WARGEAR
‘Real time’ generation of detailed layout plans of surface warships during early stage design

le Poole, Joan; Duchateau, Etienne; van Oers, Bart; Hopman, Hans; Kana, Austin A.

DOI
10.1016/j.oceaneng.2022.110815

Publication date
2022

Document Version
Final published version

Published in
Ocean Engineering

Ocean Engineering 250 (2022) 110815

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

WARGEAR: 'Real time' generation of detailed layout plans of surface warships during early stage design

Joan le Poole [a,*], Etienne Duchateau [b], Bart van Oers [b], Hans Hopman [a], Austin A. Kana [a]

[a] Delft University of Technology, The Netherlands
[b] Defence Materiel Organisation, The Netherlands

ARTICLE INFO

Keywords:
Ship layout design
Early stage design phase
Warships
Cross-correlation
WARGEAR

ABSTRACT

Generating detailed warship layouts is crucial to check technical feasibility and performance consistent with emergent requirement elucidation during early stage design. However, generating feasible detailed layouts is a complex and time consuming task.

Even today, detailed layout plans are often manually drawn using CAD software, taking up to 150 work hours to complete a single feasible layout plan, as found by the Netherlands Defence Materiel Organisation (DMO). As a result, the number of layout variations that can be generated and analysed is limited. This typically means that further detailed layout generation is postponed, increasing the risk of costly sizing and integration issues later in the design process. Therefore, a method that enables rapid insight into layout sizing issues is required.

This paper elaborates on the mathematical working mechanisms of the WARship GEneral ARrangement (WARGEAR) tool, that has been developed to support naval architects in detailing ship arrangements to space level in a matter of minutes. Contributions are: (1) a probabilistic staircase placement algorithm, (2) a network-based approach combined with probabilistic selection for allocation of spaces to compartments, (3) the use of cross-correlation to quickly arrange spaces, and (4) a 'carving'-based approach to ensure connectivity. A representative WARGEAR application case study is presented. This test shows how WARGEAR is able to confirm the feasibility of future warship arrangements at a high level of detail within minutes.

1. Introduction

Layout design is seen as an essential part of early stage naval ship design efforts (Carlson and Fireman, 1987; DeNucci, 2012; Van Oers, 2011b). It is key that decisions are made right during early stage ship design, because these have the highest impact on the overall design and are costly to change (Andrews, 2012; Mavris and DeLaurentis, 2000; Kana et al., 2016). These decisions lead to a set of design requirements that need to be fulfilled. Additionally, ships need to comply to international and naval regulations. To ensure that these requirements can be met with a technically feasible and affordable design the development of layouts is important (DeNucci, 2012; Van Oers et al., 2018; Le Poole et al., 2019). It is for this reason that naval architects at the Defence Materiel Organisation (DMO) of the Netherlands Ministry of Defence generate concept designs with various level of detail to inform decision-makers and to ensure feasibility and affordability of requirements for future warships (Van Oers et al., 2018).

Various (semi-)automated layout design tools are in use during early stage ship design (Van Oers, 2011b; Van Oers et al., 2018; Takken, 2009), but typically are designed for limited level of detail arrangement plans. Pursuing limited level of detail is favourable during early stage designs, because it reduces the required effort when requirements change, which is typical for that design stage (Duchateau, 2016). Naval architects need to consider a wide range of design aspects during the development of these concept designs. Examples of these aspects are stability, seakeeping, strength, speed, weight and system sizing. Present research focuses on system sizing. The risk of focusing on limited level of detail is that the arrangement will not fit when a higher level of detail is considered. Since most design tools are not suited to generate detailed layouts, naval architects use manual CAD software to generate detailed General Arrangement Plans (GAPs). These GAPs are developed to (1) inform the stakeholder dialogue on the impact of design requirements and on the effect of compromises, and (2) to de-risk lower level of detail arrangements and to get insight into conflicting or competing requirements. However, manual GAP development is found to be too time-consuming for the highly iterative speed of early warship design efforts. Indeed, generation of feasible GAPs can take up to 150

* Corresponding author.
E-mail address: j.j.lepoole@tudelft.nl (J. le Poole).

https://doi.org/10.1016/j.oceaneng.2022.110815
Received 31 May 2021; Received in revised form 3 February 2022; Accepted 11 February 2022
Available online 8 March 2022

work hours, as found by the DMO. Hence, acceleration of feasible warship concept design generation is considered beneficial, for three reasons (Le Poole et al., 2019):

1. A reduction in effort required for the creation of GAPs enables the generation and analysis of a larger number of layout variations. This increases the amount of design insight to support the stakeholder dialogue.
2. Earlier insight in sizing and integration problems will be enabled and therefore necessary changes to the design can be made earlier in the process.
3. More design variations can be generated and analysed, i.e. enables a more thorough investigation of possible trade-offs.

Such acceleration of concept design generation can be realised via automated design tools (Pawling, 2007, p. 194) (for examples of automated design tools see e.g. (Van Oers, 2011b; Duchateau, 2016; Nick, 2008)). Such design tools can rapidly generate a set of alternative design solutions (Van Oers, 2011b). Coupled with careful human input generation and elaborate data analysis, designers can extract design insights for decision-making. This requires designers to be familiar with the limitations and working mechanisms of design tools (Duchateau, 2016). Ultimately, the human designer is essential (1) to explore and understand the design options generated, which is an involved task (Duchateau, 2016); (2) to validate the data produced by design tools (Andrews, 2021); (3) to communicate and discuss obtained insights with relevant stakeholders (Van Oers et al., 2018); and (4) to include insights in the overall concept design (Pawling, 2007), while keeping the design balanced and coherent (Andrews, 2018, 2021).

Since such automated design tool for GAP generation was not available, the collaboration between the DMO and the Delft University of Technology in the WARGEAR (WARship GEneral ARrangement) project aimed to develop a tool to support DMO's naval architects in developing GAPs faster. WARGEAR is intended to support naval architects by complementing to concurrently used manual, human-controlled tools. Previous papers published during the project discussed a proof of concept (Le Poole et al., 2019), an integration of the WARGEAR tool and a queueing based logistic performance tool (Droste and Le Poole, 2020), and a validation and user acceptance case study aimed at the integration of the WARGEAR tool in the DMO design process (Le Poole et al., 2020). The gap in these papers is a detailed explanation of WARGEAR's working mechanisms and validation of the tool.

Although WARGEAR is directly connected to DMO's FIDES tool (Takken, 2009), the present paper aims to extend beyond a DMO-only application. Hence, a detailed description of the individual parts of WARGEAR will be provided to benefit the whole early stage ship design community. Further, the envisioned use of WARGEAR (as an automated tool) is illustrated in a case study. The applicability of automated design tools in early stage design will be discussed.

The structure of this paper is as follows. First, Section 3 will provide literature review of layout generation tools in ship design and architecture. Subsequently, a detailed breakdown of the tool's mathematical working mechanisms, as well as test cases for the various pieces of the code is provided in Section 4. Then, in Section 5 a test case will be presented in which output of the integrated tool is discussed. Section 6 provides a discussion and conclusions on the paper.

2. Definitions

The following terms are widely used in this paper, and have the following meaning:

1. Space: A space is defined as a room (e.g. a cabin or a galley) in the ship. A space has properties such as required area (RA) and allowed aspect ratios (AR).
2. Compartment: A compartment is defined as a volume inside the ship, enclosed by decks and bulkheads.
3. Functional building block (FBB): A FBB is defined as a low level of detail representation of multiple spaces serving a similar functionality (e.g. an accommodation FBB might represent multiple cabins). A FBB can overlap (partially) multiple compartments. FBBs are derived from the design building blocks used in the Design Building Block (DBB) approach (Takken, 2009; Andrews, 2003).

3. Literature review of layout generation tools

3.1. Ship layout design

Ship layout design tools can be distinguished based on a number of characteristics, such as whether tools are focused on volume or area, and the level of detail. Table 1 summarises five major tools used in ship design and provides a comparison with WARGEAR. It can be concluded that WARGEAR is most comparable with Intelligent Ship Arrangement (ISA) (Daniels et al., 2009; Nick, 2008), but needs to provide results faster. ISA is designed to support the naval architect in developing general arrangements, and takes the definition of the hull, decks, and bulkheads as input. In contrast, WARGEAR requires a medium level of detail (functional) arrangement as input (e.g. from FIDES (Takken, 2009) or DBB, which originated functional arrangements (Andrews, 2003)). Therefore, WARGEAR can be seen as an add-on to low-to-medium level of detail layout generation tools. Since early stage design phases are characterised by an iterative nature and frequently changing requirements (Duchateau, 2016), and high level of detail layouts inherently come with a reduction of design flexibility, WARGEAR is not aimed to function as a main layout generation tool used by naval architect, but rather to support naval architect in derisking lower level of detail layouts, i.e. to determine whether all systems fit (Le Poole et al., 2019). See Section 6.1 for a brief discussion on this process.

Although none of the tools investigated meet the speed requirement for WARGEAR, the following (mathematical) components are considered to be valuable for the development of WARGEAR:

1. *Visualisation and exploration of automatically generated layouts* can be supported via scatter plots (Van Oers, 2011b; Duchateau, 2016), but also requires full layouts to be generated for detailed insights (Daniels et al., 2009; Duchateau, 2016; Nick, 2008; Van Oers, 2011b; Van Oers et al., 2008).
2. *The use of networks to allocate spaces to compartments* has been investigated by Gillespie (2012), and proved to be a powerful way to deal with the multitude of system adjacency and global location requirements that need to be satisfied in a feasible layout. Networks can be used to evaluate ship designs, see for instance (Gillespie, 2012; Roth, 2017; Pawling and Andrews, 2018).
3. *Space arrangement optimisation* has been approached in different ways. For instance, ISA (Daniels et al., 2009; Nick, 2008) uses a growth based approach, while Packing (Van Oers, 2011b) utilises an overlap detection and removal approach.
4. *The sequence in which systems are arranged is important in both manual and automated design approaches*. For instance, Brown (1987), Andrews and Pawling (2008) discuss arrangement sequences for manual approaches. Examples of arrangement sequences in automated approaches can be found in: Van Oers (2011b), Nick (2008), Gillespie et al. (2013).
    - Brown (1987) proposes to start with arranging access routes and subsequently hierarchically arrange systems considering size and importance.
    - Andrews and Pawling (2008) propose:
        (a) Commence with those blocks already seen as causing design unbalance or conflict;

Table 1
Characteristics of ship layout design tools.
Source: Adapted from (Duchateau, 2016; Gillespie, 2012).

Method	Driver	Architectural	Diversity	Speed	Num. of solutions	Level of detail
DBB (Andrews and Dicks, 1997; Andrews, 2003, 2018)	volume	3D full ship	overall	hours-days/manual	few	low to high
ISA (Daniels et al., 2009; Nick, 2008)	area	2D deck	arrangement	hours/automated	hundreds	high
Packing (Van Oers, 2011b)	volume	2.5D/3D full ship	overall	hours/automated	thousands	low
FIDES (Takken, 2009)	volume	3D full ship	overall	days/manual	single	medium
Gillespie (Gillespie, 2012)	adjacency	network	arrangement	unknown[b]	few	low
WARGEAR[a]	area	2D deck	arrangement	minutes/automated	few tohundreds	high

[a]Requirements for WARGEAR.
[b]Estimated as minutes.

(b) Select the largest blocks first before tackling smaller blocks;
(c) Select the most constrained blocks before those less constrained;
(d) Start with the FLOAT blocks, then the MOVE blocks, followed by the FIGHT blocks, and, finally, the INFRASTRUCTURE blocks.

- Van Oers (2011b) proposes the following sequence for Packing: hull, decks, (optionally, very large or constrained objects), bulkheads, *other objects "in the order from large and/or constrained, to small and/or less constrained"* [emphasis added].
- ISA (Nick, 2008) commences with the arrangement of staircases and spaces on the DCD, and subsequently iteratively arranges spaces within each compartment.
- Gillespie et al. (2013) allocates communities of spaces based on descending global location preference, i.e. most restricted spaces are allocated first.

Generalising, sequential approaches tend to first tackle (perceived) large, constrained or constraining systems before arranging systems with less impact on the overall design.

3.2. Inspiration from architecture

WARGEAR has been inspired by various work from the field of architecture.

The generation of building layouts typically starts with an architectural program, which is specified by the architect in dialogue with the customer. Such architectural programs capture requirements for the building, and typically include items as: the total floor area, the building's footprint, a set with rooms, the required area per room, required aspect ratios per room, required adjacency between rooms, and type of adjacency between rooms (Merrell et al., 2010). Others divide their layout generation approach into two steps (Guo and Li, 2017; Medjdoub and Yannou, 2000). First, a topology is created to generate a rough layout that satisfy connectivity requirements from the architectural program. Second, the topology is refined into spatial layouts.

