SHIPS AND OFFSHORE STRUCTURES
https://doi.org/10.1080/17445302.2021.1926142

Multi-objective optimisation as an early design tool for smart ship internal arrangement

Panagiotis Louvros [a], Evangelos Boulougouris [a], Andrea Coraddu [b], Dracos Vassalos [a] and Gerasimos Theotokatos [a]

[a]Maritime Safety Research Centre, Department of Naval Architecture and Marine Engineering, University of Strathclyde, Glasgow, UK; [b]Department of Naval Architecture and Marine Engineering, University of Strathclyde, Glasgow, UK

ABSTRACT
Modern optimisation methodologies have revolutionised the engineering sector and pave the way for innovation. In ship design, this has been spearheaded by the introduction of the Holistic Design approach that allows more attributes and performances of the end product to be assessed accurately and concurrently even at the early design phase. In this respect, the authors present herein a methodology based on the optimisation of the functionalities that the general arrangement needs to provide (in this case; adjacency, noise and evacuation flow). The methodology allows for a large optimisation space, several objectives and intrinsic control by the user at all the stages. By re-arranging the location of the spaces onboard, in relation to their position on the hull as well as between themselves, significant improvements can be achieved. More functional objectives can be incorporated through modularity and penalty functions, keeping the overall process simple and flexible to adapt to fluid early design requirements.

ARTICLE HISTORY
Received 3 December 2019
Accepted 22 April 2021

KEYWORDS
Genetic algorithm; general arrangements optimisation; ship design; concept design

1. Introduction

The backbone for a good design is an ergonomic, adaptable and fit-for-purpose general arrangement. This will allow the successful completion of its mission whether it is commercial or warship, cargo or passenger ship. Its importance, but also the challenges for achieving it, increase with the complexity of the vessel's type. The vast computational power available nowadays, coupled with the improved accuracy and efficiency of today's performance prediction and simulation models and the available optimisation methodologies, enable the designers and decision-makers to explore the available multi-dimensional design space in an unprecedented way. Their arsenal is equipped with a plethora of methods, ranging from extremely fast and simple to very sophisticated, computationally intensive but highly accurate; have for the first time enabled the designers to implement a really holistic design optimisation (Papanikolaou 2010). This enables the decision-maker to assess the impact of the preferences expressed and trade-off between the various objectives (Boulougouris et al. 2004; Nikolopoulos and Boulougouris 2018; Priftis et al. 2018). As all decisions taken are strongly or loosely linked with the allocation of the spaces onboard the ship, the optimisation of the arrangement has the potential to improve substantially the design of the vessel.

The development of such a methodology should address properly the multi-disciplinary character of the problem and not just the topology. This suggests that its mathematical formulation is not that straightforward and may require significant computational power in order to converge to solutions which will be convincing even for the experienced professional. Following a number of attempts dating back to the 1960s Nowacki (2003, 2010), this paper will present such a methodology and provide results that depict the capabilities offered and set the stage for further development.

The general arrangement is one of the characteristics of the design incurring significant additional costs to any changes as the design progresses, as it is typical with all complex products (Walter et al. 2017). This suggests that eventually, the initial decisions affect greatly the final outcome. Therefore, the proposed method focuses on the early design stage, where unfortunately, there is limited info about the structural arrangement or other details of the design. However, this provides additional flexibility and allows the investigation of various options that can be refined either at a later stage by the designer or used as input for other models in the holistic optimisation framework (Boulougouris and Papanikolaou 2006).

Ships with significant complexity and a large number of rooms, serving different functions, such as naval and passenger ships, are those that provide the most challenging cases but also those where the benefits from such an optimisation will be greater (Yrjänäinen et al., 2019).

The convolution of the problem suggests that a proper reduction of its mathematical complexity is required, both in terms of input variables and objectives. The resulting design space should be investigated with an optimisation algorithm capable of converging quickly in order to reduce the computational cost. Therefore, it should be properly designed and adapted according to coding practices and with the available hardware limitations. Prior attempts to solve this problem provide useful insight and a benchmark base.

The authors present herein a modular, 'smart', flexible, fast and sufficiently accurate method for the early stages of the design where information is scarce; the alternatives are many, but their 'cost' is yet unknown. Although there have been several approaches on

CONTACT Panagiotis Louvros panagiotis.louvros@strath.ac.uk Maritime Safety Research Centre, Department of Naval Architecture and Marine Engineering, University of Strathclyde, Henry Dyer Building, 100 Montrose Street, Glasgow G4 0LZ, UK

assessing quickly design characteristics such as weight estimation, stability, powering, etc., there have been very few addressing the internal arrangement. The present method can lead to a better informed and thus more efficient and innovative early design process (Erikstad 2019).

2. Material and methods

2.1. Facility layout problems

The layout problem has been a sector of extensive research in architecture in the so-called Static Facility Layout Problems (SFLPs) (Hillier 1963; Hillier and Connors 1966). These are basically layout optimisation problems for factories or other buildings. The primary objective is to optimise the flow of resources (materials, people, customers etc.) between the different spaces of a facility.

Islier's (1998) algorithm is often cited in FLP studies as it includes a comprehensive description of the required formulation of the layout optimisation problem using a genetic algorithm. Most importantly, it introduces a method for mapping the layout into chromosomes (Figure 1).

The method introduces descriptions for all the operators required for applying this evolutionary method, i.e. creation, crossover and mutation processes, are also included as well as a number of objective functions, useful for implementing FLP.

Lee et al. (2002) attempt to adapt the general FLP logic to the design of a naval vessel. They point out the need to make special provisions for the passages/corridors which are not found in most FLPs. Their optimisation goals of minimising transport cost and increasing adjacency and the mathematical formulation is straightforward. Particularly interesting is the encoding/decoding process where a segmented chromosome is used to include room areas and passage locations. Graphs are used to calculate real distances between rooms (physical paths). The algorithm simultaneously defines room geometry, topology and passage location whilst also satisfying the area requirements.

Hasda et al. (2017) describe a very similar problem with adjacency and transport cost using simplified Euclidean distance rather than graphs and using a chromosome containing x, y coordinates to place rooms (decode). Zhou et al. (2006) also have a similar approach in terms of formulation of objective functions and general structure.

Parsons et al. (2008) have introduced, in the probably most complete and advanced relevant study, the Intelligent Ship Arrangements algorithm (ISA). It attempts to holistically approach the arrangement problem by utilising existing databases developed for the US Navy. Their approach is inclusive, manages to tackle complex geometries and shows good results (Nick 2008). Useful insights can also be found in Nick et al. (2006) and Daniels and Parsons (2006).

