the transverse and longitudinal centers of gravity so that the ship does not trim or heel in any direction. The fifth objective aims to minimize the structural loading of the ship by spreading the weight of the objects longitudinally to match the buoyancy distribution of the ship. The sixth objective seeks to pack the set of objects as compactly as possible so to allow extra space near the surface of the hull for future design alterations. The optimization targets the effect of object placement on upright stability, equal deck utilization, and the structural needs of the ship, giving designers a variety of design arrangements to consider while designing a ship.

The novel contributions of this paper are:

1. Devised problem formulation of ship deck packing. Defined six objectives critical for this problem and four types of constraints.
2. Showed gains obtained using heuristic initialization for all-in-one optimization problems (83.3% increase in final hypervolume).
3. Showed gains obtained by leveraging a bilevel optimization to seed an initial population for an all-in-one optimization (62.9% increase in final hypervolume from the result after heuristic initialization).
4. Showed gains obtained by strategically sequencing the optimization of many objectives using bilevel optimization (80.3% increase in final hypervolume from the result after heuristic initialization).

PREVIOUS WORK

This section provides a background for the work detailed in this paper. The first subsection contains definitions useful to optimization problems. The second subsection provides a background in object placement algorithms and bin packing optimization. The third subsection presents bilevel optimization. Finally, the fourth subsection details prior work in multi-objective ship design optimization.

Terminology

Optimization is the process of manipulating the inputs of a problem to minimize the output of the problem. Multi-objective optimization aims to minimize two or more problems, called objectives, simultaneously. When a multi-objective problem has more than three objectives, it is often termed as a "many-objective" optimization problem, due to the additional complexity of dealing with higher dimensional data. In these methods, multiple objectives typically compete, meaning that one objective is improved at the detriment of another. Multi-objective problems rely on population based optimizations where a set of individuals, each representing all of the inputs to the problem, are manipulated to produce a set output for each objective. An individual is considered dominant if, for any given objective, there does not exist another individual with that objective score lower without the detriment of the other objectives. The set of dominant individuals form a limit, called the Pareto front, that defines the best possible objective evaluations from the optimization.

Object Placement Algorithms and Optimization

From chip floorplanning problems to placing components in a satellite, the problem of placing multiple objects in a limited space is of immense importance to many industries. The problem is also extremely challenging to solve [1].

One of the most challenging aspects of object placement is the prevention of overlapping objects. Object placement in 2D and 3D is highly constrained and can be expensive to compute in order to prevent object intersection. Potential complexities in placing objects include object geometry, rotation of objects, interference checking algorithms, and morphing components. The number of constraints for interference checking is order N^2, where N is the number of objects placed by the algorithm, and interference checking functions can be computationally expensive based on the geometry of the objects [1]. Therefore, in order to simplify this complexity as much as possible, all objects placed into ships in this study will be represented as 2D rectangular footprints of the length and width of each object. This way, interference checking between any two objects is simplified to measuring an area of overlap between two rectangles. Further details of the object generation are found in the Methods section.

[1] Floorplanning is an NP-hard problem

Many heuristic algorithms [2–5] exist to pack a set of objects that minimizes the total volume of the objects placed. A common heuristic is the back-left packing algorithm, which is used to inspire the packing heuristic used later in the paper [6]. Another heuristic, pattern search algorithms, follows a structured search procedure to explore the feasibility and objective values of potential layouts, and have been used to find optimal arrangements of objects [7–11]. However, these heuristic algorithms focus on a single objective, such as minimizing the volume of packing, while satisfying constraints. In this study, there are six objectives, which makes this object placement problem considerably harder for a heuristic to solve.

Simulated annealing algorithms mimic the annealing of metals as a method to optimize combinatorial problems. These algorithms have been used to perform multi-objective object placement optimizations for 3D components and 2D chip layouts [12, 13].

Particle swarm optimization (PSO) algorithms mimic the flow of social movement, such as a school of fish or a flock of birds. Each particle tends to search through the design space, while taking social cues from the other particles to find the "optimal" location within the design space [14]. PSO algorithms have also been used in similar bin packing optimization problems and multi-objective optimization problems [15–18].

Genetic Algorithms (GA) mimic biological evolution by "breeding" and "mutating" a population of inputs to an optimization problem and then evaluating the objectives and promoting the "fittest" individual inputs to the next generation of evaluation. GA's have been used to pack objects for both compact packing optimization and for system performance optimization [1]. These population based methods allow one to model multiple objectives and constraints simultaneously. Examples of packing optimization include improving automobile performance by arranging subsystems in the vehicle. [1, 19–21]. To the best of our knowledge, GA has not been used for six objective packing optimization. The optimizations performed in this paper utilized the NSGA-II algorithm [22], which has also been used in literature for both ship design optimization [23], and object packing optimization [1]. Additionally, the NSGA-II algorithm does not required a gradient to optimize a problem. Some of objectives laid out in the next section are non-differentiable and the inputs to the optimization are not completely differentiable, suggesting that NSGA-II a good optimizer for this problem.

Bilevel Optimization

At times, it can be beneficial to separate certain objectives and evaluate them as a two stage process. For example, a system level design can be optimized, and then a subsystem or component can optimized given the system level design (i.e., a vehicle engine can be optimized for fuel efficiency, then the transmission is optimized for the given engine to further improve efficiency). Another example is to optimize resource allocation for web services [24]. Performing an optimization in two or more stages is called bilevel optimization. The bilevel optimization process is split into an upper and lower algorithm. The upper algorithm drives the optimization, while the lower algorithm is used to refine and improve the optimization. Bilevel optimization schemes have been used with genetic algorithms in different processes and for a variety of applications. Some bilevel processes are nested where for each generation in the upper algorithm, the lower algorithm performs an entire optimization. Other bilevel processes are sequential, where the upper algorithm optimizes a set of objectives, and then the lower algorithm manipulates different inputs to either further optimize the same set of objectives as the upper algorithm, or an entirely new set of objectives [25, 26]. In systems design applications, the upper algorithm drives system level design, while the lower algorithm can drive a component level design. Systems design optimization using bilevel schemes have been shown to find Pareto dominant systems level designs with optimal component designs, which effectively achieves a set of Pareto dominant all-in-one optimized designs [27].

In the case of this study, a bilevel scheme was devised to separate and evaluate some of the objectives as a tool for initialization and for sequential optimization.

Ship Design Optimization

There are many opportunities to leverage optimization methods in ship design. Prior work in ship design optimization has ranged from single objective hull form optimization, to multi-objective design analysis of deck and general arrangements. Hull form optimization with particle swarm optimization and genetic algorithms to minimize drag have been performed for all sizes of craft, ranging from planning hulls to container ships [16, 17, 28]. Other genetic algorithms have been used to analyze design trade-offs in ship general arrangements for space allocation, human dynamics optimization, and classed object placement on a single deck [23, 29–31].