To generate spatial layouts a number of approaches can be taken. Four approaches are discussed below.

1. *Tile placement*. Peng et al. (2014) presents predefined templates of (irregular) shapes with allowed shape variation, combined with a two step approach to tile a predefined domain: (1) a discrete step to select approximate template positions and (2) a continuous step to refine and reshape the templates.
2. *Treemap algorithms*. A treemap is a way of visualising elements in hierarchical structures, in which an area is subdivided into smaller pieces, where the size of each piece is related to the importance of the pieces in the hierarchy. A special treemap algorithm is the squarified treemap algorithm to generate layouts, which attempts to subdivide the domain in pieces that have an aspect ratio close to 1 (Marson and Musse, 2010). The main drawback from this method that it is only able to subdivides squares or rectangles, which does not well resemble the shape of a ship.
3. *Growth based arrangement algorithms*. Inspired by the growing of crops, growth based arrangement methods start with populating a domain with 'seeds' which are then iteratively grown to the required size. Examples are can be found in Inoue and Takagi (2008), Lopes et al. (2010), Camozzato (2015). A ship design example is ISA (Nick, 2008). Also, this algorithm was previously used in WARGEAR (Le Poole et al., 2019).
4. *Inside out arrangement methods*. While the methods described above start with a predefined domain, inside out arrangement methods start with arranging the spaces and 'wrap' the outer wall around these spaces (Merrell et al., 2010).

Although vertical connectivity of buildings is necessary, i.e. vertical adjacent floors need to be connected, not all layout generation tools are able to generate multi-floor layouts. Examples of tools that generate single floor layouts can be found in Baušys and Pankrašovaite (2005), Camozzato (2015), Inoue and Takagi (2008), Marson and Musse (2010). Multi-floor layout plan generators that include vertical connections such as staircases can be found in Guo and Li (2017), Lopes et al. (2010), Merrell et al. (2010).

Since all spaces need to be accessible, architectural programs include requirements on adjacency, i.e. connectivity, between spaces (Merrell et al., 2010). Therefore attention needs to be paid to ways in which connectivity can be modelled. Three approaches are discussed:

1. Inclusion of passageways in the architectural program, and subsequent arrangement of these passageways in a similar way to space arrangement (Baušys and Pankrašovaite, 2005; Merrell et al., 2010). In the case of Baušys and Pankrašovaite (2005) passageways or halls need to meet a minimum required area requirement.
2. Placing doors according to a topology (Lopes et al., 2010; Guo and Li, 2017). Although in ship design some spaces might be used as a pass-through towards other spaces, in general each space is directly connected to passageways. Therefore this approach is less suitable for WARGEAR.
3. Carving passageways after space arrangement has been completed (Marson and Musse, 2010). This approach eliminates a pre-arrangement decision on the number of passageways that is to be included in the layout. A major drawback however is that this approach can result in too small spaces, as area is carved away from spaces to create passageways. This could be solved by readjusting the floor plan if the final area of any room is smaller that the minimum required area (Marson and Musse, 2010).

4. Method

The gap identified in Section 1 is a detailed explanation of the WARGEAR tool and the validation of this tool. This Section will elaborate on the tool's working mechanisms in detail.

4.1. Method overview

This section elaborates the WARGEAR method, which is divided into three parts:

1. The *input* required for WARGEAR will be discussed in Section 4.2.
2. The mathematics used in the *arrangement* phase is elaborated on next. Section 4.3 explains how main passageways are generated. Subsequently, Section 4.4 elaborates on the arrangement of staircases. Next, Section 4.5 discusses the allocation of spaces to compartments. Section 4.6 introduces a novel space arrangement method. Then, Section 4.7 explains how connectivity is assured throughout the ship. Finally, Section 4.8 explains how the mathematical method is integrated and steered by a nested optimisation approach.
3. The *post processing* of the resulting detailed arrangement is discussed in Section 4.9.

The order of arrangement steps in WARGEAR is based upon the idea that, first, global ship level decisions are taken (i.e. arrangement of passageways and staircases and allocation of spaces to compartments), where global is defined as influencing the arrangement of multiple compartments. These global decisions are most constraining to the design, as well as allow for the most accurate estimation of available area for allocating spaces to compartments. Second, compartment level decisions are taken (i.e. arrangement of spaces and ensuring connectivity). The first reason for this differentiation is that it is based upon other approaches to global and local arrangement problems (e.g. Medjdoub and Yannou (2000), Michalek et al. (2002), Nick (2008)). The second reason is that compartment level decisions have less impact on the total ship arrangement. The third reason for separating ship level decisions from compartment level decisions is to improve the arrangement of spaces at a compartment level (Le Poole et al., 2019), i.e. confined compartment arrangement problems are mathematically easier solvable than holistic ship arrangement problems. Overall, the arrangement procedure in WARGEAR compares well to arrangement sequences mentioned in Section 3.1, which tend to first tackle (perceived) large, constrained or constraining systems before arranging systems with less impact on the overall design.

4.2. Input

The input required for WARGEAR consists of the following items:

- *Functional arrangement.* Functional arrangements are volume block based arrangements. At the DMO, the FIDES tool (Takken, 2009) is used by naval architects to generate functional arrangements, describing the arrangement of FBBs. However, other tools could be used as well to create a low-to-medium level of detail 2D or 3D arrangement. An example of a functional arrangement is provided in Fig. 1. This research does not aim to develop a tool that can be used by naval architects to develop a GAP from scratch. This is similar to ISA, which requires a hull form and compartmentalisation, amongst others (Nick, 2008; Daniels et al., 2009). Instead the goal is to provide overcome the challenges faced in using current tooling, namely by helping (speed-up) derisking of layouts during concept development, while keeping the flexibility of using low level of detail tools concurrently to keep up with the pace of the design process, see also Section 6.1. The functional arrangement is used by WARGEAR to set the shape of the ship's hull and superstructure, as well as the rough internal arrangement divided up into different functional needs (cabin spaces, machinery spaces, operational spaces, etc.). To determine the net available positioning area, WARGEAR considers both the floor and ceiling of functional blocks. Typically, WARGEAR is used to arrange a set of spaces (as specified below) into a predefined set of functional blocks, e.g. an accommodation block is detailed by WARGEAR by arranging a set of cabins.

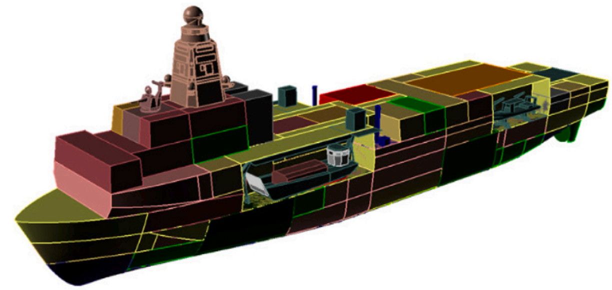


Fig. 1. An example of a functional arrangement (Van Oers, 2011a).

- *Space list.* A standardised spread sheet is used to create the space list for WARGEAR. For each space the naval architect specifies the required area, the minimum and maximum allowed aspect ratio (AR), the type of FBB the space is assigned to, as well as one or more specific FBBs that the space should be arranged in. Note that for many spaces this information is fixed and based on rules and regulations, e.g. accommodation standards for cabins. An example of such space list is shown in Table 2. During the development of functional arrangements, naval architects have an allocation of spaces to FBBs in mind (Le Poole et al., 2020). This envisioned allocation is a result from considerations of spatial, operational and cultural aspects, for instance. Hence, WARGEAR does not explicitly decide on inter-space relationships, but relies on the naval architect expressing preferences for decisions on inter-space relationships via allocation to FBBs.
- *Staircase types and arrangement options.* The naval architect needs to specify the required staircases and overall staircase arrangement options for each compartment. By specifying that a certain compartment should contain a staircase, the naval architect connects that compartment with the overhead compartment.
Currently up to three staircase types can be defined in WARGEAR. Within each staircase type an unlimited number of sizing variations can be specified. For instance, the naval architect might specify the following:
  - Type: *(Escape) ladder*
    Size: 1×1 m, 0.8×1.2 m
  - Type: *Standard staircase*
    Size: 3×1.5 m, 2.8×1.2 m, 2.6×1.1 m
  - Type: *Stairwell*
    Size: 3×3 m, 2.8×2.8 m

  If, for example, a ladder needs to be arranged, WARGEAR will attempt to arrange a 1×1 m ladder. If this attempt fails, the code will attempt to place a 0.8×1.2 m ladder. If this also fails, WARGEAR alerts the naval architect that placing the required ladder is not possible.
  Besides specifying the types and sizes of staircases, the naval architect may determine rough locations in which a staircase should be placed inside a compartment. Four options are available to choose from, namely:

  1. Port side (PS). This option will place staircases in allowed positions at the port side of the ship.
  2. Starboard side (SB). This option will place staircases in allowed positions at the starboard side of the ship.
  3. Centre line (CL). This option will place staircases in allowed positions close to the centre line of the ship.
  4. No preference. This default option will use the general rules for staircase placement as specified in Section 4.4.

  Further, specific functional blocks can be blocked for use for staircase arrangement. For instance, the naval architect may specify that no staircases can be placed in storage rooms.

Table 2
An example of the space list generated for WARGEAR.

ID	Name	Area	Aspect ratio low	Aspect ratio high	FBB[a] name	FBB IDs		
1	Officer's cabin	15	0.5	1	Accommodation cabins	25	26	27
2	Rating's cabin	20	0.5	1	Accommodation cabins	27	28	29
3	Officer's day room	40	0.5	1	Accommodation dayrooms	23	24	
4	Workshop	25	0.5	1	Workshop areas	1	5	

[a]Functional Building Block.

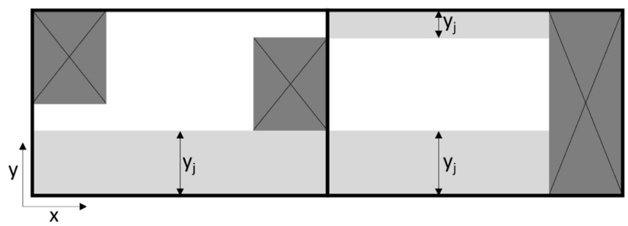


Fig. 2. Top view of a two compartment example on main passageway selection. Legend: *dark grey, black cross*: blocked area. *light grey*: available area for main longitudinal passageway. *two-headed arrow* indicate feasible y_j.

- *Run settings*. A variety of settings to run WARGEAR need to be specified. Examples are:
  1. Settings for the optimisation algorithm.
  2. File paths to relevant input files.
  3. Grid size, to control the resolution of WARGEAR's position matrices.

4.3. Passageway arrangement

Since horizontal and vertical connectivity through the ship requires significant area, the placement of passageways and staircases (see Section 4.4) needs to be taken into account in generating detailed layouts. It was considered to let WARGEAR decide upon the locations of all passageways. However, it was found that initial passageway routing in WARGEAR could yield unrealistic and even unacceptable results, as presented in Le Poole et al. (2019). Also, it was found that including the main passageways in the functional arrangement improves the initial area estimation (also on functional arrangement level), and thus leads to less technical risk. Furthermore, fixed main passageways reduce the calculation time, as the placement of main passageways is one of the main drivers for a good layout of spaces and staircases. Since design issues such as structural integrity are not considered by WARGEAR, but are taken into account by the naval architect in the functional arrangement, predefined passageways yield more realistic results.

Additionally, WARGEAR offers naval architects the option to include a single longitudinal passageway in user defined compartments. This option was included as naval architects might not model each main passageway in the functional arrangement. This might happen on less centralised decks or in small vessels, for instance. For each user defined compartment that should include a main longitudinal passageway j, WARGEAR uses a variable y_j to determine the transverse position of that passageway. y_j is used to select between transverse positions that result in the longest passageway possible. For instance, some area might be blocked, and this procedure helps to route main passageways such that blocked area is avoided to the maximum extent possible. This procedure is visualised in Fig. 2 for the case of two compartments.