Optimisation generally and genetic algorithms are no stranger to general arrangements and layout planning for ships. Multiple studies have been done in an effort to increase operability, stability

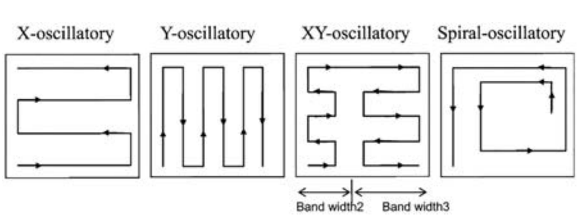

Figure 1 . Oscillatory placing procedure (Islier 1998). (This figure is available in colour online.)

or survivability of primarily naval vessels (Boulougouris and Papanikolaou 2004; Jung et al. 2018), some also using explicitly genetic algorithms for maximising damage stability as well as economic viability of Ro-Ro vessels (Boulougouris and Papanikolaou 2004; Ölçer et al. 2006). Lastly, Lee et al. (2002) utilise them for minimising transport costs and maximising adjacency of the rooms for a naval vessel.

2.2. Genetic algorithms in literature

Genetic algorithms are commonly used in problems such as they are 'operationally simple', and suitable 'for large space and black box problems' (Kubalik et al. 2002) with powerful search capability (Jo and Gero 1998; Damski and Gero 1997). The genetic algorithm firstly introduced by John Holland and later his student, Goldberg (1989), is easy to adapt to a large variety of problems with little effort, searching effectively and avoiding, ideally, any local optima. However, without guidance and within a large search space, the algorithm may be computationally and practically untenable.

Lee et al. (2002) for solving the layout problem in the architectural sector use an oscillating order of stating the rooms as shown below and segments of the chromosome carrying areas of compartments and locations of aisles and passages (Figure 2).

Another similar approach is used by Hu and Wang (2004). They propose a different segmented chromosome with even more data. The researchers do not provide a specific sweeping scheme but evaluate the performance of each one. Both options show the need for a nonlinear ordering system and the inclusion of more information.

2.3. Objective functions

Several merit functions have been proposed by researchers, depending on the design problem at hand. The merit function used in this study is the weighted sum of a number of individual objective function values as shown below:

$$\min F = w_1 \times F_1 + w_2 \times F_2 + \ldots + w_n \times F_n \tag{1}$$

where $w_1, w_2, \ldots, w_n$ are the decision-maker's relative weighting factors. The objective functions in this problem are cost functions and they are generally described by a factor multiplied by the distance. The goal of the algorithm is to decrease the sum of all these cost functions.

The distance is the Manhattan distance between the centroids except for the sound propagation. The Manhattan distance is given in Equation (2). For the sound propagation, the Euclidean distance is used, calculated by the Pythagorean Theorem (see Equation (3) and Figure 3).

$$\text{Manhattan distance} = d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{2}$$

$$\text{Euclidean distance} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3}$$

2.3.1. Adjacency

Room adjacency is the preference for specific room types to be located closer or further apart from others. This is an attribute that is directly linked to the vessel's functionality and eventually its capacity to execute her mission efficiently. The movement of personnel is one the key areas of whole-ship usability that can be improved if it is assessed early in the design process' (Andrews et al. 2007). Adjacency uses as input the distance between two rooms and the preference, indicated by a number that quantifies

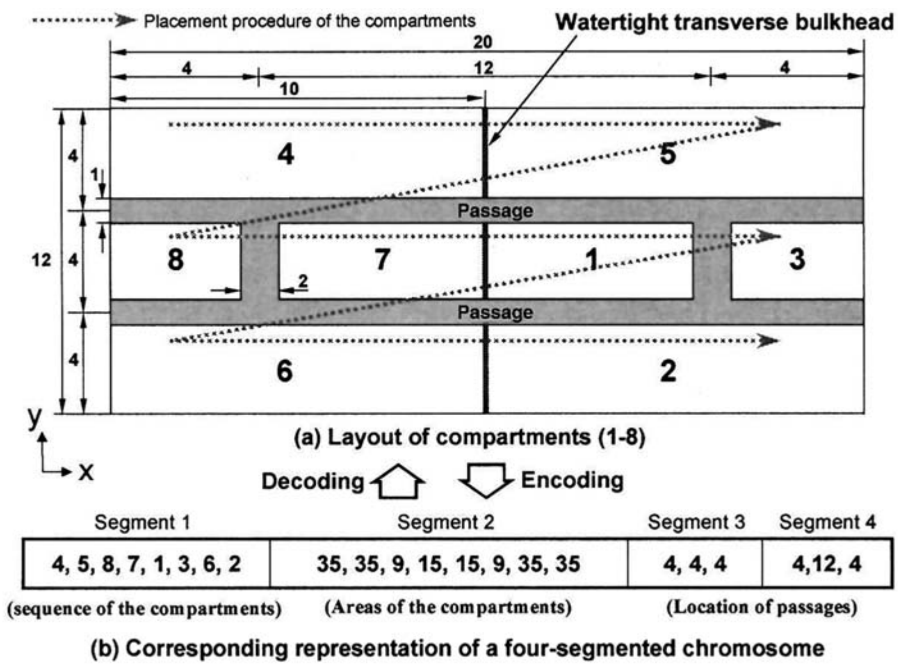

Figure 2. Example of the use of a segmented chromosome (Lee et al. 2002). (This figure is available in colour online.)

how close we want them to be:

$$F_1 = \sum_{i=1}^{N} \sum_{j=i+1}^{N-1} d_{ij} \times \mathrm{adj}_{ji} \tag{4}$$

where N is the number of rooms, i and j are indices for the room numbers, $i = 1 \ldots N$, $j = 1 \ldots N-1$, d_{ij} is the distance between room i and j and adj_{ij} is the corresponding preference factor.

2.3.2. Sound pollution propagation

Some rooms can be sources of noise depending on their function, e.g. the engine room. Especially on passenger ships, noise pollution is directly linked to passenger comfort and on warships with interference to the acoustic devices. Sound is modelled by assigning a sound production value to rooms that attenuates according to the inverse square law of sound intensity, I

$$I \propto \frac{1}{r^2} \tag{5}$$

where r is the distance separating the sound source and the specific point

$$F_2 = \sum_{i=1}^{N} \sum_{j=1, \neq i}^{N} \frac{1}{r_{ij}^2} \times S_j \tag{6}$$

where r_{ij} is the distance between room i and j and $S_j$is the sound production value of room j. Note that calculation for $i = j$ is not done. The rooms that are sensitive to sound are set by the user with Boolean operators.

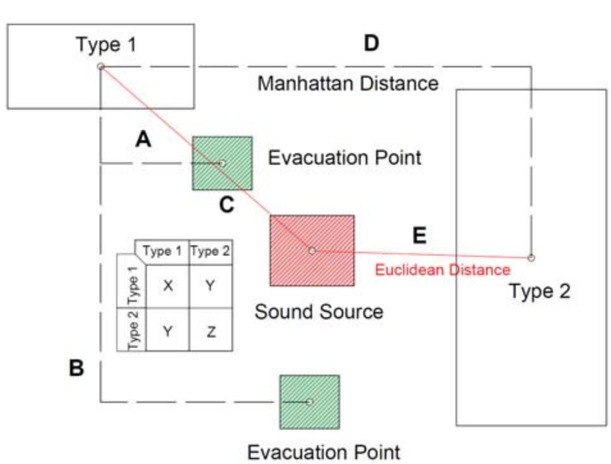

Figure 3. Explanation of objective functions. (This figure is available in colour online.)