This paper will focus on object placement optimization to target the mass distribution properties of the ship with the purpose of optimizing for the ship's stability, trim, and structural loading.

METHODS

This section details the ship design used for the study, the objects to be placed on the ship, objective functions, the constraint functions, and the optimization algorithms.

Ship for this Study

The ship used in this study is a design with a prismatic hull and four decks. The ship and deck area are detailed in Table 1.

TABLE 1. SHIP DESIGN PARAMETERS USED FOR THE OBJECT PLACEMENT OPTIMIZATION

Parameter	Value
Length Overall	118.1 m
Length of Bow Taper	15.0 m
Length of Stern Taper	15.0 m
Beam Overall	17.1 m
Beam at Stern	14.0 m
Design Draft	4.3 m
Depth	10.0 m
Number of Decks	4
Area of the Waterplane	1860.5 m^2
Submerged Volume	8000.2 m^3
Displacement	8203 tons

The volume displaced by the ship in saltwater, 8000.2 m^2, translates to a displacement of 8203 tons. Assuming that the set of objects to be placed into this hull form represents the entire ship, note that the sum of the weights of the 81 objects described in the next section is 8203 tons, so that this ship satisfies Archimede's principle of buoynacy [32]. Additionally, the area of the waterplane of this ship design is equal to the deck area allowed for object placement. This assumes that the weight and volume of the objects placed into ship incorporate the structure of the ship into their placement. In order to maintain the simplicity of the design for this problem, the waterplane of each deck is identical. The waterplane profile of this ship is a triangular bow, a rectangular mid-body, and a trapezoidal stern taper. A plan view of a deck is shown in Figure 1.

Object Generation

Eighty one objects were generated for the optimization algorithm to place into the ship. The number of objects were selected to provide computational complexity, so to provide a large number of design variables to manipulate and an exponentially large number of constraints to satisfy. Further, the sizing of the objects was selected to be approximately 50% of the total deck space available, and limiting the largest dimension to allow two objects to be placed adjacently across the beam of the vessel. Each of these objects has a specified length, width, VCG, and weight. For each object, each of these four parameters could be assigned one of three values. Eighty one objects were chosen such that every combination of length, width, VCG, and weight is seen among the set of objects. Table 2 details the possible values for each parameter.

TABLE 2. POSSIBLE PARAMETER VALUES FOR OBJECT GENERATION

Parameter	Low Value	Med. Value	High Value
Length	2.82 m	5.64 m	8.46 m
Width	2.8 m	5.6 m	8.5 m
VCG	0.83 m	1.25 m	1.67 m
Weight	50.63t	101.27 t	151.91 t

The length and width values are 1/6, 1/3, and 1/2 of 99% of the Beam Overall. The VCG values are 1/6, 1/3, and 1/2 of the height between decks. The weight values sum to the displacement of the ship, and are scaled such that the ratio of lightest to medium to heaviest objects is 1:2:3. While the set of objects does not fully represent a real set of objects to be placed into a real ship, this set provides a challenging and diverse set of objects for the optimization algorithms to manipulate to minimize the objective functions. In this instance, the objects represent static loads on a ship. They do not represent objects that might move or have variable weight during a ship's voyage(such as food stores, water storage, fuel). Future work in object placement of non-static objects will explore these instances for ship design analysis. Future work will also consider object placement of specialized objects, such as machinery, tanks, landing pads, etc. and will consider relevant constraints for these objects. Different ship placement applications may have other requirements (such as the number of objects to be placed, number of decks, and mass of each object), we want to emphasize that these choices are used to demonstrate the effectiveness of different optimization algorithms. Our tests have shown that the results generalize if other problem requirements are used too. Future work will tie our methods to measurements and requirements from real-world ships.

Objective Functions

The six objective functions used in this study provide representative considerations of object placement in a ship. The following subsections detail the six objectives.

Vertical Center of Gravity

The vertical center of gravity (VCG) objective seeks to minimize the height of the vertical center of gravity of all of the objects combined. On a ship, the VCG is critical to ensure the upright stability of the vessel. The calculation of the VCG is shown in Equation 1:

$$F_1 = \frac{\sum_{i=1}^{N} Weight_i * VCG_i}{\Delta} \quad (1)$$

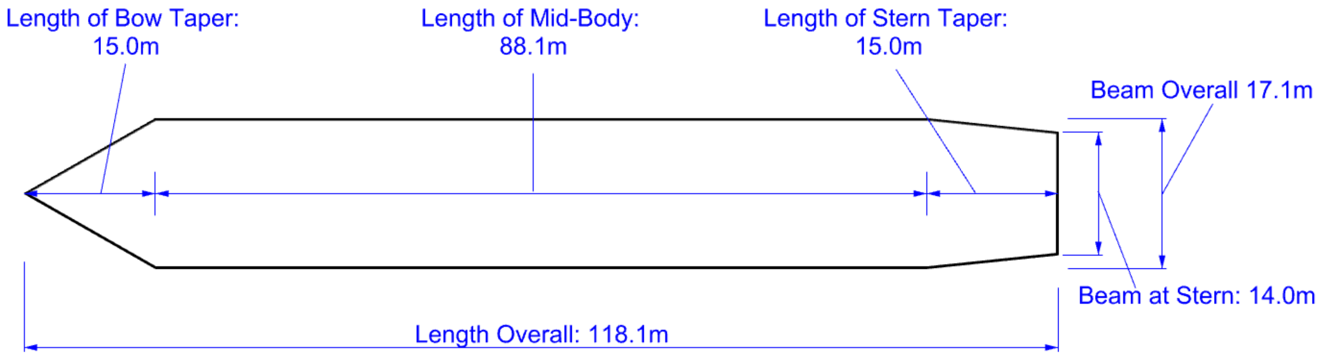

FIGURE 1. PLAN VIEW OF THE SHIP DECK SPACE.

where Δ is the ship displacement, and VCG_i is the object's VCG plus the height of the deck it is sitting on.

Deck equalization The deck equalization objective seeks to equalize the sum of the footprint areas of the objects in each deck. Equalizing the placement of objects across multiple decks ensures that artificial bottlenecks for personnel, material, or energy are limited in a real ship at sea. This objective will not directly prevent bottlenecks in and of itself, but will create an initial starting point for placement analysis [23, 29, 30]. This objective also prevents placing all objects as low as possible for the VCG objective. This target is minimized using the sum of the absolute difference between the foot print area of the objects in each deck and an even split of the objects between each deck. The deck equalization objective is defined in Equation 2 and Equation 3.

$$Util_{Deck_j} = |\sum_{i=1}^{n_j} l_i * w_i - \frac{\sum_{i=1}^{N} l_i * w_i}{M}| \tag{2}$$

$$F_2 = \frac{\sum_{j=1}^{M} Util_{Deck_j}}{\sum_{i=1}^{N} l_i * w_i} \tag{3}$$

where n_j is the number of objects in $Deck_j$ and l_i and w_i are the length and width of each object.