4.4. Staircase arrangement

While passageways tend to influence the arrangement of horizontally adjacent compartments, the staircase arrangement influences the arrangement of vertically adjacent compartments. Contrary to the arrangement of passageways, WARGEAR is used to arrange all required staircases (although the functional arrangement might include fixed staircases), because of the smaller footprint. A generic set of rules to determine the positioning of staircases was required. Therefore, various existing GAPs of naval vessels and layouts generated by WARGEAR have been analysed and compared. This analysis led to the definition of the following rules for staircase positioning:

1. A long single staircase is preferred over split individual staircases, because of structural integrity reasons as well as deck area utilisation.
2. Staircases are typically placed directly adjacent to passageways, due to logistics and structural reasons. For instance, longitudinal bulkheads tend to be alongside main passageways. A higher probability for arrangement along longitudinal passageways than transverse passageways is used.
3. If no passageways are available, staircases are placed in lobby-like areas, to avoid arrangement in functional spaces above or below, which is typically prohibited by regulations.
4. If no passageways or lobbies are available on a deck, any position can be chosen.
5. The preferred locations on all decks that need to be connected need to be considered in the staircase position selection.

These rules could become too restrictive for certain arrangement problems. However, these cases have not yet been encountered. When WARGEAR is not able to place a staircase, a warning message is dropped.

4.4.1. Splitting staircases

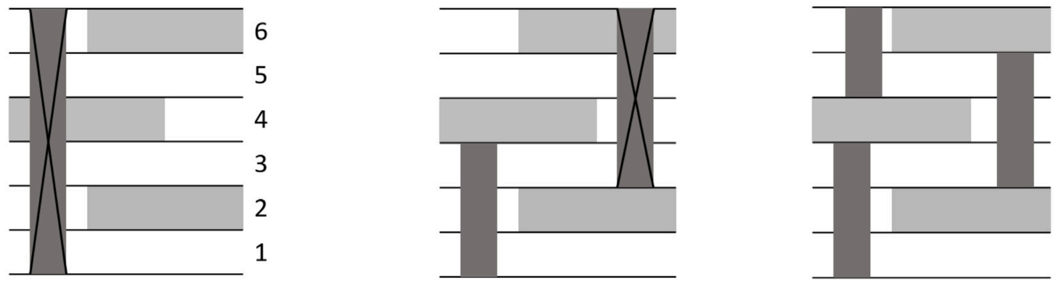
Following the first rule for staircase positioning, WARGEAR should arrange staircases such that staircases are as long as possible. The procedure used to determine how staircases are split into multiple staircases, when a long staircase cannot be arranged, is visualised in Fig. 3. The Figure shows a side-view of six decks, where it is not possible to place one staircase across all decks due to blocked area on deck 4. Two splitting operations are necessary to generate staircases such that deck 1 to 6 are connected. The splitting procedure is given in Algorithm 1.

In the example shown in Fig. 3, z_{split} is 4 and 6 respectively for the first and second splitting operation.

4.4.2. Determining preferred staircase positions

For each staircase to be arranged, the positioning guidelines outlined above are used to determine preferred positions. A staircase positioning (*SPM*) matrix is created for each deck based on these guidelines and is subsequently used in the final staircase position selection, which is discussed in Section 4.4.3. The (i,j)th element of SPM_z matrix for deck z is defined as follows:

$$SPM_z(i,j) = \begin{cases} 0, & \text{if the position cannot be used for staircases} \\ 1, & \text{if the position is directly adjacent to a longitudinal passageway} \\ 1, & \text{if the position is inside a lobby} \\ 0.75, & \text{if the position is directly adjacent to a transverse passageway} \\ 0.5, & \text{otherwise} \end{cases} \quad (1)$$



(a) The required staircase cannot be placed and needs to be split, as indicated by the black cross.

(b) First splitting operation results in one placeable and one unplaceable staircase.

(c) Next splitting operation results in two additional staircases that are placeable.

Fig. 3. Splitting one staircase in three staircases eventually to connect six partially blocked decks.

Input: {S} = set of unplaced staircases, initially containing all staircases;
Input: positioning matrices for all decks;
Output: $\{S_u\}$ = set of unplaceable staircases, initially empty;
Output: position matrices for all decks;
while *{S} is not empty* **do**
    (1) Attempt to place current first unplaced staircase i in the position matrix;
    if *Attempt is successful* **then**
        Remove staircase i from {S};
    else
        (2) Attempt to split staircase i:
        for $z_{split} = z_{min}$ **to** z_{max} **do**
            if z_{split} *causes failure of placement of current staircase* **then**
                Split current staircase into a staircase that should run from z_{min} to $z_{split} - 1$ and one that should run from $z_{split} - 1$ to z_{max};
                Remove staircase i from {S};
                Add both new staircases to {S};
                Return to (1);
            else if $z_{split} = z_{max}$ **then**
                Remove staircase i from {S};
                Add staircase i to $\{S_u\}$;
            end
        end
    end
end
for $\{S_u\}$ **do**
    Drop warning
end

Algorithm 1: Pseudo code for the arrangement of staircases

To determine the preferred locations of staircases, the *SPM*s for all relevant decks need to be taken into account. For a given staircase x that needs run from deck z_m till deck z_n, the final staircase positioning matrix for staircase x, $FSPM_x$, is generated by merging the staircase positioning matrices for decks z_m till z_n. This is done such that:

$$FSPM_x(i,j) = \begin{cases} 0, & \text{if } \prod_{z=m}^{n} SPM_z(i,j) = 0 \\ \sum_{z=m}^{n} SPM_z(i,j), & \text{otherwise} \end{cases} \tag{2}$$

An example of the combination of two staircase matrices into one FSPM is provided in Fig. 4.

4.4.3. Selecting staircase positions

The final step is to select a position from the generated *FSPM*. Obviously the more preferred locations should have a higher probability of being chosen. At the same time, the code should be able to select less preferred locations if this appears to be necessary to, for instance, arrange spaces more efficiently. Furthermore, predefined preferred staircase positions are taken into account, e.g. a specific staircase might therefore be placed starboard, while the most preferred positions in the *FSPM* are located around the centreline of the ship. The selection steps are:

1. Identify all available grid positions and their preference value.
2. Sort the available grid positions according to preference, which can be either the naval architects preference or the numerical values in the FSPM.
3. Use a probability density function and a variable x to select the location of the staircase.

Referring to the example FSPM, Fig. 4(e) results in 56 feasible positions, of which twenty have equal highest preference, i.e. preference value 1.5, two positions follow closely with preference value 1.25, and 34 positions have preference value 1. The example shows that already for very coarse grids many positions with equal preference can exist. This is even more true for higher fidelity positioning grids as used in WARGEAR. Therefore a selection function is required that (1) provides roughly equal probabilities for early, i.e. more preferred, positions, and (2) low probabilities for late entries, i.e. less to non-preferred, positions. In this research the probability density functions are based on Eq. (3), since this function, depending on the n-value, possesses the required characteristics.

$$f(x) = b(1 - x^n) \tag{3}$$

Where $b = \frac{n+1}{n+2}$, which can be derived by integrating $f(x)$, and recognising that the cumulative probability of $f(x)$ equals to 1.

Four variations of such probability density functions are visualised in Fig. 5. In this research $n = 10$ is used as it proved to sufficiently promote the use of preferred positions. For instance, $n = 2$ is less suited as it does not provide equal probabilities for earlier positions in the list. The x-axis represents the sorted set of feasible grid positions and is related to a variable x. x can be randomly generated or obtained from an optimisation algorithm. The latter is used in present research.

4.5. Allocation of spaces to compartments

The meaningful allocation of spaces to compartments has been recognised as one of the key problems in early stage ship design (DeNucci, 2012; Gillespie et al., 2013). The location of spaces and systems in a ship design impacts various performances, e.g. logistic performance (Droste et al., 2018; Le Poole, 2018), and impacts other systems, e.g. the routing of interconnections between distributed systems (Duchateau et al., 2018).

One of the inputs of WARGEAR is an allocation of spaces to FBBs by the naval architects. Since FBBs in the functional arrangement can span

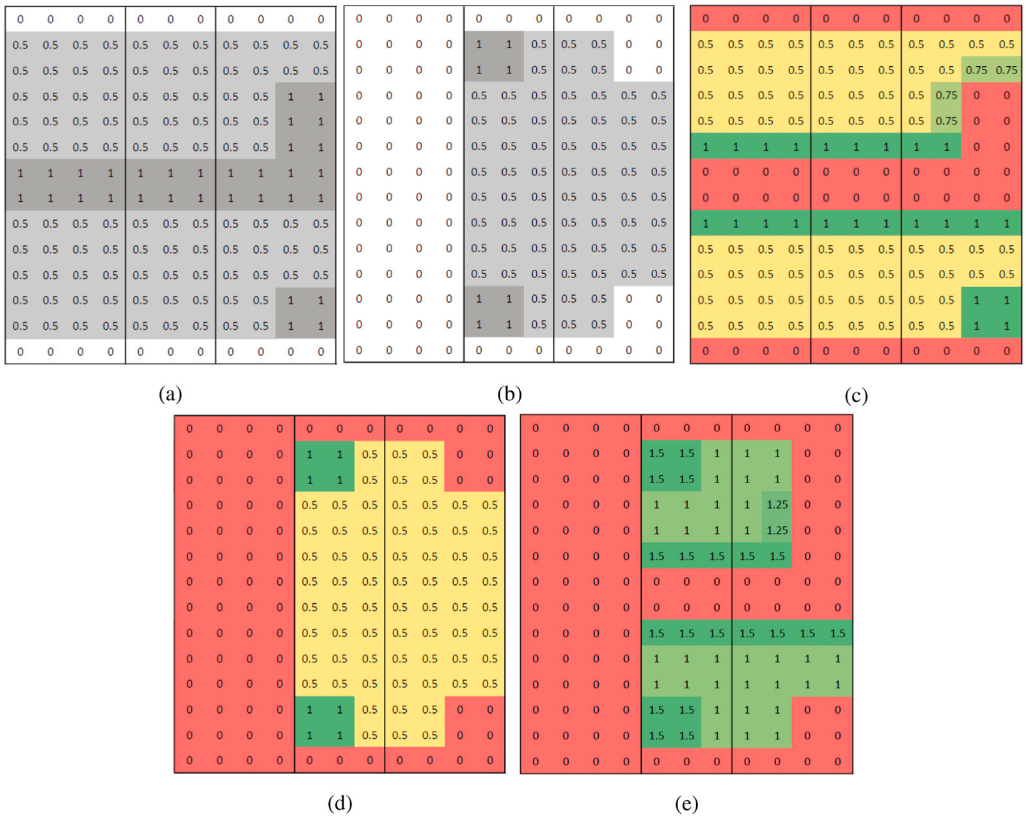


Fig. 4. Creating a staircase positioning matrix for a two deck, three compartment test case. (a) Deck 1 with a L-shape passageway and a small lobby. (b) Deck 2 with an irregular shape and two lobbies. (c) SPM_1 for deck 1. (d) SPM_2 for deck 2. (e) Aggregated FSPM for the two decks, where higher values represent more preferred positions for staircases.

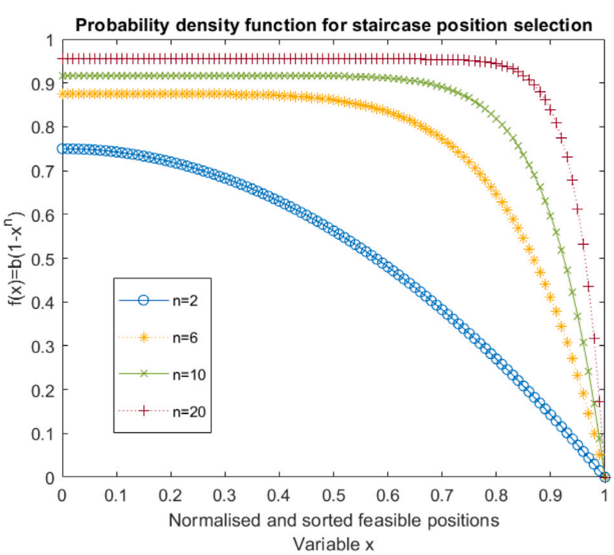


Fig. 5. Four probability density functions for staircase selection for different n-values based on the function $f(x) = b(1 - x^n)$.

multiple compartments, and spaces are arranged per compartment, an allocation of spaces to compartments is required (Le Poole et al., 2019). Previously, only the allowed FBBs were considered for allocation, without considering the required area of spaces and the available area in compartments. This led to infeasible allocations, i.e. too many spaces could be allocated to compartments (Le Poole et al., 2019).

Gillespie (2012) used a network partitioning method with predefined preferences for global positions to allocate spaces to compartments. This method is useful to obtain allocations that maximise the collective preference of the layout. However, the available area in compartments is not considered, which can cause over-utilised compartments. Also, defining all relationships between systems for each ship design (project) is considered to be too time consuming.