2.3.3. Evacuation flow

This simple consideration of the evacuation scenario considers how many occupants need to evacuate each room and how far the exit of the space is; called evacuation points. The evacuation points can in reality be a stairway leading to a muster station or an exit to a space that needs to be traversed to reach the muster station or deck where embarkation to life rafts/rescue boats takes place. For each room, the value is calculated to the nearest evacuation point

$$F_3 = \sum_{i=1}^{N} d_{ie} \times \mathrm{NoOcc}_i \tag{7}$$

where d_{ie} is the distance of a room to the nearest evacuation point. NoOcc_i is the user-defined number of occupants of each room.

2.3.4. Chromosome

The proper definition of the chromosome structure is very important. The chromosome is used by the fitness function and the optimisation process is occurring on the data contained therein. The chromosome structure is explained here in Figure 4.

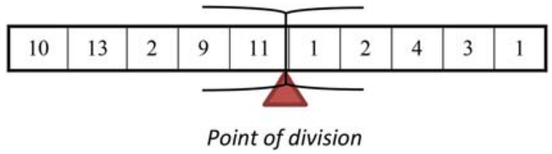

Figure 4. Example of x-segmented chromosome. (This figure is available in colour online.)

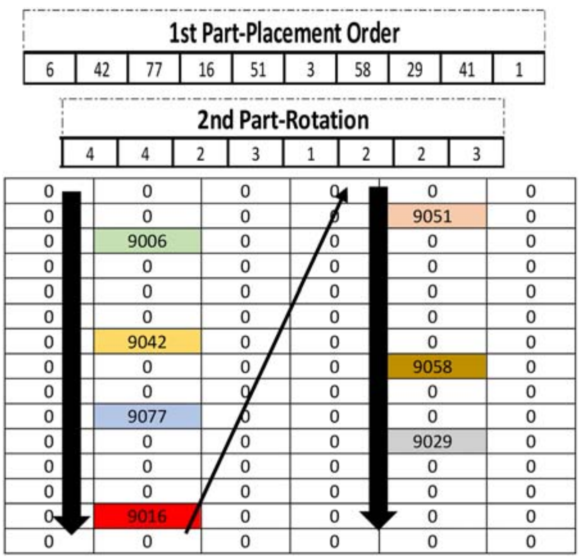

Figure 5. Placement example with space matrix. (This figure is available in colour online.)

Chromosome, $\boldsymbol{x}$ represents a row vector segmented into two parts. The segmented chromosome will serve the purpose of allowing multiple data to be associated with each individual.

Therefore, the length of the chromosome is not the number of individuals (variables in MatLab jargon, see Mathworks 2019). The number of variables, NoRooms, is the total number of rooms of any type that are specified by the user. The chromosome is divided as follows:

- From 1 to NoRooms,

The placement order sequence; contains the unique index number of each room. The order of listing encapsulates the location information (see also Figure 5).

- From NoRooms + 1 to NoRooms×2,

The numbers associated with the cells in the first half of the chromosome allow for clockwise rotation of the rooms from their user definition, as given in Table 1:

Table 1. Rotation information coding

Cell value	Result in rotation
1	0°
2	+90°
3	+180°
4	+270°

Explanation of the fitness calculation for demo case in Figure 3 follows:

$$\begin{aligned} &\text{Fitness}_{\text{Type1}} = w_1 \times Y \times D \\ &+ w_2 \times \frac{\text{Soundvalue}}{C^2} \\ &+ w_3 \times A \times \text{No}_{\text{occupants}}\text{Type1} \end{aligned} \tag{8}$$

where A is the Manhattan distance from Type 1 to evacuation point, B, similarly for Type 1 to another evacuation point, D is the Manhattan distance between the two room types while C and E the Euclidean distances from the sound source.X, Y, Z are adjacency preferences numbers.

3. Methodology

The deck layout is described by a user-defined rectangular matrix. It adapts to the hull's shape by 'blocking' the corresponding cells.

The rules for assigning values to the cells are as follows (Table 2):

1. If larger than 0, then the cell is part of a room of type equal to the number inserted, e.g. if the cell contains 8 then the area is part of room type 8.
2. If less than 0 then the space is unavailable for allocating any value and should be avoided. The negative values can also vary depending if they are user input, stationary rooms etc.
3. If the number is larger than 9000 then the cell is designated as a centroid of a room. For example, if the value is 9008 then the cell is the centroid of a room of type 8.
4. Zero values signify unallocated space. In the beginning, the matrix will be filled with zeros.

This is a straightforward approach as the process of entering numbers to specify rooms is common in the literature (Lee et al. 2002; Hu and Wang 2004). By using the appropriate value, we can pass more information that describe the room, e.g. entrances, exits etc.

3.1. Decode function

The decode function is the link between the row vector of data (chromosome) and the layout in the matrix. This process incorporates not only the placement of the rooms but also the logical tests for ensuring that the constraints are met. For example, the decoding function checks the status of the area it is about to write in. If it is occupied by other rooms, corridors or user-defined shapes then there is a 'collision'. To avoid collisions and resolve them, the function will move to a new empty space to put the next room. Note that the decode function utilises heuristics to limit uneccesary logical processes that are the source of the largest computational effort.

Below is an example of how the placement algorithm works (see Table 2). In the matrix, only the centroids of the rooms are depicted (Figure 5). However, the rooms are created around them. Thus, the gaps between the centroids.

Table 2. Example of 2D space matrix

−10	−10	−10	−10	−10	−10	−10	−10	−10
1	1	1	3	3	0	−10	−10	−10
1	9001	1	3	9003	2	2	−10	−10
1	1	1	3	3	9002	2	0	−10
2	2	3	3	3	0	0	−10	−10
9002	2	3	9003	3	0	−10	−10	−10
−10	−10	−10	−10	−10	−10	−10	−10	−10

3.2. Data structures

The algorithm must be able to access user-defined matrices to access all the data characteristic for the rooms. These are:

1. Room type matrix,
2. Sound sensitivity,
3. User input for pre-registered data in space matrix,
4. Preference adjacency type matrix,
5. Preference adjacency type matrix for user input,
6. Definition of room layouts (Length × breadth and Centroid location),
7. Evacuations points,
8. Passenger number,
9. Sound production,
10. (Optional) Corridor rows,
11. (Optional) Corridor columns.

All the data stored in matrices. An adjacency matrix will contain the intended distances of each room from other rooms with a value. For example, see Table 3.

The values will quantify if a room should be close to another or not. In this simple example, the cabins should be close to an evacuation point ($\mathrm{adj}_{ij} = 4$) but far from the noise of machinery ($\mathrm{adj}_{ij} = -1$). The range of numbers may vary as can be seen from the literature, where there exists a variety of schemes (Islier 1998; Lee et al. 2002; Nick 2008; Tanaka et al.; Daniels et al. 2010). The distance of a room from another is multiplied by the factor in the adjacency matrix, so a high factor amplifies the result and makes the contribution towards the total fitness value larger, thus pushing the algorithm to find a better solution.