Transverse and Longitudinal Centers of Gravity The Transverse and Longitudinal Centers of Gravity (YCG and LCG, respectively) affect how the ship is heeled and trimmed. The target YCG of the ship is to center the weight on the centerline of the ship, or Y = 0 for this problem. The target LCG for the ship is to center the weight at the longitudinal center of buoyancy (LCB) of the ship. Equation 4 and Equation 5 show the objective functions for YCG and LCG, where the center of gravity of each object is the geometric center of its footprint:

$$F_3 = |\frac{\sum_{i=1}^{N} weight_i * (Y_i + \frac{1}{2} w_i)}{\sum_{i=1}^{N} weight_i}| \tag{4}$$

$$F_4 = |\frac{\sum_{i=1}^{N} weight_i * (X_i + \frac{1}{2} l_i)}{\sum_{i=1}^{N} weight_i} - LCB| \tag{5}$$

Longitudinal Bending Moment The maximum longitudinal bending moment is minimized by matching the longitudinal weight distribution of the objects with the longitudinal distribution of hydrostatic force by the ship's hull. The purpose of minimizing the maximum longitudinal bending moment is to reduce the loading on the structure of the ship, leading to the potential to reduce the weight of the structure of the ship [33]. Equation 6, Equation 7, Equation 8, and 9 detail the process of calculating the distributed load, vertical sheer force, and bending moment on the ship's structure and treating the ship as a simple beam:

$$Dist.Load = f(x) = Sec.Weight(x) - Sec.Area(x) * \rho * g \tag{6}$$

$$VerticalSheer(x) = V(x) = \int_0^{LOA} f(x)dx \tag{7}$$

$$BendingMoment(x) = \int_0^{LOA} V(x)dx \tag{8}$$

$$F_5 = max(|BendingMoment(x)|) \tag{9}$$

where the Sectional Weight is the weight per unit length at a given X location along the ship's length, and the Sectional Area is the cross sectional area of the submerged portion of the ship at a given X location.

Bounding Box The bounding box objective minimizes the area of a rectangle in the X-Y plane that encompasses all of the objects. This can allow free space near the limits of the deck space to allow the hull morph in potential future design optimizations. The bounding box is calculated by finding the extreme forward-most, aft-most, port-most, and starboard-most protrusions ($X_{fwd}, X_{aft}, Y_{port}, andY_{stbd}$, respectively) from the objects and computes the area of the bounding box with Equation 10:

$$F_6 = (X_{fwd} - X_{aft}) * (Y_{stbd} - Y_{port}) \tag{10}$$

Constraint Functions

The following subsection details the four types of constraints used in optimization algorithms. The stability constraint, the protrusion constraints, and the intersection constraints are used in the AiO optimization scheme. A fourth type of constraint, the Deck Utilization Constraint, is used in the upper level algorithm for the bilevel optimization scheme along with the GM constraint. The protrusion and intersection constraints are used in the lower algorithm in the bilevel optimization scheme.

Stability Constraint The stability constraint ensures that a design produces a ship that is hydrostatically stable (positive GM). The constraint is satisfied if the height of the metacenter is greater than the VCG of the objects placed in the ship, meaning the GM is positive. Equation 11 rearranges this condition as an inequality:

$$0 \geq VCG - (VCB + BM_t) \tag{11}$$

Protrusion Constraints The protrusion constraints are a set of constraints to ensure that no object intersects the limits of the hull. There are five protrusion constraints for each object, for a total of 405 protrusion constraints for the 81 objects in a design. One protrusion constraint ensures that the X position of an object is within the limits of the hull, while the other four constraints ensure that the corners of each object do not exceed the bounds of the hull given the y position and x position. Equation 12 checks the X position of the object. Equations 13-16 check the Y position of the object. Similar constraints have also be defined in prior work [23].

$$0 \geq X_i + l_i - LOA \tag{12}$$

$$0 \geq -\frac{Beam(X_i)}{2} - Y_i \tag{13}$$

$$0 \geq Y_i + w_i - \frac{Beam(X_i)}{2} \tag{14}$$

$$0 \geq -\frac{Beam(X_i + l_i)}{2} - Y_i \tag{15}$$

$$0 \geq Y_i + w_i - \frac{Beam(X_i + l_i)}{2} \tag{16}$$

Intersection Constraints The intersection constraints check the area of intersection of two objects. For the 81 objects, there are 3,240 unique object intersections to check. Equations 17-19 detail the intersection constraints for two objects. We adopted the intersection constraints from prior work [23].

$$P_{xij} = (X_i + l_i - X_j) * (X_i - X_j - l_j) - |(X_i + l_i - X_j) * (X_i - X_j - l_j)| \tag{17}$$

$$P_{yij} = (Y_i + w_i - Y_j) * (Y_i - Y_j - w_j) - |(Y_i + w_i - Y_j) * (Y_i - Y_j - w_j)| \tag{18}$$

$$0 \geq P_{xij} * P_{yij} \tag{19}$$

In situations where objects i and j are assigned to different decks, the intersection constraint value is returned as zero since their their deck assignment supersedes their X-Y position intersection.

Deck Utilization Constraint The deck utilization constraint ensures that no individual deck is overloaded with objects during the upper algorithm phase of the bilevel optimization scheme. There are M = 4 deck utilization constraints. Equation 20 details the constraint calculation for a set of n

objects in on deck j:

$$0 \geq \sum_{i=1}^{n_j} (l_i * w_i) - 0.75 * Area_{deck_j} \tag{20}$$

The limit of 75% of the total deck space was chosen to avoid potential limits on the true capacity of the deck space due to the bow and stern tapers due to the rectangular shape of the objects and the limits on object rotation. Further, this allows the objects to be packed onto the deck using the initialization heuristic described in the next section. Note that other limits can also be selected based on the requirements in different scenarios. This choice is not expected to alter any of our findings, as it only makes the problem more or less constrained for each algorithm.

Individual Design and Heuristic Initialization

An individual design is represented by *81 objects × 7 parameters = 567 values*. For each object, the seven values needed to evaluate its placement are the X position, the Y position, the deck assignment, length (x-direction), width (y direction), the weight, and the VCG of each object. The genetic algorithm, however, can manipulate 4 parameters: X position, Y position, deck assignment, and rotation of each object. The X and Y positions of each object are defined by the forward-port side corner of each object. The deck assignment is an integer value corresponding to index of the deck, where index = 0 is the bottom-most deck. The rotation of each object is a binary value indicating a 0° or a 90° rotation of the object. This value swaps the length and width values for any object that is rotated 90°.