Since the functional arrangement already satisfies the major relationships, e.g. accommodation should not be placed adjacent to main machinery spaces, and naval architects already have an allocation of spaces to compartments in mind (Le Poole et al., 2020), the aim is to develop a method that considers this envisioned allocation, as well as the available area in each relevant compartment and the required area for the spaces, such that the best possible (from an area perspective) allocation of spaces to compartments is obtained. To improve the probability that allocation of spaces is possible, the order in which spaces are allocated and compartments are used needs to be considered carefully, as elaborated below. Actual arrangement needs to be done to check whether spaces actually fit in the possibly irregular positioning matrix.

The available area for spaces in compartment j is defined by Eq. (4).

$$A_{available,j} = A_{compartment,j} - A_{blocked,j} - A_{staircase,j} - A_{passageways,j} - A_{allocated\ spaces,j} \quad (4)$$

In which $A_{compartment,j}$ is the total useable area between two bulkheads within the ship's hull or superstructure. The useable area is defined as the minimum of the floor and ceiling area in a compartment, resulting from flare or tumblehome hull or superstructure shapes. $A_{blocked,j}$ is the area of functional building blocks in compartment j that cannot be used for space arrangement, e.g. exhaust casings and HVAC rooms. The area used by staircases and passageways in compartment j is

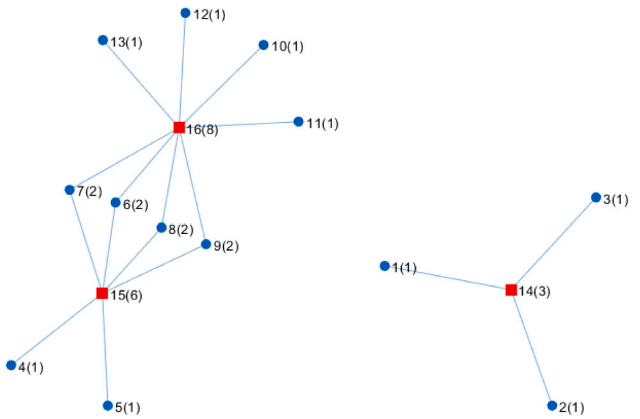


Fig. 6. Network of the relation between spaces and compartments. Spaces are represented by round nodes and compartments by square nodes. Numbers indicate node IDs and (degree).

given by $A_{staircase,j}$ and $A_{passageways,j}$ respectively. $A_{allocated\ spaces,j}$ is the total required area RA of spaces already allocated to compartment j. Compartments with more available area are more likely to be used than compartments with less available area, since the probability that spaces fit is higher for the former.

The success of the allocation of spaces is also dependent on the size of spaces and the order in which spaces are allocated. Indeed, the probability that a large space fits in a compartment that is already partly used by other spaces is lower than the probability that a small space fits in the same compartment.

In this paper, the list of allowed FBBs per space is translated into a network that represents the relations (edges) between spaces (nodes) and the compartments (nodes) that comprise the allowed FBBs, see for instance Fig. 6. For all nodes the degree is calculated, and given in parenthesis. The degree of a node is defined as the number of edges that connect to that node. For example, the degree of space 1 is 1, as it is only connected to compartment 14, whereas the degree of compartment 14 is 3, as it is connected to the spaces 1 to 3. Subsequently the degree of the nodes representing compartments and spaces is used to determine (1) which spaces should be allocated first and (2) which compartments should be used first. Although a more thorough study has been performed, the example in Fig. 6 is used to explain the two rules:

1. Consider the case that the available area in compartment 16 is insufficient for space 6 to 13, but the area of compartment 15 and 16 is sufficient for space 4 to 13. If space 6 to 9 are allocated to compartment 16 before space 10 to 13, then spaces 10 to 13 cannot be allocated to compartment 16. Therefore, comparing the degrees of the spaces 10 to 13 with the degrees of the spaces 6 to 9, one can find that *allocating spaces with a lower degree prior to spaces with a higher degree is desirable*.
2. Using the rule above, spaces 1 to 5 and 10 to 13 can be allocated. Subsequently, spaces 6 to 9 need to be allocated. Although the example is not very complex, it can be argued that *it is desirable to use compartments with a lower degree prior to compartments with a higher degree*, since the probability that compartments contain spaces that still need to be allocated is less for compartments with a smaller degree. Therefore, the probability that spaces cannot be allocated is also smaller. In the example, spaces 6 to 9 will be allocated to compartment 15 first, and the spaces that do not fit will be allocated to compartment 16.

Summarising, the following four statements are considered when allocating spaces to compartments, to maximise the probability that spaces are successfully allocated from an area point of view:

1. Large spaces are to be allocated prior to smaller spaces.
2. Compartments with more unused area are to be used prior to compartments with less unused area.
3. Spaces that are allowed in only a few compartments are to be allocated prior to those that are allowed in more compartments.
4. Compartments that are connected to less spaces are to be used prior to those connected to more spaces.

A roulette wheel selection method is used to select between available compartments for space i. In general, roulette wheel selection assumes that the probability of selection is proportional to the fitness of an individual. If N individuals are considered, each with a fitness $w_i > 0 (i = 1, 2, \dots, N)$, then the selection probability of individual i is given by Eq. (5) (Lipowski and Lipowska, 2012).

$$p_i = \frac{w_i}{\sum_{i=1}^{N} w_i} \qquad (i = 1, 2, \dots, N) \tag{5}$$

Subsequently the roulette wheel is constructed with sectors whose size is proportional to $w_i\,(i = 1, 2, \dots, N)$. Selection of an individual i is done by randomly selecting a point x at the roulette wheel and identifying the corresponding sector (Lipowski and Lipowska, 2012). In this paper, the fitness is defined as the probability $P_{allocation,ij}$ that space i is allocated to compartment j. $P_{allocation,ij}$ is given by Eq. (6), provided that the allocation i to j, $Alloc_{ij}$ is allowed. The first fraction in this equation relates the statements 1 and 2 listed above. The second fraction relates the statements 3 and 4. The cumulative roulette wheel selection probability $P_{sel,ij}$ that compartment j is selected for space i is given by Eq. (7). Subsequently the cumulative selection probability vector for the allocation of space i $P_{cum,sel}$ is given by Eq. (8).

$$P_{allocation,ij} = \begin{cases} \frac{A_{available,j}}{RA_i} \cdot \frac{Degree_{space,i}}{Degree_{comp,j}} & \text{if } A_{available,j} \geq RA_i \text{ and } Alloc_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$P_{sel,ij} = \frac{P_{allocation,ij}}{\sum_{j=1}^{N_{comp}} P_{allocation,ij}} \qquad (j = 1, 2, \dots, N_{comp}) \tag{7}$$

In which N_{comp} is the number of compartments.

$$P_{cum,sel}(j) = \sum_{n=1}^{j} P_{sel,ij} \qquad (j = 1, 2, \dots, N_{comp}) \tag{8}$$

A careful reader might have found that the inclusion of the required area of spaces (RA_i) and the degree of spaces ($Degree_{space,i}$) in Eq. (6) does not influence the outcome of the analysis, as these cancel out in Eq. (7).

The selection of a compartment on the roulette wheel would be usually done by generating a random number x. However, to assure traceability and to allow for regeneration of each layout, in this paper variables provided by the optimisation algorithm are used, see Section 4.8. After each allocated space, $A_{available,j}$ is updated. The order in which spaces are allocated is determined via multi-level sorting. Spaces are sorted by ascending degree first (statement 3.) and then by descending RA (statement 1.).

The method for allocation of spaces to compartments is summarised in Fig. 7. To illustrate the allocation method a small example will be elaborated on. Assume a functional arrangement with four FBBs (named A-D) spanning respectively 1, 1, 2, and 1 compartments (named 1–5). The area of the FBBs is respectively 40, 40, 100, and 50 m^2. The areas of the compartments is respectively 40, 40, 50, 50, and 50 m^2. Ten spaces are allocated to the four FBBs, see Table 3. The corresponding allocation of spaces to compartments also provides the degrees of the spaces and compartments. After determining the degree of the spaces, the order in which spaces will be allocated can be established, by sorting by Degree space first, followed by sorting by RA. Subsequently

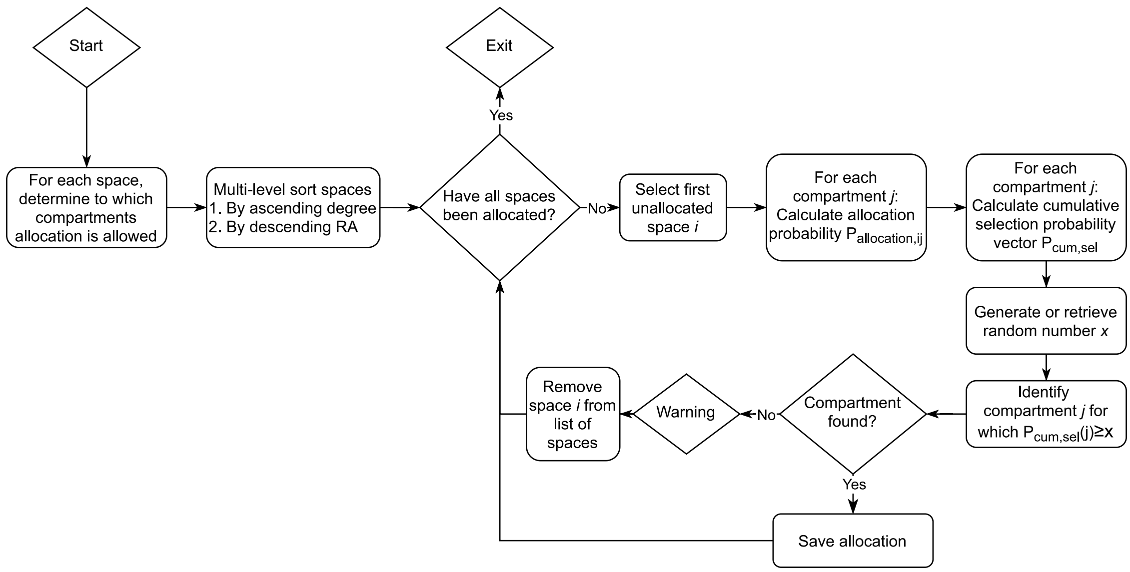


Fig. 7. Flow chart of the allocation of spaces to compartments.

Table 3
Example of ten spaces and their allocation to FBBs.

Space	RA	Allocated to FBB	Compartments					Degree space	Order index
			1	2	3	4	5		
1	15	A, B	1	1	0	0	0	2	1
2	15	A, B	1	1	0	0	0	2	2
3	15	A, B	1	1	0	0	0	2	3
4	15	C, D	0	0	1	1	1	3	7
5	15	C, D	0	0	1	1	1	3	8
6	50	C, D	0	0	1	1	1	3	4
7	30	A, B, C, D	1	1	1	1	1	5	10
8	20	B, C, D	0	1	1	1	1	4	9
9	20	C, D	0	0	1	1	1	3	5
10	20	C, D	0	0	1	1	1	3	6
Degree compartment			4	5	7	7	7		

Table 4
Allocation of ten spaces (see Table 3) to five compartments.

Space[a]	RA	Randomly selected compartment	$A_{available,j}$				
			1	2	3	4	5
–			40	40	50	50	50
1	15	2	40	25[b]	50	50	50
2	15	2	40	10	50	50	50
3	15	1	25	10	50	50	50
6	50	4	25	10	50	0	50
9	20	3	25	10	30	0	50
10	20	5	25	10	30	0	30
4	15	3	25	10	15	0	30
5	15	3	25	10	0	0	30
8	20	5	25	10	0	0	10
7	**30**	**0**[c]	**25**	**10**	**0**	**0**	**10**

[a]Spaces are sorted based on degree and RA.
[b]$A_{available,j} - RA_i = A_{available,2} - RA_1 = 40 - 15 = 25$.
[c]No compartment with sufficient area is available.

each space is allocated. Following the steps in Fig. 7, for each space $P_{allocation,ij}$, $P_{sel,ij}$, and $P_{cum,sel}$ are calculated. Then a random number is drawn, and the corresponding compartment is determined. Before allocating the next space, the available area in the selected compartment is updated, see Table 4. For space 7, the selected compartment is 0, which means this space could not be allocated because there is no compartment available that has sufficient area available to fit space 20. In the supplementary data, see Section 7, the allocation method is tested. This test supports the conclusion that the proposed allocation method performs equally or better than variations to the method. Hence, the method as presented above is used in the remainder of the paper.