Similarly, the rest of the matrices hold values like whether the sound should be minimised for a room, the sound produced in a room, the points of evacuation (coordinates), the occupants of each room etc. Examples can be seen in the following tests (Figures 6–11). See flowchart of the algorithm below (Figure 12).

3.3. Weighting factors and selection

As seen in Equation (8), the three objectives are aggregated in a single value with the use of a penalty function.

The values of the weighting factors will, therefore, greatly influence the optimisation result. The exact values and balance of the factors are mostly a trial and error process. The effects of the weighting factors are obvious in the optimisation result; thus, it is trivial for the designer to make changes. If cabins are very close to sound-producing rooms then it is evident that either one of the other objectives is prioritised or that the sound objective needs a larger weighting factor. The above operation necessitates the designer to judge an arrangement and try some iterations before deciding on the mix of weighting factors.

4. Results

A major challenge in developing this algorithm is the difficulty to independently verify the results. It is meaningless to compare the individual iterations between algorithms. However, the rational way to benchmark the methodology is to use simple tests and evaluate the results using maths and common sense. Afterwards, the optimum arrangements produced by different algorithms can be compared using the same objective. The first approach is used for the first test which verifies that the code works properly.

Table 3. Example of a preferred adjacency type criteria

Adjacency preference	Cabin	Exit	Machinery space
Cabin	3	4	−1
Exit	4	1	2
Machinery space	−1	2	3

4.1. Simple evaluation tests

For these tests, a rectangular area 60×60 is used. For the simple tests, three room types are defined. Cabin, Entertainment area and Machinery. The objective is to set simple relations and verify the result. To this effect, the rooms have the following attributes:

1. Cabins are sensitive to sound, contain four occupants and should be close to other cabins and entertainment but far from machinery spaces.
2. Entertainment rooms should be close to cabins and other entertainment areas but also far from machinery. However, they are not sensitive to sound and they are less affected by adjacency to Machinery.

The above is quantified in the adjacency matrix and in the other matrices seen (Tables 4–6).

The evacuation point is placed on the top left corner. The matrix below gives the number of each of the room types (Table 7).

A designer would arrange the rooms with the following mind-set:

1. Keep all the room types close to each other.
2. Put the Cabins (1) and Entertainment rooms (2) close to the evacuation point (Top left corner).
3. Move the machinery rooms far from the Cabins but closer to the Entertainment rooms. Actually, the entertainment rooms will be between the cabins and machinery rooms to increase the distance between them.

Table 8 provides the legend to read the following plots.

Table 4. Adjacency matrix in Test #1

Adjacency Preference	Cabin	Entertainment	Machinery
Cabin	5	4	1
Entertainment	4	3	2
Machinery	1	2	3

Table 5. Sound production matrix in Test #1

Cabin	0
Entertainment	30
Machinery	100

Table 6. Occupants of each room in Test #1

Cabin	4
Entertainment	1
Machinery	0

Table 7. Room number for each type in Test #1

Cabin	80
Entertainment	15
Machinery	20

Table 8. Legend for layout plots

Cabin	1 (Red)
Entertainment	2 (Blue)
Machinery	3 (Grey)
Centroids of rooms (black cell in middle of room)	9001 (Black)
Evacuation point	−10 (Green)

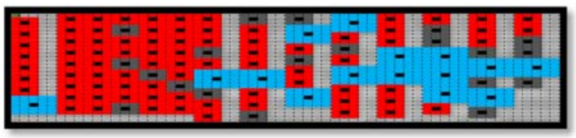

Figure 6. Best layout at 11 generations. (This figure is available in colour online.)

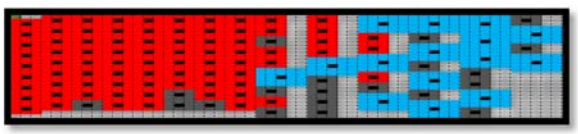

Figure 7. Best layout at 74 generations. (This figure is available in colour online.)

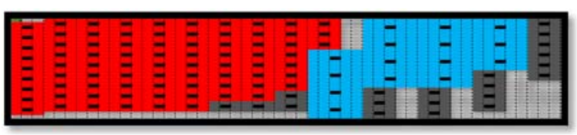

Figure 8. Final result of the algorithm. (This figure is available in colour online.)

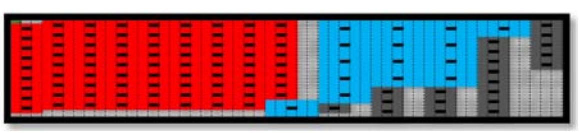

Figure 9. Algorithm result in Test #2. (This figure is available in colour online.)

The best solution was achieved after 683 generations with a population of 300 each (Figures 6–8). The elapsed time was 178s.

In a second test, the same adjacency matrix is used, but a population of 600 is used and the sound production values change (see Table 9). We expect a greater drive to separate the noisy machinery spaces from the cabins.

Solution is achieved after 314s for 612 generations (200 stall generations) (Figure 9).

It is evident that the changes introduced in the second test improved the adherence to the logic at the expense of time (almost double the time to achieve the solution). The large number of stall generations, generations where no improvement was found (see Mathworks 2019), guarantees that the solution is practically an optimum. The number of stall generations is a way of examining convergence and is decided by experience; one-third of the total generations seems to be more than enough.

As with all evolutionary algorithms, the generation size and the number of generations may influence the convergence to the global optimum. The examples shown herein are indicative of results produced when the algorithm runs for ample time and with a large number of stall generations before the algorithm is terminated.

Table 9. Sound production of rooms in Test #2

Cabin	0
Entertainment	20
Machinery	300

The layouts follow the logical approach that was theorised above and provide a convincing result. The balance of the weighting factors greatly affects the result. If one of the objectives is amplified, the layout will be skewed to satisfy it over the other ones. It is observed that the use of the wrong weighting factor balance may prohibit the algorithm from locating the global optimum (as expected from the user).

4.1.1. Introducing obstacles

Below is the result of the second test's data but with the light beige coloured cells being defined by the user as inaccessible. The solution is achieved after 409 generations (200 stall generations) and 174s (Figures 10 and 11).

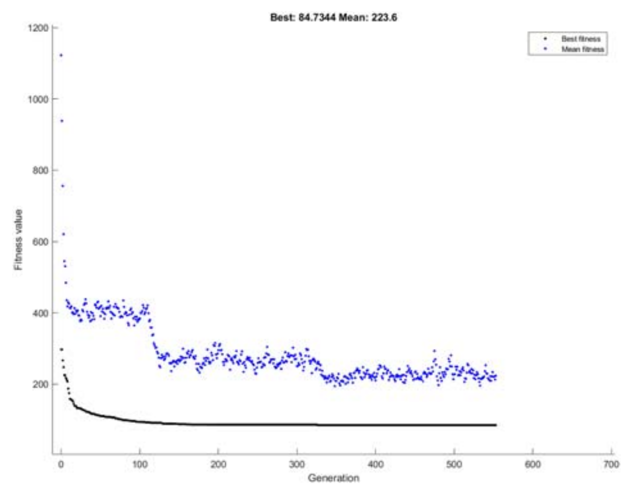

Figure 10. Final solution for test with user entered obstacle. (This figure is available in colour online.)