Heuristic Initialization for Individual Designs Given the high number of constraints in this problem, utilizing a heuristic to generate an individual design will improve the optimization algorithm's ability to optimize instead of just focusing on searching for feasible designs. An efficient heuristic modeled after the back-left 2D packing algorithm [6] was devised to assign objects to different decks, pack them without any constraint violations, and spread them longitudinally to allow for better initial evaluation of the LCG and Bending Moment Objectives. The Heuristic Individual initialization follows this algorithm:

Step	Operation
1	Generate a random sequence of the N objects to be placed.
2	Randomly rotate the N objects either 0° or 90° with 50% probability for each.
3	For each object, i in the sequence, of the N objects:
4	- Assign a deck to an object. $Deck = imoduloM$ or with a prescribed deck assignment from the upper algorithm of the bilevel scheme.
5	- For each deck, the starting drop position is the front of the mid-body on the port edge of the deck.
6	- Place an object at the drop position for that object's deck
7	- The drop position for each deck is the aft, port corner of the previous object placed on a particular deck.
8	- If the drop point results in an object protruding from the hull, then reset the drop point for that deck to be the front of the mid-body and y = starboard-most protrusion of objects on that deck
9	Repeat Steps 4 through 8 for all N Objects

This heuristic initializes an individual ship design to avoid constraint violations, places an approximately equal number of objects on each deck, spreads them longitudinally first to achieve a better initialization for F4 and F5. While this heuristic is not meant to optimally pack the objects such as other 2D packing heuristics, it is used to generate a population of individuals that avoids violating any of the 3646+ constraints in this problem, allowing the optimization algorithm to focus on optimizing rather than reducing constraint violations.

Optimization Method

This section details the two different optimization methods used to produce the results section of this paper.

Six-Objective All in One Optimization The six-objective, all in one optimization (AiO) scheme leverages the standard NSGA-II algorithm from the Pymoo package for the Python programming language [34]. All six objectives and the GM, protrusion, and intersection constraints were all evaluated for each individual in each generation. For the purposes of this paper, the crossover and mutation parameters were left in the default state. The algorithm manipulates the X-position, Y-position, rotation, and deck assignment of all N objects to minimize the six objectives. Two different optimizations were performed using the AiO scheme. The first optimization had a population of 500 ship designs seeded using a completely random initialization without regard for any of the constraints. The second optimization had a population of 500 ship designs seeded using the heuristic initialization described in the previous section, having zero constraint violations across the initial population. Both of these optimizations were performed for 10,000 generations.

Bilevel Optimization The bilevel optimization scheme used in this paper separates the evaluation of the six objectives into a two-stage optimization with an upper and a lower algorithm. The bilevel scheme devised for this paper is detailed in this section.

The upper algorithm optimizes the VCG and deck equalization objectives by manipulating the deck assignment of the N objects. The GM constraint and the deck utilization constraints constrain the upper algorithm. These two objectives were selected because their only design variable is the deck assignment and that these affect the bulk of the constraint evaluations, but are not constrained by them. The X-position, Y-position, and rotation of the 81 objects are strongly affected by the constraint evaluations, yet do not affect the VCG or deck equalization objectives, therefore we can manipulate the deck assignment freely with limited constraints and can optimize for these objectives in the upper algorithm. The upper algorithm is initialized with a population of 500 design arrangements using the deck assignment heuristic from Step 4 of the heuristic initialization process. After 500 generations, an optimal individual is selected as the individual having the minimum VCG from the set of designs with the ten most-minimal values of deck equalization. Through experimentation, the difference in the minimal values of deck equalization was found to be the difference between one object placement difference in any given deck. Choosing the design with the minimum VCG from this set of design arrangements ensures that the population of design arrangements is seeded with objects that have an optimal deck assignment for the first two objectives. It is important to note that the deck assignment chosen by this process is not necessarily dominant, but is certainly well rounded as having both a low VCG and equally utilized decks. The advantage of seeding the lower algorithm with equalized decks is that each deck has a similar amount of free space to manipulate the x-position, y-position, and rotation to gain overall improvements in the remaining four objectives in the lower algorithm.

The lower algorithm is a second optimization that is seeded by the deck assignments determined by the upper algorithm. Following the heuristic initialization process, the objects are placed into their prescribed deck assignments. Two different lower algorithms were used to produce results. The first lower algorithm repeated the 6-objective, AiO optimization from the previous section, but with every individual seeded with the same deck assignment for each object as determined by the upper algorithm, and otherwise following the heuristic initialization process. The second method also initialized the population the same way, but limited NSGA-II to only manipulating the X-position, Y-position, and rotation of each object. Additionally, instead of evaluating all six objectives, this lower algorithm only evaluated the remaining four objectives as the VCG and deck equalization objectives cannot change without the manipulation of the deck assignment variables. Further, only the protrusion constraints and intersection constraints were evaluated in this lower algorithm. Both methods for the lower algorithm were evaluated for 9,500 generations so that the total number of generations run for the bilevel scheme was 10,000 generations.

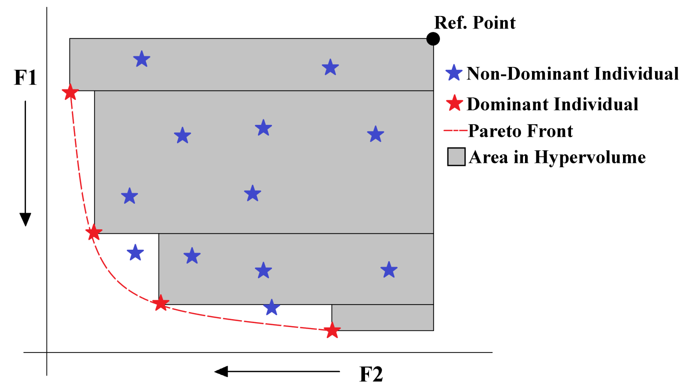

FIGURE 2. TWO DIMENSIONAL VISUALIZATION OF HYPERVOLUME EVALUATION METRIC, DOMINANCE, AND THE PARETO FRONT FOR A MINIMIZATION PROBLEM

Hypervolume Calculation A measure of a Pareto front in multi-objective optimization problems is the hypervolume metric, where each dominant individual acts as a vertex to an N-dimensional prism that is positioned in relation to a reference point [35]. N is the number of objectives in the optimization problem and the reference point is typically the worst case scenario for all the objectives [34]. Figure 2 shows a hypervolume measurement for a two objective minimization problem. The blue stars represent non-dominant individuals, while the red represents dominant individuals. In order to increase the hypervolume of a population, the optimization algorithm tries to further minimize the objectives and increase the number of dominant individuals to increase the resolution of the Pareto front. Figure 2 shows an example of two objectives and a 2-D hypervolume.

We use hypervolume to measure a six objective optimization with a 6-D hypervolume measurement. The optimization schemes are evaluated using the hypervolume indicator in the pymoo package [34]. The hypervolume of a population indicates a combined measure of the spread of the dominating individuals and the degree to which the objectives are minimized. The hypervolume measurements shown in the next section were computed using only dominant and feasible individuals in the population. A worst case scenario for each objective was devised to act as a reference point of the hypervolume calculation. Table 3 shows these values for the reference point. Since all objectives could tend to zero without the presence of constraints, the objective values were scaled between zero and one, with one being the worst case scenario value for a given objective. This equalizes the prominence of each objective in the hypervolume calculation.