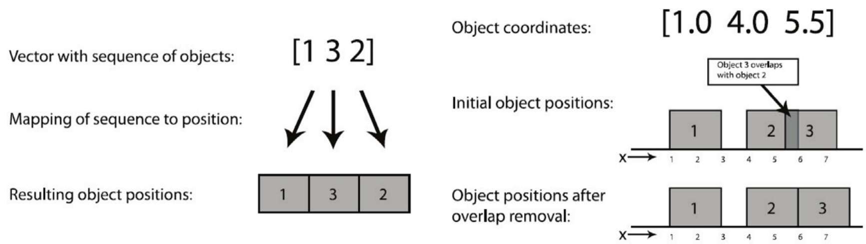
4.6. Space arrangement

In this section a novel space arrangement method based on cross-correlation is proposed and compared to the seed and growth algorithm used in WARGEAR previously (Le Poole et al., 2019). The main reason to change the space arrangement method is the speed advantage of cross-correlation over seed and growth based arrangement methods. In addition, cross-correlation requires less optimisation variables to arrange spaces, improving the optimisation algorithm's capability to find suitable answers in limited amount of time.

In Section 3.2 four approaches used in architectural tools were discussed. Further, two approaches can be taken to the parametrisation of space positions (Van Oers, 2007), namely based on the sequence in which spaces are arranged, see Fig. 8(a) and based on (initial) coordinates and overlap detection and removal, Fig. 8(b).

However, it was found that the seed and growth algorithm is time consuming due to its iterative nature. Also, the optimisation algorithm lacks control over space arrangement via initial seed locations, because spaces grow 'randomly' around the seed. As shown in Fig. 9, different seed locations might result in the same arrangement of spaces. Therefore a feasible, and even optimal, solution might be easily found for simple arrangement problems, but more complex arrangement problems can be more challenging for the seed and growth method. Indeed, large changes in input parameters might only result in small changes in layouts. Therefore, the optimiser needs to be robust enough to explore the whole design space, since early convergence might lead to stopping in local optima.

Instead of applying a method of iterative growth of spaces to identify how spaces could best be arranged in a given positioning area, this paper proposes a method that can quickly assess which positions in a position matrix are suitable for a given space. Growth based approaches start with a single point in the positioning matrix and grow spaces till these meet their required area. In contrast, the proposed method attempts to directly place the most preferred feasible space shape in the available area, and thus requires fewer iterations.



(a) Arranging spaces based on sequence

(b) Arranging spaces based on coordinates and overlap detection and removal

Fig. 8. Two approaches to parametrisation of space positions (Van Oers, 2007).

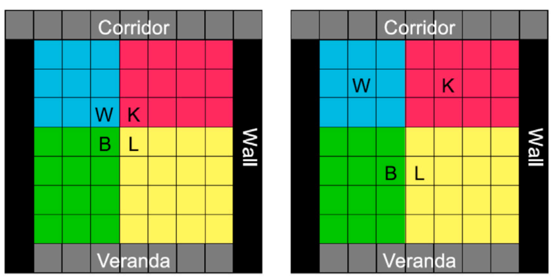


Fig. 9. Different initial positions (indicated by [W,K,B,L]) can lead to the same layout using seed and growth space arrangement algorithms (Inoue and Takagi, 2008).

The new proposed space arrangement method uses cross-correlation, a mathematical operation that expresses how one function is correlated to another function (Bourke, 1996). To the best of the authors knowledge, this method is new for layout design, although a variant can be found in the Packing methodology (Van Oers, 2011b). Inspiration for this new arrangement method was found in the fields of signal processing (Burrus and Parks, 1985; Najafi et al., 2020), neural network image recognition (Lo et al., 1995; Li et al., 2019), and probability theory (Pruinelli et al., 2019).

Since the space arrangement problem for WARGEAR focuses on a 2D deck plan, the mathematical expression for 2D cross-correlation has been adopted. The expression of the 2D cross-correlation between m_Axn_A-matrix A and m_Bxn_B-matrix B has been given in Eq. (9). Matrix C is a m_Cxn_C-matrix that contains to what extent matrix B can be placed at position $(j-1, k-1)$ in A, with $m_C = m_A - m_B + 1$ and $n_C = n_A - n_B + 1$. Note that other dimensions of C are also possible, for instance m_Axn_A. However, in this paper m_Cxn_C is used as it provides only feasible positions within A. If $C(j+1, k+1) = \sum_{i=1}^{n_{\text{elements in B}}} B_i$, then matrix B can be placed at position $(j+1 : j+m_B, k+1 : k+n_B)$ in A.

$$C(j+1, k+1) = \sum_m \sum_n A(m,n) \cdot \bar{B}(m-j, n-k) \tag{9}$$

Where

$$j = 0 : m_A - m_B \tag{10}$$

$$k = 0 : n_A - n_B \tag{11}$$

$$m = 1 : m_A \tag{12}$$

$$n = 1 : n_A \tag{13}$$

$$1 \le m - j \le m_B \tag{14}$$

$$1 \le n - k \le n_B \tag{15}$$

$$j, k, m, n \in \mathbb{Z}^{\geq} \tag{16}$$

Further, $\bar{B}$ denotes the complex conjugate of B, although for the arrangement problem $\bar{B} = B$, because of the absence of complex numbers in B.

Matrix A represents the positioning matrix, and contains ones when a position is available and zeros otherwise. Matrix B represents a space and contains ones only, since WARGEAR arranges rectangle spaces only (Le Poole et al., 2019). (Note, arranging irregular shapes is also possible using this method). Let B_i represent matrix B for space i. The area of B_i is equal to the size of space i, while its dimensions are based on the allowed aspect ratio of space i. Since a space might not fit in A, it is necessary to vary the aspect ratio of B_i. Since the objective is to satisfy the area and aspect ratio requirements for all spaces, the dimensions of B_i need to be varied such that both requirements are met to the maximum extent possible. To achieve this, first the table $B_{range,dimensions}$ is created for space i. It contains all lengths and widths that satisfy Eqs. (17), (18), and (19) and therefore might be considered for space i.

$$Area_i = Length_i \cdot Width_i \le RA \cdot (1 + AOP) \tag{17}$$

In which AOP is a constant Area Overshoot Percentage, which allows spaces to be AOP % larger than RA. In this paper $AOP = 20$.

$$min(AR) \le AspectRatio_i \le max(AR) \tag{18}$$

$$\frac{1}{max(AR)} \le AspectRatio_i \le \frac{1}{min(AR)} \tag{19}$$

In which $AspectRatio_i$ is given by Eq. (20) and AR is a two element vector containing the range of allowed aspect ratios for space i. Typically $AR = [0.5\ 1]$.

$$AspectRatio_i = \frac{Length_i}{Width_i} \tag{20}$$

Subsequently table $B_{range,dimensions}$ is sorted by the following sorting method. If table $B_{range,dimensions}$ contains rows in which $Area = RA$:

1. The first subset contains the rows in which $Area = RA$.
2. The second subset contains the rows in which $Area > RA$. These rows are sorted based on *increasing* size.
3. The third subset contains the rows in which $Area < RA$. These rows are sorted based on *decreasing* size.

Otherwise:

1. The first subset contains the rows in which $Area > RA$. These rows are sorted based on *increasing* size.
2. The second subset contains the rows in which $Area < RA$. These rows are sorted based on *decreasing* size.

Each subset is sorted on increasing aspect ratio.

Following the generation of table $B_{range,dimensions}$, the arrangement method outlined in Algorithm 2 is used to arrange space i. The sorting method used ensures that (1) the aspect ratio requirement is always met, and (2) the area requirement is met to the maximum extent possible, while limiting exceeding the required area.

```
index = 0;
while space i is not arranged do
    index = index+1;
    select L_index and B_index from B_range,dimensions;
    create L_index x B_index -matrix B_i;
    cross-correlate matrices A and B_i to matrix C, using
     Equation (9);
    if at least one position is available then
        select an available position;
        arrange space i at the selected position;
        update matrix A;
    else
        return
    end
end
```

Algorithm 2: Pseudo code for the arrangement of space i

To clarify how the method outlined above leads to the identification of feasible positions for a given positioning matrix A and a space B, consider the following matrices A and B, shown respectively in Matrices (21) and (22). Then the cross-correlated matrix C of the matrices A and B is given by Matrix (23). The only positions in C that satisfy $C(j+1,k+1) = \sum_{i=1}^{n_{\text{elements in B}}} B_i$ are $(1,1)$ and $(3,3)$. Therefore the only positions where B can be placed in A are $A(1:2,1:2)$ and $A(3:4,3:4)$. The accuracy of this answer is clear from comparing Matrices (21) and (22).

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{21}$$

$$B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{22}$$

$$C = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 2 & 2 \\ 0 & 2 & 4 \end{bmatrix} \tag{23}$$

However, three issues regarding space arrangement still remain to be addressed.

1. In which order should multiple spaces be arranged? There are three variations possible: (1) large spaces are arranged prior to small spaces, (2) small spaces are arranged prior to large spaces, and (3) spaces are, from a space size perspective, arranged in a mixed order. The order of the latter is selected via variables provided by the optimisation algorithm.
2. How to choose between available positions for space i? Indeed, the simple example shown above already gives two possible positions for space B. In a larger positioning matrix, such as one representing a compartment or a deck in a ship, the number of feasible positions for a given space will likely be large. Four options for choosing from feasible positions are proposed in this paper, which will be elaborated.

   (a) *Choose the first available position*: This causes spaces to be asymmetrically arranged in a compartment, which is not preferred from a ship's stability point of view, Fig. 10(a). This causes spaces to be arranged in the aft starboard side of a compartment. More problematically is location of void space. Such void space is best used to arrange other spaces, or to place passageways to ensure connectivity. In the case such void space is absent between spaces, significant area needs to be taken from spaces or spaces needs to be rearranged. Note that void space can be sensible in ship design, for instance to increase the ship's length for seakeeping (Keuning and Pinkster, 1995) or for future growth (Ferreiro and Stonehouse, 1991), although such void space usually is product of design margins and not a surprising result from the arrangement process.

   (b) To improve the symmetry of the arrangement *choose positions as close to the ship's centreline (CL) as possible*. This causes spaces to be arranged aft in a compartment predominantly, Fig. 10(b). Again, this option does not allow passageways to 'emerge' from the arrangement of spaces.

   (c) *Choose positions as far from the centre of the compartment as possible*. This method first arranges spaces in the corners of the compartment, and subsequently closer to the centre of the compartment, Fig. 10(c). Generally, this proves to result in a more useful distribution of void space to be used for connectivity.

   (d) *An optimisation algorithm selects a position*. The previous three options do not require variables. Although that might lead to a fast answer, as no optimisation is required, optimality is not guaranteed. To allow WARGEAR to find more optimal designs, the fourth option is to use variables provided by an optimisation algorithm to select from available positions, Fig. 10(d). However, this might also require space overlap detection and removal.

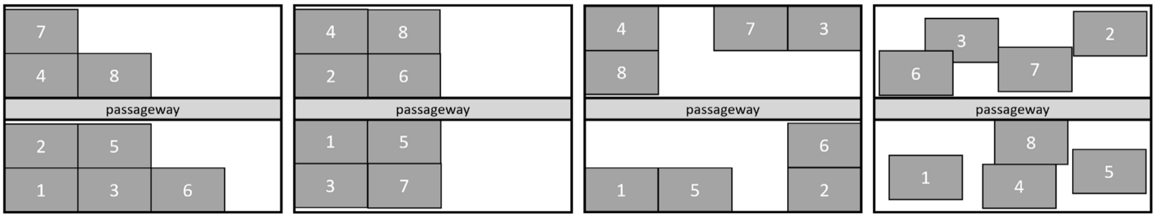
3. The orientation in which a space is arranged can significantly impact the success of the arrangement of other spaces. In some cases it would be beneficial to change the orientation of spaces. For instance, it might be useful to arrange space i in transverse direction although in table $B_{range,dimensions}$ the first row contains values for a longitudinally arranged space as this could allow other spaces to be more effectively arranged. To allow WARGEAR to change the orientation of spaces, an additional variable might be used to select whether a space is arranged in longitudinal or transverse direction.

In the supplementary data, see Section 7, the cross-correlation based space arrangement method is extensively tested. The test showed that the cross-correlation method is, on average, 20 times faster than the seed and growth method. Also, the quality (in terms of difference between required and achieved area of spaces) of the layouts generated by the cross-correlation method is better than generated by the seed and growth algorithm. In addition, an optimiser selected arrangement order of spaces, combined with optimiser selected orientation of spaces was found to perform well. From a layout quality perspective, the position selection options A and C outperform options B and D. Selecting an option by the optimisation algorithm was found to perform similar to options A and C. The optimisation algorithm typically choose options A and C as well, underlining the performance of these position selection options.