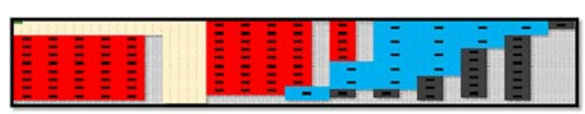

Figure 11. Fitness-generation history. (This figure is available in colour online.)

4.2. Cruise ship lower decks case study

For a more realistic case, the algorithm is applied to a cruise ship general arrangement (Pasia et al. 2019). The goal is to optimise the layout of the lower decks while keeping fixed the main and auxiliary machinery spaces. In the particular design, more than nine different types or rooms are located below the main deck, including stores, cabins and messes. In total, they cover the area available at the original deck plan based on the hull's 3D model. In Table 10, the minimum area for the various spaces is given.

The dimensions of the rooms are largely arbitrary except for the cabins. Large rooms are split into smaller pieces that are held close by large adjacency preferences (Crew mess, hospital) (Table 11).

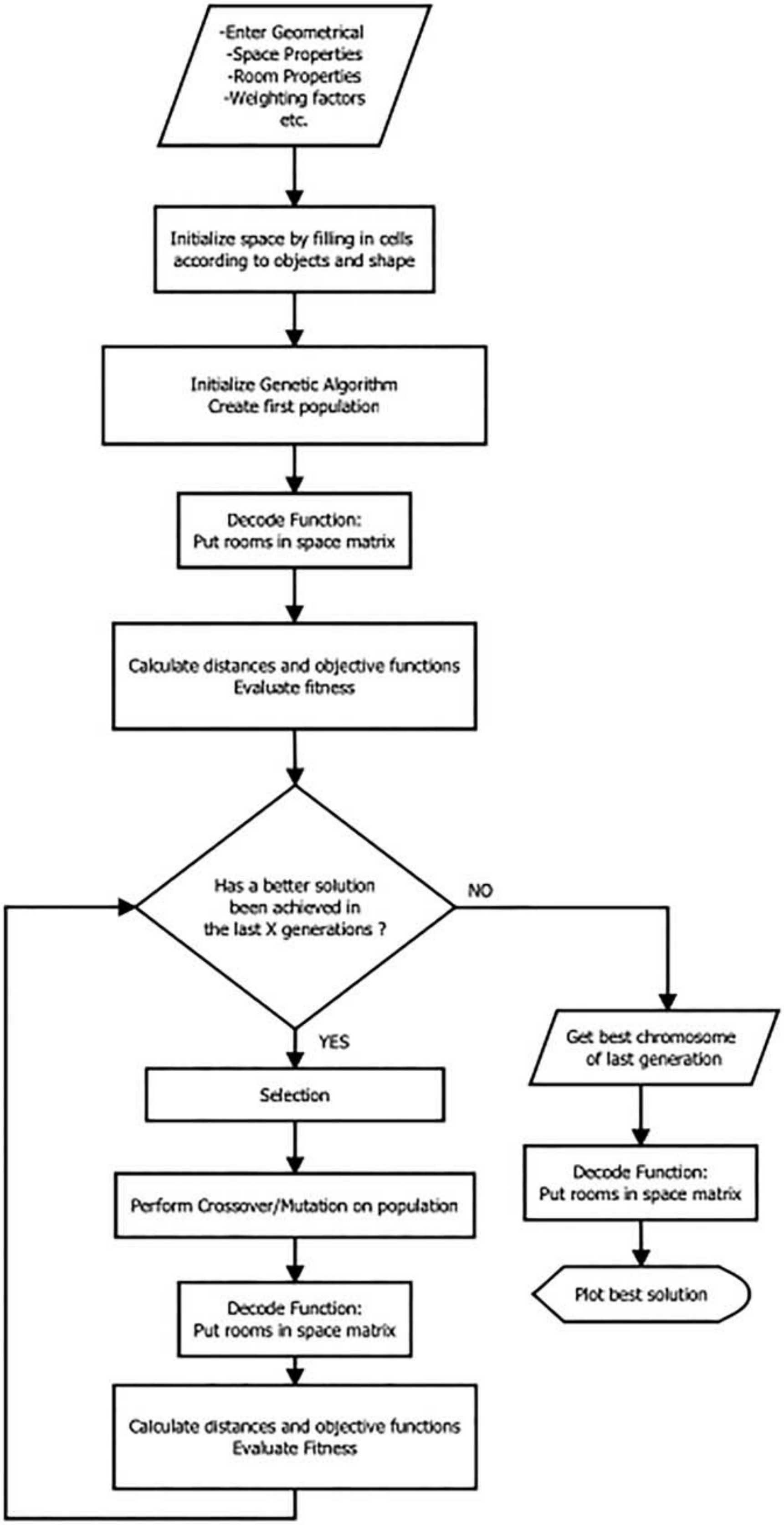

Figure 12. Flowchart of the algorithm. (This figure is available in colour online.)

Table 10. Actual cumulative area of rooms vs. required

Room type	Area (m^2)	Total Area (m^2)	Required Area (m^2)	Outcome
Crew cabin	16	4240	N/A	PASS
Galley	136	408	385	PASS
Hospital	300	600	600	PASS
Crew mess	340	680	671	PASS
Officer's mess	60	60	58.5	PASS
AC room	24	312	300	PASS
Linen room	96	1920	100	PASS
Store	50	1900	1857	PASS
Laundry room	50	400	400	PASS

Table 11. Room-related data

Room type	*L* (m)	*B* (m)	Occupants	Number
Crew cabin	4	4	3	220
Galley	17	8	30	3
Hospital	15	20	20	2
Crew mess	17	20	30	2
Officer's mess	6	10	10	1
AC room	4	6	0	13
Linen room	12	8	0	20
Store	10	5	0	38
Laundry room	10	5	1	8

Table 12. Sound-related data

Sound pollution	Sound non-dim	Sound concern yes or no (1/0)
Crew cabin	0	1
Galley	40	1
Hospital	0	1
Crew mess	10	1
Officer's mess	40	1
AC room	70	0
Linen room	0	0
Store	0	0
Laundry room	0	0

The noise from the engine room spaces is modelled with a value of 500, gradually diminishing with the transverse bulkhead separation, placed around the casing and aft of each deck to correspond with the actual machinery present there (Table 12). The casing is entered as a fixed space (orange box in Figure 13). Table 13 depicts the adjacency preferences.