TABLE 3. WORST CASE SCENARIOS FOR EACH OBJECTIVE AND ITS CORRESPONDING VALUE

Objective	Worst Case Scenario	Value
1	All objects are placed on the top deck	$10m$
2	All objects are placed on one deck	1
3	All objects are placed on the edge of the hull	$8.55m$
4	All objects are placed at the front of the ship	$59m$
5	All objects are placed at the LCB	$2.38*10^9 Nm$
6	All objects are spread across the whole deck plane	$2019.51m^2$

RESULTS

This section reviews the results of the four different optimization tests. The following subsections contain results on constraint violations, the hypervolume performance, and some views of generated designs. We used the NSGA-II algorithm implementation through the pymoo package [34].

Constraint Violations

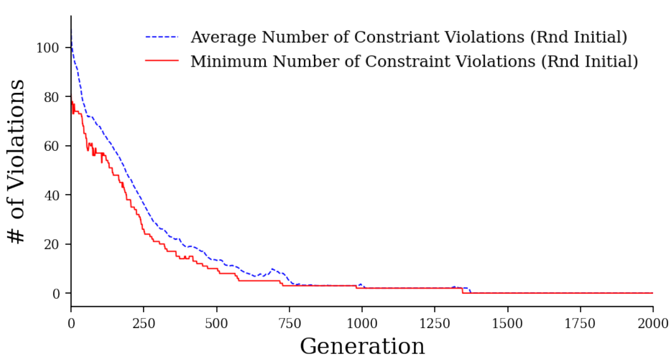

FIGURE 3. AVERAGE AND MINIMUM NUMBER OF CONSTRAINT VIOLATIONS FOR THE RANDOMLY INITIALIZED POPULATION DURING THE BEGINNING OF THE 6-OBJECTIVE, AiO OPTIMIZATION. DUE TO THE HIGHLY CONSTRAINED PROBLEM, THE ALGORITHM NEEDED MORE THAN 1000 GENERATIONS TO ACHIEVE ZERO CONSTRAINT VIOLATIONS.

The only optimization test that produced constraint violations was the AiO optimization with a randomly initialized the population. Figure 3 shows the average number of constraint violations across the population and the minimum number of constraint violations seen in an individual. The graph only shows the first 2000 generations of the optimization since the remainder of the optimization saw zero average constraint violations across the population.

Optimization Results

The hypervolume metric was used to evaluate the performance of the optimization algorithms. Figure 4 the growth of the hypervolume for all four optimization tests. The black line shows the hypervolume for the randomly initialized AiO optimization. The green line is the result of heuristically initialized, AiO optimization. The red line is the hypervolume starting at Generation = 500, for the Bilevel method where the lower algorithm is the 4-objective optimization. Finally, the orange line shows the hypervolume for the Bilevel method where the lower algorithm is the AiO optimization. Table 4 shows the initial and final hypervolume values for each optimization scheme. It is important to note that the bilevel scheme with the four objective lower algorithm produces an 80.3% larger hypervolume than the AiO scheme with the heuristic initialization.

TABLE 4. INITIAL AND FINAL HYPERVOLUME VALUES FOR EACH OPTIMIZATION SCHEME. WE NOTE THAT THE BILEVEL SCHEME ACHIEVES THE HIGHEST HYPERVOLUME WHICH IS 80.3% LARGER THAN AN ALL-IN-ONE OPTIMIZATION SCHEME.

Scheme	Initial HV	Final HV
AiO, Random Init.	0.000	0.072
AiO, Heuristic Init.	0.058	0.132
Bilevel, AiO Lower.	0.144	0.215
Bilevel, 4-Obj. Lower.	0.144	0.238

Individual Design Visualization

This subsection presents a sample of design arrangements generated by the optimization algorithm. These designs were generated utilizing a python script for the Rhinoceros CAD software. The Z scale is tripled to allow visualization of the objects on all four decks. The X and Y position and the length and width of each object is exactly scaled to the footprint of each deck shown in gray. The height of each object is a representa-

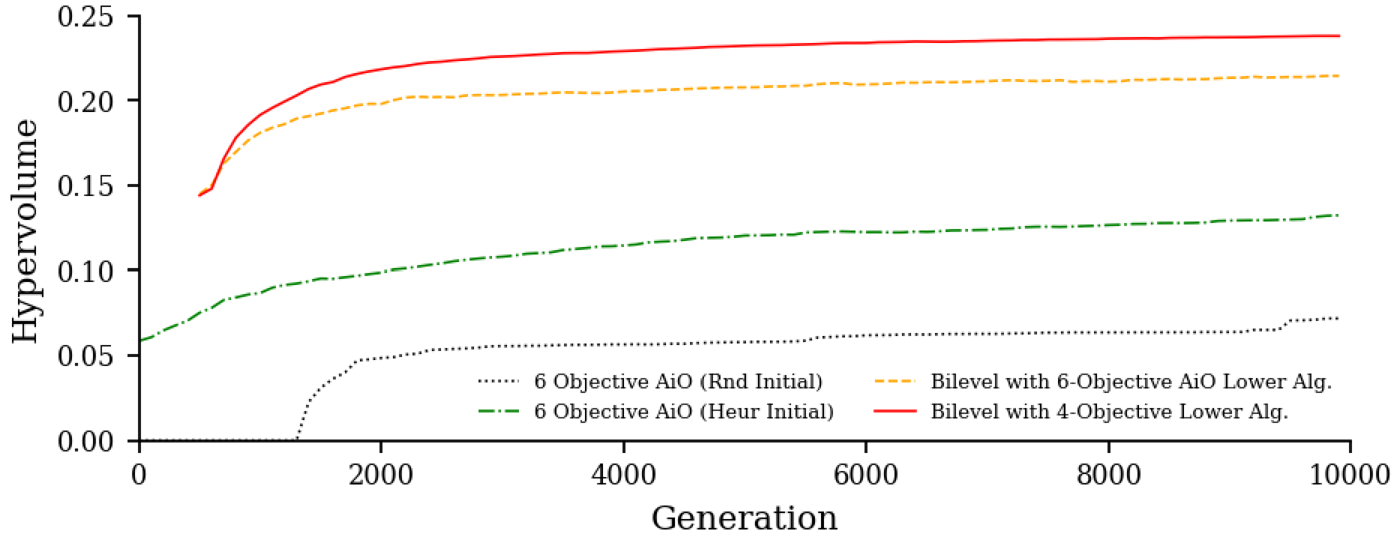

FIGURE 4. HYPERVOLUME OF DOMINANT AND FEASIBLE DESIGN ARRANGEMENTS VERSUS ITERATION FOR THE DIFFERENT OPTIMIZATION SCHEMES.