4.7. Ensuring connectivity

Although the main passageways are taken from the functional arrangement, or generated by WARGEAR (see Section 4.3), and staircases are placed prior to space arrangement, (see Section 4.4), an additional step is required to ensure all spaces are connected and can be properly accessed. This additional step checks the connectivity between passageways, staircases, and spaces in a compartment, and corrects connectivity if it is found to be insufficient. The following connectivity checks are implemented by default, but can be turned off by the naval architect:

1. *Passageway–passageway*. Multiple passageways, e.g. two parallel passageways, can be connected if necessary. For instance, to



(a) Option A. (b) Option B. (c) Option C. (d) Option D.

Fig. 10. Visualisation of space position selection options in an example of a compartment with a central passageway.

generate escape routes between port-side and starboard passageways.

2. *Passageway–staircase*. Staircases need to be accessible from passageways, if passageways and/or staircases exist.
3. *Staircase–staircase*. In the case of multiple staircases in one compartment, these staircases are to be connected, again to create escape routes.
4. *Space–passageway*. Basic feasibility requires spaces to be accessible from the main passageways.
5. *Space–staircase*. If main passageways are absent, e.g. below the damage control deck, spaces are to be connected to staircases, again because of basic feasibility requirements.

Connectivity is ensured *after* spaces have been arranged and is based on an existing carving method (Marson and Musse, 2010), discussed in Section 3. Depending on the selection of connectivity checks demanded by the naval architect, the algorithm arranges additional passageways to connect passageways, staircases, and spaces. The width of these passageways is controlled via a single parameter set by the naval architect. Using a network representation of the arrangement of passageways, staircases, and spaces in each compartment, the algorithm uses the walls of spaces as potential locations of additional passageways. This reflects observations made in a study into GAPs generated by naval architects, namely that passageways tend to share walls with spaces. The algorithm uses the following steps to ensure connectivity in a compartment:

1. A network representation of passageways, staircases, and spaces, in the form of an adjacency matrix A is created. $A(i,j)$ is 1 if node i and j are connected, and 0 otherwise. Edges and nodes located at bulkheads are removed, because additional passageways located at bulkheads lead to a large reduction of space size. Indeed, these passageways can only use area on one side of the bulkhead.
2. A matrix D containing the distances between connected nodes is generated. $D(i,j)$ is defined as follows

$$D(i,j) = \begin{cases} 0.5 d_{ij}, & \text{if edge } ij \text{ is part of a passageway or staircase} \\ d_{ij}, & \text{if edge } ij \text{ is not adjacent to a space} \\ \frac{d_{ij}}{f(\overline{\text{obj}_{\text{connected spaces}}})}, & \text{if edge } ij \text{ is shared by one or more spaces} \end{cases} \tag{24}$$

In which d_{ij} is the Manhattan distance between nodes i and j, defined as:

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{25}$$

And $\text{obj}_{\text{connected spaces}}$ is defined by Eq. (26) in which connected spaces are represented by subscripts $1:n$.

$$\text{obj}_{\text{connected spaces}} = [max(100, \frac{AA_1}{RA_1} * 100), \dots, max(100, \frac{AA_n}{RA_n} * 100)] \tag{26}$$

Because spaces do not always meet their required area, and it is preferred that as little area as possible is lost due to carving additional passageways, an exponential function $f(\overline{\text{obj}_{\text{connected spaces}}})$ is used, and is defined by Eq. (27). The growth factor gf used in this paper is 0.06.

$$f(\overline{\text{obj}_{\text{connected spaces}}}) = (1 + gf)^{\overline{\text{obj}_{\text{connected spaces}}} - 100} \tag{27}$$

3. A matrix E containing information which side of a wall is preferred to place an additional passageway, if this wall is selected. The preferred side is determined based on the objective value of the spaces sharing walls. Matrix E is not utilised in the routing, but only to make the final decision where to carve, i.e. from which spaces to take area if necessary. Spaces that meet their required area are more likely to be carved from, than spaces that are too small.
4. Each connectivity check is performed by routing between a pair of systems, e.g. passageway and space, using a Dijkstra's shortest path algorithm.
5. At the location of each found path an additional passageway is carved.

A visual explanation of this approach is provided in Fig. 11. Fig. 11(c) shows that, after carving an additional passageway, two spaces do not meet their required area anymore. An overall readjustment of the allocation of spaces, position of staircases, and arrangement of spaces might lead to an improved arrangement.

The performance of the carving method will be investigated in the final test case using the integrated methodology presented in Section 5.

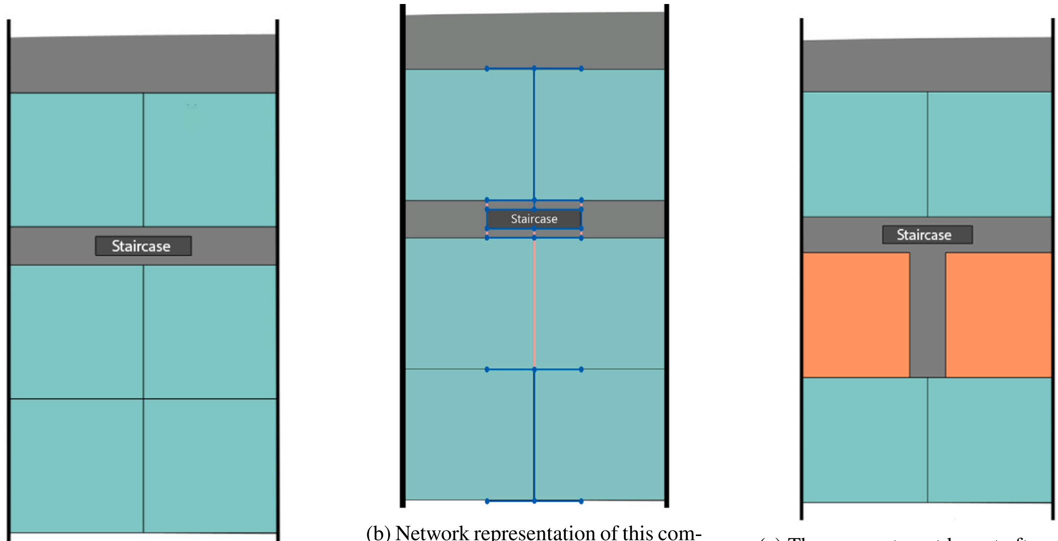
4.8. Optimisation problem

The previous Sections described the major mathematical methods in WARGEAR. In Fig. 12 a flow chart of the *arrangement* phase of the tool is presented. This section will elaborate how a nested optimisation approach is used in WARGEAR to steer the arrangement of detailed arrangements. A nested optimisation (or bi-level) approach is an operation research technique to solve hierarchical decision-making problems (Oduguwa and Roy, 2002), simplifying a large optimisation problem into smaller optimisation problems. In WARGEAR, the outer optimisation loop steers the placement of passageways, staircases, and the allocation of spaces to compartments. The inner optimisation loop controls the arrangement of spaces in individual compartments. The optimisation algorithms provides variables for the various elements of WARGEAR. These elements (e.g. the staircase arrangement algorithm) use the provided variables to choose from feasible options (e.g. choose from feasible staircase positions).

The optimisation problem for the outer optimisation loop is defined as follows. The objective function F for a complete layout is given by Eq. (28).

$$\text{Minimise } F = \sum_{i=1}^{n_{space}} max(0, RA_i - AA_i) \tag{28}$$

All constraints, for instance that spaces should be within allowed aspect ratios and space positions should be within the ship's hull and



(a) Compartment with six spaces and a staircase. All spaces meet their required area (indicated by green).

(b) Network representation of this compartment. Note that edges and nodes at bulkheads (left and right) are removed. Red lines indicate edges that need to be carved to ensure connectivity.

(c) The compartment layout after carving required additional passageways. Two spaces don't meet their required area now (indicated by orange).

Fig. 11. Visual explanation of the carving process for a compartment with six spaces and a staircase.

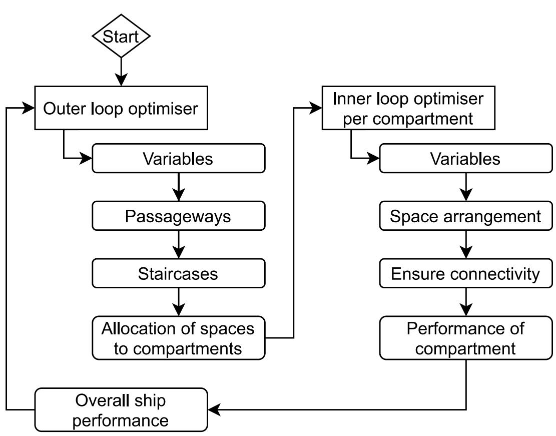


Fig. 12. Organisation of WARGEAR's mathematical methods inside a nested optimisation approach.

superstructure, are controlled by the WARGEAR code. Variables $x_k \in [0,1]$, generated by the optimisation algorithm, are used to select from feasible options for decision k. For instance, if a space can be allocated to four compartments, a variable is used to select between one to four only.

Variables x_k are used for:

1. Passageways
2. Staircases
3. Allocation of spaces to compartments

The optimisation problem for the inner optimisation loop is defined as follows. The objective function F_j for compartment j is given by Eq. (29).

$$\text{Minimise } F_j = \sum_{i=1}^{n_{space,j}} max(0, RA_i - AA_i) \tag{29}$$

Variables x_j are used for space arrangement only, as the algorithm used to ensure connectivity does not use variables.

The best arrangements of individual compartments generated in the inner optimisation loop, are combined into a single layout for each attempt in the outer optimisation loop. For instance, referring to the case study presented in Section 5 and Table 5, a total of $(NumIt + 1) * PopSize = (20 + 1) * 10 = 210$ arrangement attempts will be made by the outer optimisation loop. For each attempt the inner optimisation loop will arrange each compartment in $(5 + 1) * 3 = 18$ attempts. Only the best arrangement of each of these 18 attempts will be used to construct a full layout. Therefore, only 210 layouts will be presented to the naval architect. In contrast, suppose six compartments need to be arranged. Then, evaluating each possible combination would result in $210 * 18^{n_{\text{compartments}}} = 210 * 18^6 = 7.14 * 10^9$ layouts, but yield no more insights into layout sizing and integration issues.

The optimisation algorithm used for both the inner and outer optimisation is a Particle Swarm Optimisation (PSO). A PSO has been chosen because it was readily available and is easy to use. Also it was found that the PSO implementation used in WARGEAR can find sufficiently feasible solutions in limited time. Hence, no effort was spent in finding a more sophisticated optimisation algorithm. Although better performing algorithms might find slightly better solutions faster, it is not expected that using a more sophisticated optimisation algorithm will lead to better or more insights into space sizing and integration issues, which is the principle goal of WARGEAR. Refer to (Coello et al., 2004; Eberhart and Shi, 2001; Kennedy, 2010; Poli et al., 2007) for a more detailed description of PSOs.

4.9. Post processing and visualisation

After generating a set of layouts, the results are presented to the naval architect for further analysis. First, a scatter plot of the scores of all generated layouts is shown. Scatter plots were found to be a simple and familiar way of visualisation of the relation between two (or more) characteristics of a set of designs (Duchateau, 2016). Second, the naval architect selects a design from the scatter plot for detailed analysis. For an example, see Section 5.

To allow for quick recognition of the context of the spaces arranged by WARGEAR, the functional arrangement is used as a basis for the visualisation (Le Poole et al., 2020). All functional blocks from the functional arrangement that are not further detailed by WARGEAR are

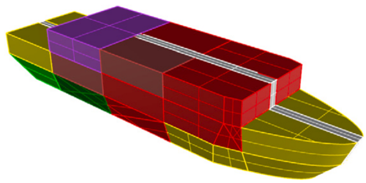


Fig. 13. 3D visualisation of a notional surface vessel.

plotted using the same colours as the functional arrangement. This helps identify areas in a compartment that cannot be used for space arrangement, such as exhaust stacks.

To support direct insight into the quality of the arrangement, spaces arranged by WARGEAR are coloured in accordance with the objective value of that space (Le Poole et al., 2019). Green is used to indicate that a space meets its required area, while red shades indicate to which extent a space did not meet its required area.

Also, main space properties, such as space name, number, required area, and achieved area, are provided in the detailed layout plan. Especially the information on required and achieved area supports the naval architect in better understanding to which extent a space did not meet its required area (Le Poole et al., 2020). It was considered to provide a measure based on $\frac{AA}{RA}$ instead of providing both RA and AA. Such fraction would provide quick insight, but is limited in giving insight in absolute numbers. For instance, consider three spaces with a required area of 10, 20, and 40 m^2. Assume these spaces have an achieved area of respectively 8, 18, and 32 m^2. Then, a fraction-based measure would yield 0.8, 0.9, and 0.8 respectively. This result already tells the second space has been best arranged, despite it lacks 10% of its required area. However, the result does not convey the serious problems with the third space, as it misses 8 m^2 in absolute numbers, which might be harder to be solved. Therefore, both RA and AA are provided, as these give insight in both relative as absolute (possible) lack of required area. As a result, the naval architect will likely start to address the arrangement issue of the third space, using the detailed arrangement together with the detailed textual and numerical information provided by WARGEAR.