The evacuation points are set at the stairways leading to the upper decks where the mustering and evacuation takes place. There are four such staircases given by the preliminary general arrangement. One is located at the aft, one at the aft end of the casing, one at the fore end of the casing and the last one well forward (seen in red).

Below are the best and final results used for the general arrangements (Figures 14–23). The optimisation result is the raw output while the following arrangement resulted in some minor fine-tuning of the algorithm solution by a human designer. See Figure 24 for legend.

Figure 13. Illustration of casing and evacuation points. (This figure is available in colour online.)

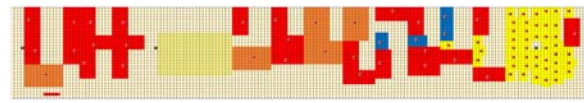

Figure 14. Deck 2 optimisation result. (This figure is available in colour online.)

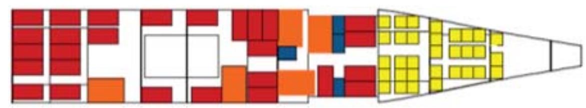

Figure 15. Deck 2. (This figure is available in colour online.)

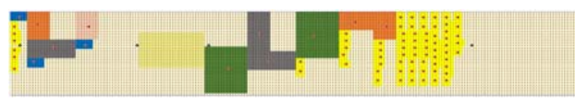

Figure 16. Deck 3 optimisation result. (This figure is available in colour online.)

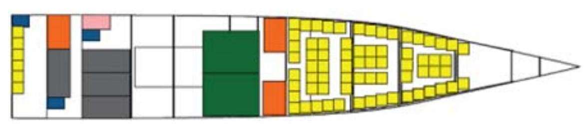

Figure 17. Deck 3. (This figure is available in colour online.)

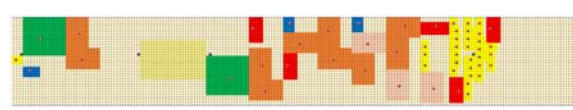

Figure 18. Deck 4 optimisation result. (This figure is available in colour online.)

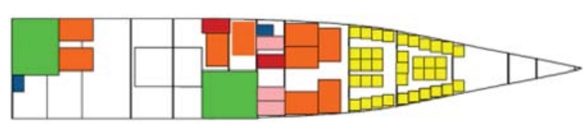

Figure 19. Deck 4. (This figure is available in colour online.)

Table 13. Adjacency preferences for cruise ship rooms

Adjacency	Crew cabin	Galley	Hospital	Crew mess	Officer mess	AC room	Linen room	Store	Laundry room
Crew cabin	5	2	0	2	0	1	1	0	2
Galley	2	3	0	3	0	2	0	4	0
Hospital	0	0	5	0	0	0	0	0	1
Crew mess	2	3	0	5	0	0	0	0	0
Officer mess	0	0	0	0	0	0	0	0	0
AC room	1	2	2	1	1	−1	0	0	0
Linen room	1	4	0	0	0	0	2	0	4
Store	0	4	0	0	0	0	0	0	0
Laundry room	2	0	1	0	0	0	4	0	0

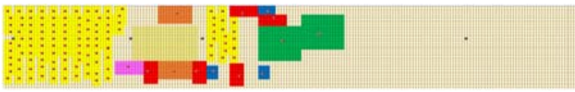

Figure 20. Deck 5 optimisation result. (This figure is available in colour online.)

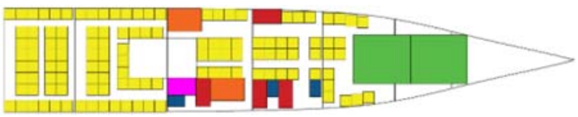

Figure 21. Deck 5. (This figure is available in colour online.)

Figure 22. Actual GA of Deck 5. (This figure is available in colour online.)

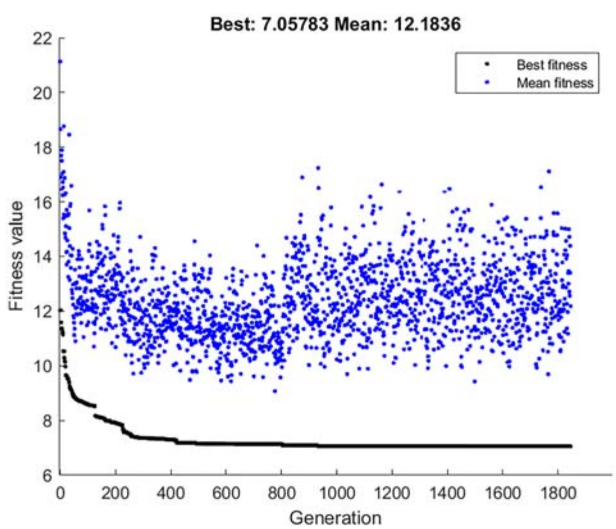

Figure 23. Fitness-generation history of a deck optimisation. (This figure is available in colour online.)

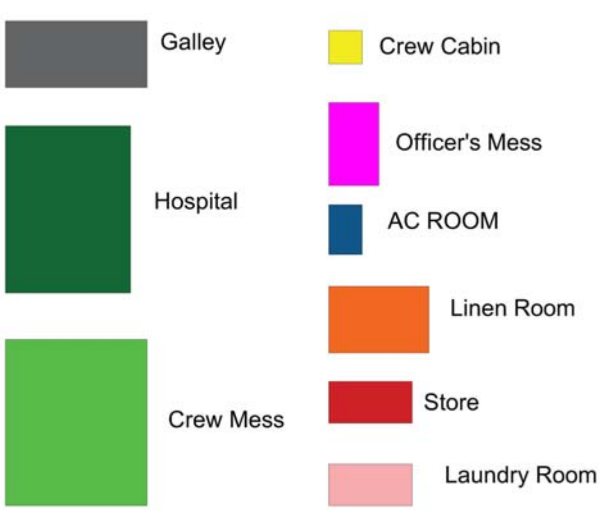

Figure 24. Room footprints. *Casing space left blank in GA of decks as is not defined by algorithm. (This figure is available in colour online.)

5. Discussion

5.1. Interpretation of findings

The results seen above are representative of the capabilities of the algorithm. It gave promising results for the simple case while the more realistic test had some inaccuracies that were addressed. The large number of rooms of small relative size and square dimensions appear to allow a more 'fluid' and smooth transition between the rooms.

It was also apparent that the numbers entered for the weighting factors greatly changed the layout. Care needs to be taken to ensure that the balance represented reflects accurately the user preferences. These weighting factors allow great flexibility to the user to set the objective priorities and shape the layout according to his/her preference.

The application for the cruise ship was representative to a real case scenario with time constraints and the target was to get a result as good as designer-produced preliminary general arrangement (Figure 25).

Adherence to expected logic is evident in Figure 26; cabins (yellow) are far from sound from casing (dark red), messes (green) and cabins that contain most of the passengers/crew are also placed close to evacuation points. AC (blue), galleys (grey), laundry rooms (pink) that produce moderate noise are also kept far from the most populated areas. Lastly, notice the galleys (grey) adjacency to mess halls and stores (orange). Note that the arrangement has human alterations regarding the geometrical arrangement but not the topology and the relations between the rooms.