tive of the object's VCG, and the color signifies the weight of the object. The objects weighing 50.63 tons are blue, 101.27 tons: green, and 151.91 tons: red. Figure 5a shows the optimized design from the bilevel scheme with the 4-objective lower algorithm having the minimum bounding box area of $1194.13m^2$. Figure 5b the design arrangement from the same scheme having the minimum maximum-bending moment of $2.48*10^7Nm$. Finally, Figure 5 shows the design arrangement from the heuristically initialized, AiO optimization scheme having the minimum maximum-bending moment of $5.30*10^7Nm$. The arrangement seen in a) was generated by arranging the objects using a back-left, first fill heuristic [5] to place the set of objects into the ship, filling the bottom most deck first. While this heuristic was able to place all of the objects efficiently in the ship, this arrangement does not utilize all of the available decks equally for the F2 objective. In addition, this arrangement does not account for the F3 or F4 objectives, or YCG and LCG. This heuristically generated design would actually heel the ship to starboard, and trim the ship forward. This lack of consideration of the weight and center of gravity of each object and the aggregated set suggests that packing heuristics alone are not feasible for ship deck arrangements. The optimized arrangements seen in b), c), and d) account for all of the objectives, such that the ship floats with minimal heel and trim in these packing arrangements. The arrangements seen in b) and c) are devised from the bilevel method with the four objective lower algorithm, notice that the objects assigned to the decks are the same between the two figures yet they minimize different objectives. It is interesting to note that the VCG and deck equalization objective have competed by placing the larger objects on higher decks as they account for a larger footprint areas, but with less weight, while placing the smaller objects on the lower decks, fitting more weight lower on the ship. The deck assignments for these objects seem to make sense as these two objectives were specifically targeted in the upper algorithm. Arrangements seen in Figure 5 c) and d) are the individuals with the minimized bending moment objective for the bilevel and AiO optimization schemes, respectively. Notice that the spread of objects resulting from the AiO optimization do not quite follow the same "large objects sit higher" pattern seen from the bilevel optimization.

DISCUSSION

This section provides a discussion on the results of the four optimizations performed for this paper. The first subsection analyzes the heuristic initialization of the population. The second subsection discusses Bilevel Optimization as a tool for population initialization. The third subsection discusses Bilevel Optimization as a tool to separate objectives and improve the optimization results. The hypervolume metric is used to describe the ability of the optimization methods to produce a set of optimal individuals: the greater the final hypervolume, the better the optimization method. This Discussion will not necessarily compare the objective function values of optimal individuals produced by the different optimization methods. Instead, it will focus on hypervolume as a general metric for both the spread and objective function scores of individuals along the Pareto front produced by each optimization method. The final subsection discusses pos-

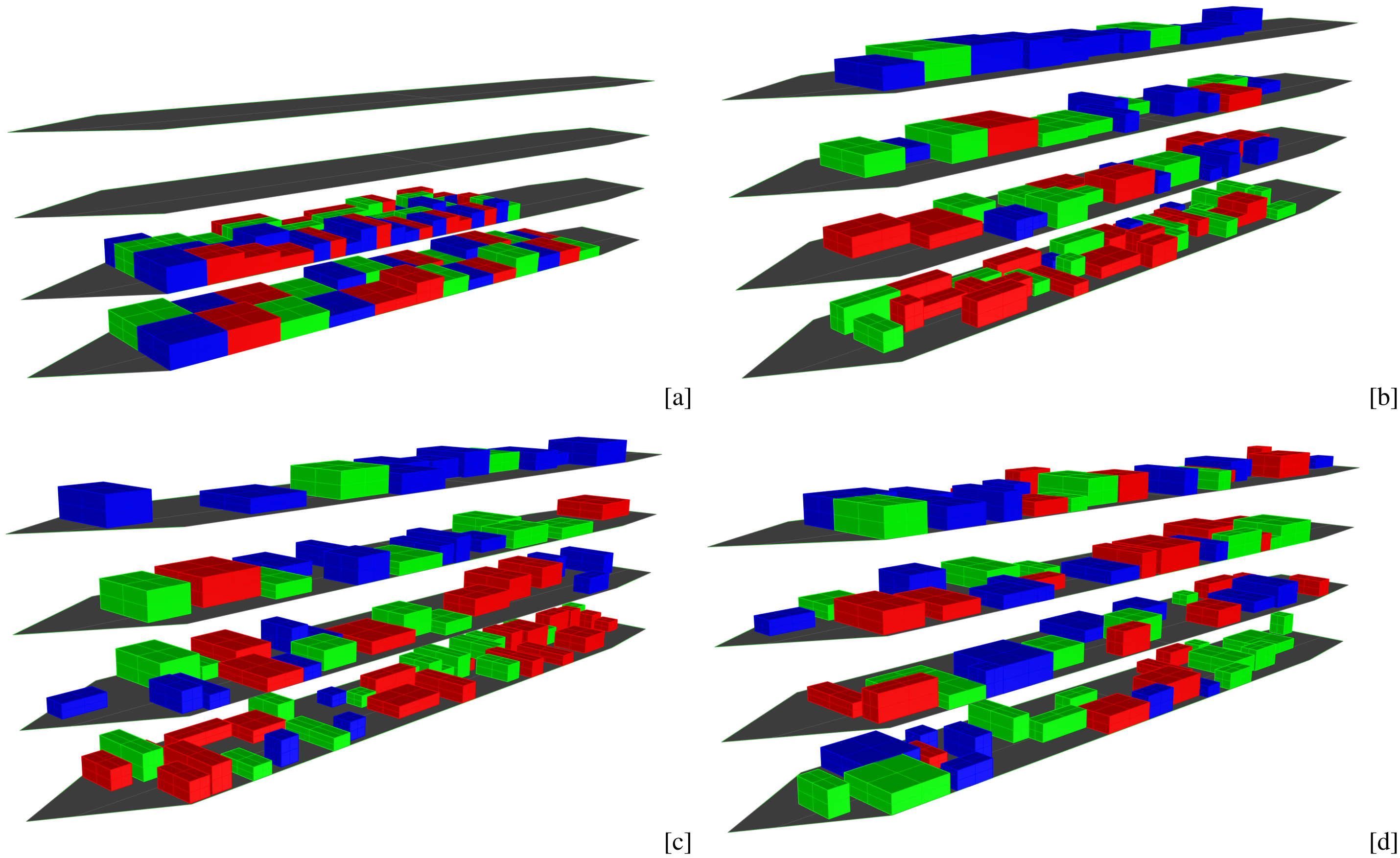

FIGURE 5. a) ARRANGEMENT OF A BEST LONG SIDE FIT HEURISTIC PACKING AS A COMPARISON TO THE OTHER OPTIMIZED ARRANGEMENTS OF OBJECTS b) DESIGN ARRANGEMENT WITH MINIMUM BOUNDING BOX AREA RESULTING FROM BILEVEL SCHEME. b) DESIGN ARRANGEMENT WITH MINIMUM MAXIMUM-BENDING MOMENT RESULTING FROM BILEVEL SCHEME. c) DESIGN ARRANGEMENT WITH MINIMUM MAXIMUM-BENDING MOMENT RESULTING FROM 6-OBJECTIVE, AiO SCHEME.

sible future implementation of a bilevel optimization scheme to sequence ship design.