5. Case study

This section elaborates on the generation of detailed layouts for a notional surface vessel. The aim of this case study is twofold (Pedersen et al., 2000):

1. To demonstrate and test the integrated mathematics of the WARGEAR method.
2. To demonstrate how the WARGEAR method can generate and analyse detailed layouts and to derisk functional arrangements.

The functional arrangement of the notional surface vessel is shown in Fig. 13. Note that this vessel does not reflect a particular naval vessel or contain systems such as engines or weapon systems. The main reason is that WARGEAR would not be used to arrange these systems. As such, adding such systems would not add to this case study. In this case study the arrangement of cabins, galley, mess, and stores will be investigated. The deck plans of the functional arrangement, available area in each compartment and functional building block, and the full space list can be found in the repository linked in Section 7. Note that based on the available area in the compartments, the required area of spaces, and the initial allocation of spaces to compartments, all spaces should fit. WARGEAR will be used to check whether all spaces indeed fit when staircases and possible local additional passageways are taken into account.

Table 5
PSO settings.

	PSO 1	PSO 2	Explanation
NumIt	20	5	Number of iterations
PopSize	10	3	Population size
w	0.5	1	Inertia weight
w_{damp}	0.9	0.4	Inertia Weight Damping Ratio
c1	0.5	2	Personal Learning Coefficient
c2	2.5	2	Global Learning Coefficient

Table 6
Summary of results of the two cases.

Test number	A	B	#B	A| B [m^2]	Run time [s]
	Minimum F obtained [m^2]	Minimum number of spaces not-allocated			
1	12.60	0	163	12.60	737.05
2	3.69	0	210	3.96	1065.41

5.1. Case study settings

The case study consists of two cases. Insights gained from the layouts generated by WARGEAR in the first case will be used to update the input for the second test. In the supplementary data, see Section 7, the first case is used in parallel with two additional cases to validate the combination of the space arrangement and passageway carving approaches in the context of the complete methodology. The settings for the optimisation algorithm are kept constant in each case, as these prove to lead to optimisation convergence and generate useful layouts, as further discussed in Table 6 and Section 5.3 . Table 5 summarises the settings for the optimisation algorithm. Note that the inner optimisation loop might not run towards convergence. This was deliberately chosen to reduce overall calculation time, and yields sufficient results.

5.2. Initial results

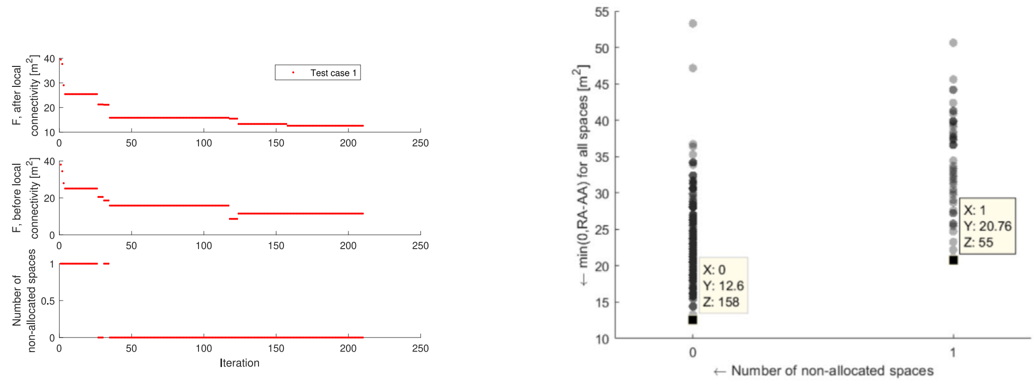
The results of the first case are summarised in Fig. 14(a) and Table 6. The graph shows the convergence of the F-value, Eq. (30), across all iterations. The objective function is given in Eq. (30). Also the number of non-allocated spaces, the average compartment area utilisation, and the objective value F before local connectivity is ensured are shown for the best performing layout.

$$F = \sum_{i=1}^{n_{space}} \max(0, RA_i - AA_i) \tag{30}$$

Next, two layouts generated in case 1 will be compared to identify possible sizing and integration issues. Indeed, in all tests the allocation of one space appeared to be challenging, pointing to possible issues. This might be caused by the available area in compartments, the initial allocation of spaces to functional building blocks, and/or required space size. Furthermore, no layout has yet been generated that meets all spatial requirements for all spaces. Fig. 14(b) shows a scatter plot with all layouts generated in case 1, and their objective score (y-axis) and the number of non-allocated arranged spaces. The two layouts that will be evaluated are (1) the overall best performing layout (ID 158, Fig. 15(a)) and (2) the best performing layout that misses one space (ID 55, Fig. 15(b)), as indicated in the scatter plot.

The following observations can be made:

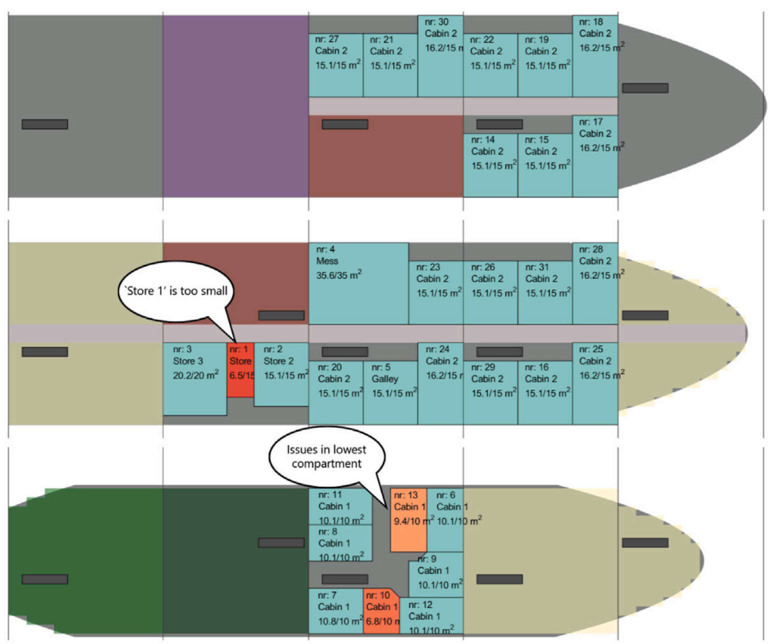
1. The local passageway carving method ensured connectivity on the lower deck in both Figs. 15(a) and 15(b). On the other decks, connectivity was already in place after space arrangement, since all spaces were directly accessible from the main passageway. Although this demonstration is limited in size, it shows that the carving method works as intended.



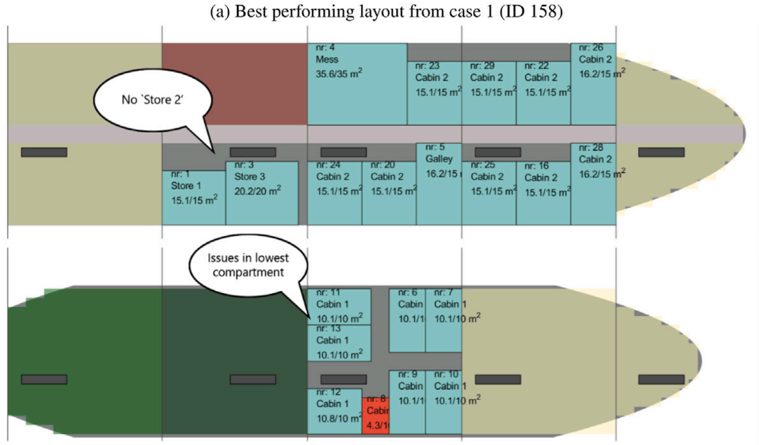
(a) Convergence and other data for the initial case.

(b) Scatter plot of case 1. Each dot represents a layout. The two selected layouts are discussed in this paper.

Fig. 14. High level case study results of case 1.



(a) Best performing layout from case 1 (ID 158)



(b) Lowest two decks of the best performing layout that misses one space from case 1 (ID 55). The upper deck is arranged similarly as layout 158, Figure 15a.

Fig. 15. Two selected layouts from case 1.

2. The space that could not be allocated in Fig. 15(b) is 'Store 2' on the middle deck. The starboard arrangement of a staircase in the second compartment from aft results in insufficient available space to allocate this store.

3. If 'Store 2' is successfully allocated, it is still not possible to properly arrange the three stores in one compartment, as can be seen in Fig. 15(a). This insight could be used to question the spatial requirements for the stores.
4. Although the allocation of spaces to compartments is not completely equal for the two layouts under consideration, most compartments have been arranged similarly. The main difference is found in the arrangement of the lowest arranged compartment. It appears that in Fig. 15(a) this compartment is arranged less realistically than in Fig. 15(b). Also, it seems that one cabin of 10 m^2 cannot be properly arranged.
5. The discretisation of the functional arrangement into positioning matrices with a grid size of 0.6×0.6 m^2 results in quite substantial differences between the predefined deck shape in the functional arrangement and the detailed layout generated by WARGEAR, see Figs. 15(a) and 15(b). In reality the available area will also be smaller due to structural elements and insulation along bulkheads and hull. However, this level of detail is not considered in WARGEAR.

The results of the quick investigation of two layouts lead to the following proposed changes to the input for case 2:

1. One cabin with a required area of 10 m^2 will be removed from the space list.
2. The required area of 'Store 3' is reduced to 15 m^2, such that three similar stores will be arranged in the notional surface vessel.
3. The grid resolution will be increased to a grid size of 0.3×0.3 m^2, to increase the resemblance of the functional arrangement in the detailed layouts.

5.3. Results from updated input

The proposed changes to the input of case 2 yield the following results, see Table 6 and Fig. 16.

1. The increased resolution by a factor of four of the positioning matrices lead to an increase of the running time (1065 s) by a factor of 1.45 (compared to 737 s for case 1).
2. No more issues regarding allocation of spaces have been encountered.
3. The best performing layout has already been found in iteration 58 of 210, which indicates that the PSO is a sufficient optimisation algorithm for WARGEAR.
4. The three stores can be properly arranged in one compartment.
5. Six spaces do not meet their required area by 0.4 m^2, but this can be easily corrected by the naval architect as sufficient space is available in corresponding compartments to slightly adjust the shape of other spaces to create additional useful area.

5.4. Conclusions from case study

The main aim of the case study into a notional surface vessel presented above was used to demonstrate and test the integrated mathematics of the WARGEAR method. The results showed that WARGEAR functions as expected, i.e. the integration of the several pieces presented in this paper is successful.

Secondly, the case study aimed to demonstrate the usefulness of WARGEAR by studying two generated layouts in more detail to identify possible layout sizing and integration issues. This was indeed possible, and led to two changes in requirements of the notional surface vessel and one change of the run settings. These changes led to a detailed layout that met almost all spatial requirements and can be used by a naval architect as a starting point for GAP development. These results show that WARGEAR is indeed a useful addition to the naval architect's tool set to reduce the risk of spatial requirements earlier in the design process.

6. Discussion, conclusions and future research

6.1. Discussion

This paper elaborated on an automated design tool, WARGEAR. Explicitly, WARGEAR builds on a human-defined functional arrangement, and is intended to be used concurrently to use of human-controlled tools during early stage design. The reason for human involvement is that one of the key goals of generating concept designs is requirements elucidation (Andrews, 2011; Van Oers et al., 2018). Since ship design inherently involves many stakeholders (e.g. users, naval architects, design specialists), alignment and balancing of preferences, requirements and budget forms an important part of early stage design (Van Oers et al., 2018). Hence, human control is essential as many design changes follow after considerable deliberation of many (non-)technical aspects (Van Oers et al., 2018; Andrews, 2018). As such, a concept design also represents the status quo of negotiated design aspects. While automated tools might be used to provide more detailed information and insights in the current concept design (e.g. WARGEAR, but also other detailed performance prediction tools), the human is still required to interpret this information and insight and to decide whether and which design changes are required. Careless use of design tool output can lead to significant design errors, requiring costly design rework or, in the worst case, a 'bad' ship to be launched (Andrews, 2021). Hence, WARGEAR is clearly not intended to replace manual design tools. First, WARGEAR only considers a limited number of design aspects, while all design aspects need to be considered. Second, the human is responsible for ultimate design decisions, which requires understanding, rather than accepting, of tool output.