As a method in establishing an initial layout with no pre-existing topology or geometry ('white paper'); the algorithm

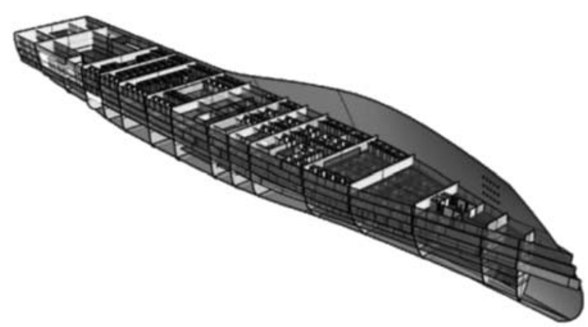

Figure 25. Three-dimensional model of Cruise ship case study with optimised decks incorporated (Pasia et al. 2019). (This figure is available in colour online.)

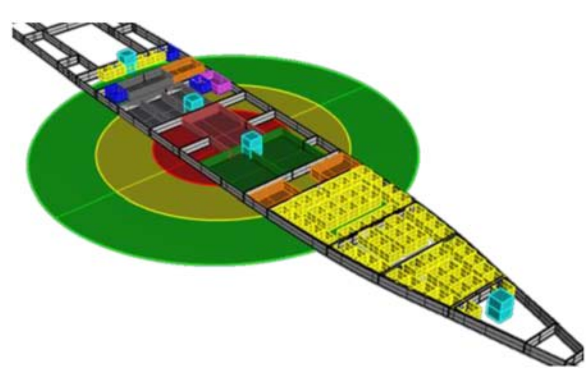

Figure 26. Visualisation of sound intensity (red–yellow–green) and evacuation points (turquoise) on optimised Deck 3. (This figure is available in colour online.)

Figure 27. MARE Cruise ship [Group Design project] with decks optimised exploded (Pasia et al. 2019). (This figure is available in colour online.)

showed its merits. Inaccuracies in the layout are more geometrical, rather than topological in nature and can be further fine-tuned by the user (Figure 26). Setting it up was simple and fast and the optimisation processes took minutes for each deck. No guessing or back tracking was needed, and the design is theoretically the best for this stage. The evacuation simulations conducted, using this layout, satisfied required times to muster and evacuate with no changes, thus proving the efficiency of this layout (Figure 27) (Pasia et al. 2019).

5.2. Further research

The introduced algorithm responded well to the simple tests but was mildly inaccurate in the real case study since many considerations, notably corridors, are not taken into account. The solution to that is that the designer then rationalises the arrangement which takes significantly less time than starting from scratch. The designer will also have to consider other objectives, beyond the optimised ones that will need to be satisfied. This will be achieved by integrating the present tool in the Holistic Optimisation Framework (Papanikolaou 2010; Priftis et al. 2018), linked with the other parametric models.

There are many ways the existing algorithm can be improved either by improving the basic features or by including more considerations and integrating with drafting tools. A major step would be to replace the Manhattan distance and the centroids with the actual walkable (path) distance and entry/exit points. However, the distance calculation will require a path-finding algorithm like Dijkstra's algorithm. Boulougouris and Papanikolaou (2002) outline how an evacuation process can be modelled using wayfinding algorithms (A*) and supporting graphics. This improvement will allow the algorithm to be more useful in evacuation-related calculations. The actual path to an exit coupled with congestion models can provide the contemporary designer with an accurate tool to predict and optimise the evacuation time at an early stage, considering all the other factors. However, it is noted that the Manhattan distance is usually quite accurate in calculating real distance and requires much less computational power. This is especially true for naval and passenger ships where the arrangement is largely following a grid. It is also commonly used in layout problems (Leno et al. 2016) and in ships, e.g. Igrec et al. (2019).

Moreover, for the process to be inclusive and applicable to a whole ship, the algorithm will need to expand to 3D with cross-deck application utilising staircases connecting different levels. Similarly, for naval vessels, many special needs can be catered for early in the design stage. For example, the survivability of the vessel can be enhanced through the intelligent allocation of system components along the length of the hull ensuring that non-redundant equipment is not placed in the same compartment. Considering noise isolating materials, their use can also be captured with proper adjustment of the weighting factors. This would allow optimising for less material, weight and cost versus better noise isolation.

Lastly, other applicable objectives can be; minimising the length of cabling/piping in the engine room and arranging the architecture of systems as in Brefort et al. (2018), also optimising the stability given the weight of the rooms, or improving the ease of construction.

6. Conclusions

Ships are becoming increasingly larger and more complex, pushing the boundaries to attain better commercial capabilities. Changes to the design, after the main decisions regarding topology have been taken, lead unavoidably to revisions. These changes become exponentially more expensive the later in the design spiral they are introduced. Design tools for the hull and superstructure of ships are already widespread in use. A synthesis of the general and internal design tools within an optimisation framework will largely help the naval architect start from a design that is balanced and takes into account all the phases of the ship's existence: design, production, operation, decommissioning. This will lower the design cost; by minimising revisions, improve the reliability of a design and lead to even more innovative designs as the computerisation of the design process has done in the past recent years.

The work presented above shows how simple optimisation tools can be used along with modified chromosome technique to array large numbers of rooms in a deck optimisation scenario at a very modest timeframe. An innovative use of the algorithm for the deck arrangement of a novel cruise ship showcases that a large number of rooms can be placed with relatively few issues and with great flexibility as to their location. Notable aspects of the work are the modularity and speed, both of which make it very useful for use in early design stages where requirements are fluid and there is little time to invest in an extensive study. In contrast to existing methods, this algorithm requires minimal previous work, provides clear results whilst allowing for great flexibility on the number, type and objectives of optimisation.

Lastly, the proposed methodology keeps the designer in the loop with the use of editable databases and weighting factors down to the rationalisation of arrangements to account for unoptimazable concerns. The application of the algorithm in an academic ship design project in real time provides merit to the above.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Panagiotis Louvros http://orcid.org/0000-0001-8623-0680
Evangelos Boulougouris http://orcid.org/0000-0001-5730-007X
Andrea Coraddu http://orcid.org/0000-0001-8891-4963
Dracos Vassalos http://orcid.org/0000-0002-0929-6173
Gerasimos Theotokatos http://orcid.org/0000-0003-3547-8867

References

Andrews D, Casarosa L, Pawling R, Galea ER, Deere S, Lawrence P. 2007. Integrating personnel movement simulation into preliminary ship design. In: Proceedings of the RINA International Conference on Human Factors in Ship Design, Vol. 150. p. 117–228.

Boulougouris EK, Papanikolaou AD. 2002. Modeling and simulation of the evacuation process of passenger ships. Proceedings of Int. Congress of the International Maritime Association of the Mediterranean (IMAM 2002); Crete.