Heuristic Initialization of Population

The optimizations performed for this paper further confirm that the initialization of a population is imperative to producing a larger hypervolume of optimal and feasible designs. As seen in Figure 4 and Figure 3, the randomly initialized population did not produce any dominant and feasible individuals for the first $1/8^{th}$ of the generations, and required the remainder of the generations to produce a hypervolume that was close to the hypervolume produced by the heuristically initialized population at the start of the optimization process. The heuristic initialization of the AiO optimization scheme allowed the optimization to produce an 83.3% larger hypervolume than the randomly initialized AiO optimization after 10,000 generations, as seen in Table 4. This heuristic intialization algorithm is computationally efficient, placing objects in order N computations; however, it does so without a guarantee that all objects will be placed without any constraint violations due to the presence of multiple decks. In future work, we will test other existing heuristic algorithms that guarantee object placement to our application with the requirement of multiple decks.

Bilevel Optimization to Initialize a Population

The results also indicate that bilevel optimization is an effective tool to seed and heuristically initialize a population. The upper algorithm of the bilevel scheme only manipulated the deck assignment of the 81 objects for 500 generations, then utilized an optimal deck assignment arrangement with the heuristic object placement process to initialize a population for the AiO optimization scheme, which ran for the remaining 9,500 generations. The starting hypervolume for this bilevel optimization was greater than the final hypervolume of the 10,000 generation optimization of the AiO scheme. As seen in Table 4, utilizing the bilevel scheme to initialize the AiO optimization resulted in a 62.9% increase the final hypervolume over the heuristic initialization of the AiO problem. The initial optimization from the upper algorithm pushed the dominant set of individuals much more quickly than trying to optimize all of the objectives simultaneously. Additionally, this upper optimization was performed with less computational effort than the AiO scheme since the optimization algorithm only checked for five constraints, evaluated two objective functions, and manipulated 81 variables per individual, instead of the 3,646 constraints, six objectives, and 324 variables per individual. Further, after the bilevel initialization, the AiO optimization also continued to increase the hypervolume for the remaining 9,500 generations, confirming that there were still opportunities to further optimize these design arrangements after the population initialization.

Bilevel Optimization to Separate Objectives

Bilevel Optimization is shown to be an effective tool for separating objectives to evaluate many objective problems in sequence rather than in parallel. This claim is only true where objectives, constraints, and design variables can be completely decoupled into multiple sets. In this paper, the VCG objective, deck equalization objective, and GM constraint are solely reliant on the deck assignment of the objects, allowing these to be optimized initially and then held constant while the remaining four objectives, 3,645 constraints, and design variables are evaluated and manipulated in the lower algorithm. As seen in Table 4, the sequencing of objectives in a bilevel scheme improved the final hypervolume by 80.3% over the heuristically initialized AiO problem. The benefit to this optimization scheme is it reduces the number of decisions NSGA-II makes in crossover, mutation, design evaluation, allowing for greater design space exploration within the confines of predetermined deck assignment arrangements. This is evident in Figure 4, where the bilevel scheme with the 4 objective lower algorithm produces a greater hypervolume at the end of 10,000 generations than any other optimization scheme proposed in this paper.

Implications of Bilevel Optimization for Ship and Systems Design

Sequencing of decoupled objectives using a bilevel optimization scheme is shown to improve the quality of the hypervolume of the Pareto dominant set of individuals. In this instance, the deck assignment of a set of objects was optimized for deck utilization and VCG, then the positioning of the objects was optimized for ideal weight and spatial distribution. This optimization scheme accounts for the physical implications of arranging objects. Other ship arrangement papers, such as those authored by Wang, Parsons, and Daniels account human factors and the spatial relations between classed objects [23, 29, 30]. Strategically sequencing the optimization for object placement with a bilevel scheme can allow the optimization of a ship on a systems level and subsystems level. In this case, the optimization algorithm will not need to evaluate the complexity of a whole ship design for every individual, or slow optimization progress by leveraging a limited population of designs across a large number of objectives to map a Pareto front. Figure 5c shows the optimized arrangement of the objects that has the minimum longitudinal bending moment applied to a structure that would support this ship. This design was generated using the bilevel scheme, where the deck assignments were preserved after the upper algorithm. Decoupling objectives allowed this individual design to have a low center of gravity, equalized deck usage, and a weight distribution that will reduce the structural needs of this vessel. Similar decoupling of design variables and objectives can design the bow and stern of a ship, as seen in Feng et al.'s recent Paper [28].

More broadly speaking, bilevel schemes can be used to sequence the optimization of systems on the system, subsystem, and component level. Following the model used in this paper, a system level design can be optimized in an upper algorithm, such as the positioning and sizing of components. Then the lower algorithm will optimize the components while constraining the optimization based on the system level optimization, to further improve the system level objective. Potential applications of bilevel optimization for sequencing systems design include automobiles, mobile electronics, medical devices, and factory/workstation design.

CONCLUSION

This paper presented a novel method to solve a many-objective, combinatorial optimization problem for bin packing pertaining to the arrangement of objects across multiple decks of a ship. This problem was heavily constrained to ensure that objects placed do not intersect each other and do not protrude from the ship. Additionally, a heuristic for packing items into a ship was devised following a pattern similar to the back-left 2-D packing algorithm, but to spread the objects across multiple decks. Four different optimization methods were tested using the NSGA-II optimization algorithm. The results showed that population initialization to prevent infeasible designs is imperative to producing a wide variety of feasible designs for further evaluation after the optimization is completed. Further, it was shown that when some of the objective functions and design variables can be decoupled from the remainder of the problem, a bilevel optimization scheme is effective at population initialization, producing an initial population with a larger hypervolume than designs produced from optimization initialized with the heuristic initialization alone. Further, leveraging the same bilevel scheme to optimize the six objectives as a two stage sequence to evaluate two objectives, then the remaining four objectives produces a larger hypervolume at the end of the optimization. This improved performance is likely due to the simplification of the optimization problem. This allows the crossover and mutation operators to produce better results when focusing on fewer objectives than all six objectives evaluated in parallel.

Future work in this project space includes further expansion of bilevel optimization schemes to produce optimized deck arrangements that can be sequenced with structural and/or hull design. Such potential opportunities include bin packing optimization for machinery and/or outfitting to produce initial stage designs work includes investigation into systems optimization utilizing a bilevel scheme to sequence the optimization of different objectives for a wider variety of industries and use cases.

ACKNOWLEDGEMENT

We would like to thank Ashley Margetts for her work to produce the CAD models of the designs seen in Figure 5. We also thank the National Defense Science and Engineering Graduate Fellowship program for supporting Noah Bagazinski's studies while working on this project.