Boulougouris EK, Papanikolaou AD. 2004. Optimization of the survivability of naval ships by genetic algorithms. In: Proceedings of the International Conference on Computer and IT Applications and in the Maritime Industries 2004; Siguenza, Spain, May 9–12. p. 175–189.

Boulougouris EK, Papanikolaou AD. 2006. Hull form optimization of a high-speed wave piercing monohull. In: Proceedings of 9th Int. Marine Design Conference-IMDC06; Ann Arbor-Michigan, USA. p. 21.

Boulougouris EK, Papanikolaou AD, Zaraphonitis G. 2004. Optimization of arrangements of Ro-Ro passenger ships with genetic algorithms. Ship Technol Res. 51(3):99–105.

Brefort D, Shields C, Jansen A, Duchateau E, Pawling R, Droste K, Jaspers T, Sypniewski M, Goodrum C, Parsons M, et al. 2018. An architectural framework for distributed naval ship systems. Ocean Eng. 147:375–385. https://doi.org/10.1016/j.oceaneng.2017.10.028.

Damski JC, and Gero JS. 1997. "An Evolutionary Approach to Generating Constraint-Based Space Layout Topologies." CAAD Futures 1997. [Conference Proceedings / ISBN 0-7923-4726-9] M?nchen (Germany), 4-6 August 1997, 855–864. doi:10.1007/978-94-011-5576-2_65.

Daniels A, Parsons M. 2006. An agent based approach to space allocation in general arrangements. In: Proceedings of the 9th International Marine Design Conference; May. p. 673–696.

Daniels AS, Tahmasbi F, Singer DJ. 2010. Intelligent ship arrangement passage variable lattice network studies and results. Nav Eng J. 122(2):107–119. https://doi.org/10.1111/j.1559-3584.2008.00153.x.

Erikstad SO. 2019. Design for modularity, Chapter 10. In: Papanikolaou A, editor. A holistic approach to ship design.. Springer Nature Switzerland AG. ISBN 978-3-030-02809-1.

Goldberg D. 1989. Genetic algorithms in search, optimization, and machine learning. Boston: Addison-Wesley..

Hasda R, Bhattacharjya R, Bennis F. 2017. Modified genetic algorithms for solving facility layout problems. Int J Interact. 11(3):713–725.

Hillier FS. 1963. Quantitative tools for plant layout analysis. J Ind Eng. 14(1):33–40.

Hillier FS, Connors MM. 1966. Quandratic assignment problem algorithm and the location of indivisible facilities. Manage Sci. 13(1):42–57.

Hu M, Wang M. 2004. Using genetic algorithms on facilities layout problems. Int J Adv Manuf Tech. 23(3–4):301–310.

Igrec B, Pawling R, Thomas G, Sobey A, Rigby J. 2019. An interactive layout exploration and optimisation method for early stage ship design. RINA, R Inst Nav Archit–19th Int Conf Comput Appl Shipbuilding, ICCAS 2019. 2(September).

Islier A. 1998. A genetic algorithm approach for multiple criteria facility layout design. Int J Prod Res. 36(6):1549–1569.

Jo J, Gero J. 1998. Space layout planning using an evolutionary approach. Artif Intell Eng. 12(3):149–162.

Jung S-K, Roh M-I, Kim K-S. 2018. Arrangement method of naval surface ship considering stability, operability and survivability. Ocean Eng. 152:316–333.

Kubalik J, Lazansky J, Zikl P. 2002. Layout problem optimization using genetic algorithms. In: Camarinha-Matos L. M., Afsarmanesh H., Ma\vrík V., editors. Knowledge and technology integration in production and services. Balancing Knowledge and Technology in Product and Service Life Cycle. US: Springer; p. 493–500. https://doi.org/10.1007/978-0-387.

Lee K, Han S, Roh M. 2002. Optimal compartment layout design for a naval ship using an improved genetic algorithm. Mar Technol. 39:159–169.

Leno IJ, Sankar SS, Ponnambalam SG. 2016. An elitist strategy genetic algorithm using simulated annealing algorithm as local search for facility layout design. p. 787–799.

Mathworks Matlab. 2019. Documentantion, How the genetic algorithm works; (accessed 2019 Dec 3). https://uk.mathworks.com/help/gads/how-the-genetic-algorithm-works.html.

Nick EK. 2008. Fuzzy optimal allocation and arrangement of spaces in naval surface ship design [PhD thesis]. University of Michigan.

Nick E, Parsons M, Nehrling B. 2006. Fuzzy optimal allocation of spaces to zone decks in general arrangements. In: Proceedings of the 9th International marine Design Conference; May. p. 651–672.

Nikolopoulos L, Boulougouris E. 2018. A methodology for the holistic, simulation driven ship design optimization under uncertainty. In: Proceedings of the 13th International Marine Design Conference (IMDC 2018), June 10–14, Helsinki, Finland; CRC Press. 16 p.

Nowacki H. 2003. Design synthesis and optimization – an historical perspective. OPTIMISTIC-Optimization in Marine Design, 39th WEGEMT Summer School, Berlin, Germany, March 19–23. p. 1–27.

Nowacki H. 2010. Five decades of computer-aided ship design. Comput-Aided Des. 42(11):956–969.

Ölçer AI, Tuzcu C, Turan O. 2006. An integrated multi-objective optimisation and fuzzy multi-attributive group decision-making technique for subdivision arrangement of Ro-Ro vessels. Appl Soft Comput J. https://doi.org/10.1016/j.asoc.2005.01.004

Papanikolaou A. 2010. Holistic ship design optimization. Comput-Aided Des. 42(11):1028–1044.

Parsons M, Chung H, Nick E, Daniels A, Liu S, Patel J. 2008. Intelligent ship arrangements: a new approach to general arrangement. Nav Eng J. 120 (3):51–65.

Pasia V, Ferrara S, Lipsanen A, Louvros P, Chu KF, Tan YS, Zuniga-Werner SJ. 2019. Group design project: design of expedition cruise ship. MARE. University of Strathclyde.

Priftis A, Boulougouris E, Turan O, Papanikolaou A. 2018. Parametric design and multi-objective optimisation of containerships. Ocean Eng. 156:347–357.

Tanaka T, Shinoda T, Wakida K. 2015. An evaluation method for design planning for general arrangement on ship's superstructure. 12th International Marine Design Conference 2015 Proceedings volume 3.

Walter M, Leyh C, Strahringer S. 2017. Knocking on industry's door: Needs in product-cost optimization in the early product life cycle stages. Complex Systems Informatics Modelling Quarterly. (13):43–60.

Yrjänäinen A, Johnsen T, Dæhlen JS, Kramer H, Monden R. 2019. Market conditions, mission requirements and operational profile. Chapter 4. In: Papanikolaou A., editor. A holistic approach to ship design. Cham: Springer Nature Switzerland AG; p. 75–121.

Zhou G, Ye M, Cao Z, Ye F. 2006. A genetic algorithm approach on a facility layout design problem with aisles.