REFERENCES

[1] Fadel, G., and Wiecek, M., 2015. "Packing optimization of free-form objects in engineering design". In *Optimized Packings with Applications*, G. Fasano and J. Pinter, eds., 1 ed., Vol. 2 of *Springer Optimization and Applications*. Springer, Chapter 3, pp. 37–66.

[2] Lodi, A., Martello, S., and Monaci, M., 2002. "Two-dimensional packing problems: A survey". *European journal of operational research,* ***141***(2), pp. 241–252.

[3] Lodi, A., Martello, S., and Vigo, D., 1999. "Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems". *INFORMS Journal on Computing,* ***11***(4), pp. 345–357.

[4] Huang, E., and Korf, R. E., 2013. "Optimal rectangle packing: An absolute placement approach". *Journal of Artificial Intelligence Research,* ***46***, pp. 47–87.

[5] Jylänki, J., 2010. "A thousand ways to pack the bin-a practical approach to two-dimensional rectangle bin packing". *retrived from http://clb. demon. fi/files/RectangleBinPack. pdf.*

[6] B., S., R, W., and Perry, K., 2018. *Greedypacker*, latest ed. GitHub, Los Angeles. See also URL: https://github.com/solomon-b/greedypacker.

[7] Yin, S., and Cagan, J., 2000. "An Extended Pattern Search Algorithm for Three-Dimensional Component Layout ". *Journal of Mechanical Design,* ***122***(1), 01, pp. 102–108.

[8] Berkey, J. O., and Wang, P. Y., 1987. "Two-dimensional finite bin-packing algorithms". *Journal of the Operational Research Society,* ***38***(5), pp. 423–429.

[9] Alvelos, F., Chan, T. M., Vilaça, P., Gomes, T., Silva, E., and de Carvalho, J. M. V., 2009. "Sequence based heuristics for two-dimensional bin packing problems". *Engineering Optimization,* ***41***(8), pp. 773–791.

[10] Huang, W. Q., Li, Y., Li, C. M., and Xu, R. C., 2006. "New heuristics for packing unequal circles into a circular container". *Computers Operations Research,* ***33***(8), pp. 2125–2142.

[11] Denton, S., Kim, J., Torres, M., and Secnot, 2021. *Rectpack*, latest ed. GitHub, Spain. See also URL: https://github.com/secnot/rectpack.

[12] Szykman, S., and Cagan, J., 1997. "Constrained Three-Dimensional Component Layout Using Simulated Annealing". *Journal of Mechanical Design,* ***119***(1), 03, pp. 28–35.

[13] Szykman, S., and Cagan, J., 1995. "A Simulated Annealing-Based Approach to Three-Dimensional Component Packing". *Journal of Mechanical Design,* ***117***(2A), 06, pp. 308–314.

[14] Kennedy, J., and Eberhart, R., 1995. Particle swarm optimization.

[15] Liu, D., Tan, K., Huang, S., Goh, C., and Ho, W., 2008. "On solving multiobjective bin packing problems using evolutionary particle swarm optimization". *European Journal of Operational Research,* ***190***(2), pp. 357–382.

[16] Knight, J. T., Singer, D. J., and Collette, M. D., 2015. "Testing of a spreading mechanism to promote diversity in multi-objective particle swarm optimization". *Optimization and Engineering,* ***16***, June, pp. 279–302.

[17] Knight, J. T., Zahradka, F. T., Singer, D. J., and Collette, M. D., 2014. "Multiobjective Particle Swarm Optimization of a Planing Craft with Uncertainty". *Journal of Ship Production and Design,* ***30***(04), 11, pp. 194–200.

[18] Poli, R., Kennedy, J., and Blackwell, T., 2007. "Particle Swarm Optimization, an Overview". *Swarm Intelligence,* ***1***, pp. 33–57.

[19] Gantovnik, V., Tiwari, S., Fadel, G. M., and Miao, Y., 2006. "Multi-objective vehicle layout optimization".

[20] Phadte, S., 2017. "Vehicle layout optimization using multi-objective genetic algorithms". *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, pp. 1–10.

[21] Miao, Y., Fadel, G. M., and Gantovnik, V., 2008. "Vehicle configuration design with a packing genetic algorithm". *International Journal of Heavy Vehicle Systems,* ***15***, pp. 433–448.

[22] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. "A fast and elitist multiobjective genetic algorithm: Nsga-ii". *IEEE Transactions on Evolutionary Computation,* ***6***(2), pp. 182–197.

[23] Wang, H., and Chen, S., 2021. "An approach to ship deck arrangement optimization problem using an improved multiobjective hybrid genetic algorithm". *Mathematical Problems in Engineering,* ***2021***, August, p. 24.
[24] Tan, B., Ma, H., and Mei, Y., 2017. "A nsga-ii-based approach for service resource allocation in cloud". In 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 2574–2581.
[25] Sihna, A., Malo, P., and Deb, K., 2018. "A review on bilevel optimization: From classical to evolutionary approaches and applications". *IEEE Transactions on Evolutionary Computation,* ***22***(2), pp. 276–295.
[26] Colson, B., Marcotte, P., and Savard, G., 2007. An overview of bilevel optimization.
[27] Dundurand, B., Guarneri, P., Fadel, G., and Wiecek, M., 2014. "Bilevel multiobjective packaging optimization for automotive design". *Structural and Multidisciplinary Optimizaiton,* ***50***, October, pp. 663–682.
[28] Feng, Y., el Moctar, O., and Schellin, T., 2022. "Parametric hull form optimization of containerships for minimum resistance in calm water and in waves". *Journal of Marine Science and Applications*, January.
[29] Daniels, A., and Parsons, M., 2008. "A hybrid agent — genetic algorithm approach to general arrangements". *Ship Technology Research,* ***55***(2), pp. 78–86.
[30] Parsons, M., Chung, H., Nick, E., Daniels, A., Liu, S., and Patel, J., 2008. "Intelligent ship arrangements (isa): a new approach to general arrangement". *Naval Engineers Journal,* ***120***(3), pp. 51–66.
[31] Boulougouris, E. K., Papanikolaou, A. D., Zaraphonitis, G., and NTUA-SDL, 2004. "Optimization of arrangements of ro-ro passenger ships with genetic algorithms". *Ship Technology Research,* ***51***(3), pp. 99–105.
[32] Zubaly, R., 1996. *Applied Naval Architecture*. Cornell Maritime Press.
[33] of Shipping, A. B., 2021. "Hull construction and equipment". *Rules for Building and Classing Marine Vessels*, pp. 51–66.
[34] Blank, J., and Deb, K., 2020. "pymoo: Multi-objective optimization in python". *IEEE Access,* ***8***, pp. 89497–89509.
[35] Zitzler, E., Brockhoff, D., and Thiele, L., 2007. "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration". In Evolutionary Multi-Criterion Optimization, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, eds., Springer Berlin Heidelberg, pp. 862–876.