6.2. Conclusion

Concluding, this paper has presented a detailed explanation of a novel ship layout design tool, named WARGEAR. WARGEAR is aimed to help naval architects to derisk lower level of detail layouts earlier in the design process, by rapidly generating high level of detail layouts. The level of detail of the generated layouts is space level without furniture. The tool is automated, to allow for near-real time feedback to the naval architect. This is especially important to support the highly iterative nature of early stage ship design. WARGEAR utilises a predefined functional arrangement as a starting point for detailed arrangements, to ensure compliance to overall design considerations, such as style and operational practices.

The mathematical working principles are based on probability theory, network theory and cross-correlation. Based on an investigation of naval architect developed general arrangement plans and experiments a set of generalised rules have been developed. These rules are believed to enable WARGEAR to generate layouts for a wide range of concept designs, such that spaces fit and connectivity is ensured, both on a global ship level (e.g. staircases and passageways) and compartment level (i.e. all spaces connect to main passageways and/or staircases). The order in which systems (e.g. passageways, staircases, spaces) are arranged is determined based on available area in compartments, required area for systems while maximising the probability that subsequent arranged spaces fit. Hence, larger, high impact systems are arranged prior to smaller, low impact systems, in such a way that as much useful area is left for these subsequently arranged systems. Also, designer's preferences on system positioning is taken into account.

A case study was presented to demonstrate and test the integrated method, as well as to assess the tool's usefulness. The results indicated that WARGEAR is able to generate detailed layout in a matter of minutes, and that these layouts can be used to identify sizing and

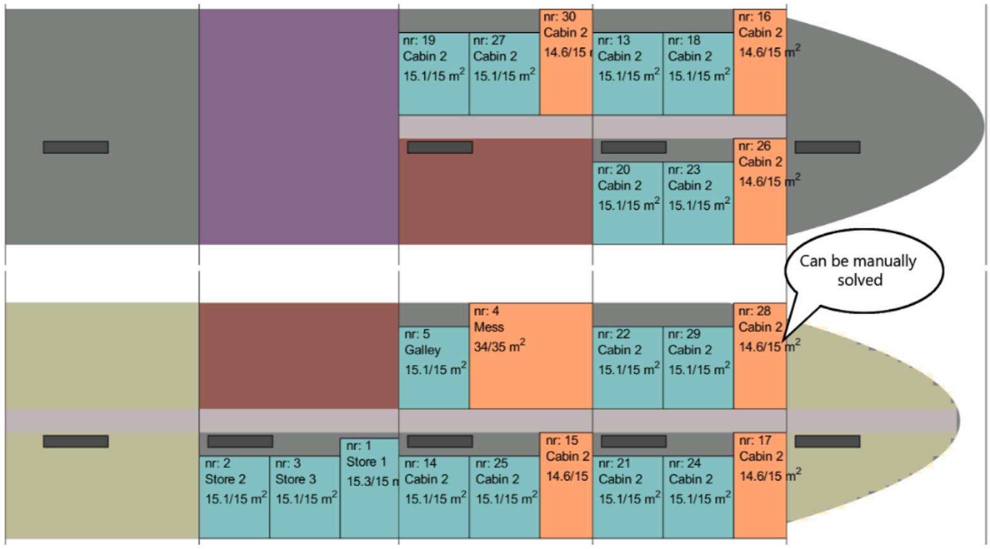


Fig. 16. Upper two decks of the best performing layout from case 2 (ID 58). The lowest deck was properly arranged.

integration issues. Further, WARGEAR is able to show how changes to requirements impact the quality of the resulting layouts.

In conclusion, WARGEAR is able to provide meaningful contributions to the early stage design of ship layouts, by providing early and rapid insight into possible sizing and integration issues. Thus, WARGEAR helps naval architects develop layouts at a higher level of certainty about technical feasibility.

6.3. Future research

Future work will focus on the application of WARGEAR in the actual ship design process. More specifically, the application of (near) real-time layout generation tools in interactive, real-time design processes involving multiple stakeholders and design disciplines will be investigated. This will help inform and align stakeholders earlier in the design process and thus support both the elucidation and refinement of requirements as the development of agreed upon concept designs. A key aspect will be the application of design rationale as a means to capture stakeholder preferences and use this design rationale to improve the quality of layouts (DeNucci, 2012; Le Poole et al., 2020). This application of real-time layout generation tools and reuse of design rationale in the context of collaborative decision-making is expected to contribute to common understanding and add to negotiations on conflicting and competing design issues. Ultimately this supports requirements elucidation and concept design generation during early stage ship design efforts.

7. Disclaimer, supplementary data availability

The content of this paper is the personal opinion of the authors. Specifically, it does not represent any official policy of the Netherlands Ministry of Defence, the Defence Materiel Organisation, or the Royal Netherlands Navy. Furthermore, the results presented here are for the sole purpose of illustration and do not have an actual relation with any past, current or future warship procurement projects at the Defence Materiel Organisation.

The data underlying the case study, as well as detailed tests of the methodology presented in the paper can be found in the following repository: https://doi.org/10.4121/19106903. Due to confidentiality, source code of the tools used in this paper is not openly available. Access to the code may be granted for research and educational purposes. This is subject to written permission from the authors, the Delft University of Technology, and the Defence Materiel Organisation of the Netherlands Ministry of Defence.

CRediT authorship contribution statement

Joan le Poole: Conceptualisation, Methodology, Writing. **Etienne Duchateau:** Conceptualisation, Review. **Bart van Oers:** Conceptualisation, Review. **Hans Hopman:** Review. **Austin A. Kana:** Conceptualisation, Review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the Netherlands Defence Materiel Organisation for their financial and in-kind support of the WARGEAR project. Further, the extensive comments of the reviewers have been gratefully received.

References

Andrews, D., 2003. A creative approach to ship architecture. Trans. Royal Inst. Nav. Archit. Int. J. Marit. Eng..
Andrews, D., 2011. Marine requirements elucidation and the nature of preliminary ship design. IJME 153 (Pt A1), 23.
Andrews, D., 2012. Art and science in the design of physically large and complex systems. Proc. Royal Soc. A Math. Phys. Eng. Sci. 468 (2139), 891–912.
Andrews, D., 2018. The sophistication of early stage design for complex vessels. Trans. Royal Inst. Nav. Archit. A: Int. J. Marit. Eng. 1–54.
Andrews, D., 2021. Design errors in ship design. J. Mar. Sci. Eng. 9 (1), 34.
Andrews, D., Dicks, C., 1997. The building block design methodology applied to advanced naval ship design. In: Proceedings of the International Marine Design Conference, Newcastle, United Kingdom. pp. 3–19.
Andrews, D., Pawling, R., 2008. A case study in preliminary ship design. Trans. Royal Inst. Nav. Archit., RINA, Int. J. Marit. Eng. 150 (Part A3), 2008.
Baušys, R., Pankrašovaite, I., 2005. Optimization of architectural layout by the improved genetic algorithm. J. Civ. Eng. Manag. 11 (1), 13–21.
Bourke, P., 1996. Cross correlation. Auto Correlation—2D Pattern Identification.
Brown, D., 1987. The architecture of frigates. In: Proc. Warship 1987: Int. Symp. on ASW, London. RINA.
Burrus, C.S., Parks, T., 1985. Convolution Algorithms. Citeseer, Citeseer: New York, NY, USA.

Camozzato, D., 2015. A Method for Growth-Based Procedural Floor Plan Generation (Ph.D. thesis). Pontifícia Universidade Católica do Rio Grande do Sul.

Carlson, C., Fireman, H., 1987. General arrangement design computer system and methodology. Nav. Eng. J. 99 (3), 261–273.

Coello, C., Pulido, G., Lechuga, M., 2004. Handling multiple objectives with particle swarm optimization. IEEE Trans. Evol. Comput. 8 (3), 256–279.

Daniels, A., Tahmasi, F., Singer, D., 2009. 'Intelligent ship arrangement (ISA) methodology improvements and capability enhancements. In: Proceedings of International Marine Design Conference, Trondheim, Norway.

DeNucci, T., 2012. Capturing Design: improving Conceptual Ship Design Through the Capture of Design Rationale (Ph.D. thesis). Delft University of Technology.

Droste, K., Kana, A., Hopman, J., 2018. Process-based analysis of arrangement aspects for configuration-driven ships. In: Marine Design XIII, Volume 1: Proceedings of the 13th International Marine Design Conference (IMDC 2018), June 10-14, 2018, Helsinki, Finland. CRC Press, p. 327.

Droste, K., Le Poole, J., 2020. Integrating detailed layout generation with logistic performance assessment to improve layout insights in early stage warship design. In: 15th International Naval Engineering Conference, 5-9 October, 2020.

Duchateau, E., 2016. Interactive Evolutionary Concept Exploration in Preliminary Ship Design (Ph.D. thesis). Delft University of Technology.

Duchateau, E., de Vos, P., van Leeuwen, S., 2018. Early stage routing of distributed ship service systems for vulnerability reduction. In: Marine Design XIII, Volume 2. CRC Press, pp. 1083–1096.

Eberhart, R., Shi, Y., 2001. Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Vol. 1. IEEE, pp. 81–86.

Ferreiro, L., Stonehouse, M., 1991. A comparative study of US and UK frigate design. SNAME Trans. 99, 147–175.

Gillespie, J., 2012. A Network Science Approach to Understanding and Generating Ship Arrangements in Early-Stage Design (Ph.D. thesis). University of Michigan.

Gillespie, J., Daniels, A., Singer, D., 2013. Generating functional complex-based ship arrangements using network partitioning and community preferences. Ocean Eng. 72, 107–115.

Guo, Z., Li, B., 2017. Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system. Front. Archit. Res. 6 (1), 53–62.

Inoue, M., Takagi, H., 2008. Layout algorithm for an EC-based room layout planning support system. In: 2008 IEEE Conference on Soft Computing in Industrial Applications. IEEE, pp. 165–170.

Kana, A., Shields, C., Singer, D., 2016. Why is naval design decision-making so difficult? In: Warship 2016: Advanced Technologies in Naval Design, Construction, & Operation, 15-16 June 2016, Bath, UK.

Kennedy, J., 2010. Particle swarm optimization. In: Encyclopedia of Machine Learning. Springer, pp. 760–766.

Keuning, J., Pinkster, J., 1995. Optimisation of the seakeeping behaviour of a fast monohull. In: Fast'95 Conference.

Le Poole, J., 2018. Integration of aboard logistic processes in the design of logistic driven ships during concept exploration: Applied to a landing platform dock design case. (M.Sc. Thesis). Delft University of Technology.

Le Poole, J., Duchateau, E., Van Oers, B., Hopman, J., Kana, A., 2019. Semi-automated approach for detailed layout generation during early stage surface warship design. In: Proceedings of the International Conference on Computer Applications in Shipbuilding, Issue 3, Rotterdam, the Netherlands. pp. 37–48.

Le Poole, J., Duchateau, E., Van Oers, B., Kana, A., 2020. A case study into an automated detailed layout generation approach in early stage naval ship design. In: 15th International Naval Engineering Conference, 5-9 October, 2020.

Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., Tao, B., Xu, S., Liu, H., 2019. Hand gesture recognition based on convolution neural network. Cluster Comput. 22 (2), 2719–2729.

Lipowski, A., Lipowska, D., 2012. Roulette-wheel selection via stochastic acceptance. Physica A 391 (6), 2193–2196.

Lo, S., Lou, S., Lin, J., Freedman, M., Chien, M., Mun, S., 1995. Artificial convolution neural network techniques and applications for lung nodule detection. IEEE Trans. Med. Imaging 14 (4), 711–718.

Lopes, R., Tutenel, T., Smelik, R., De Kraker, K., Bidarra, R., 2010. A constrained growth method for procedural floor plan generation. In: Proc. 11th Int. Conf. Intell. Games Simul. Citeseer, pp. 13–20.

Marson, F., Musse, S., 2010. Automatic real-time generation of floor plans based on squarified treemaps algorithm. Int. J. Comput. Games Technol. 2010, 7.

Mavris, D., DeLaurentis, D., 2000. A probabilistic approach for examining aircraft concept feasibility and viability. Aircr. Des. 3 (2), 79–101.

Medjdoub, B., Yannou, B., 2000. Separating topology and geometry in space planning. Comput. Aided Des. 32 (1), 39–61.

Merrell, P., Schkufza, E., Koltun, V., 2010. Computer-generated residential building layouts. In: ACM SIGGRAPH Asia 2010 Papers. pp. 1–12.

Michalek, J., Choudhary, R., Papalambros, P., 2002. Architectural layout design optimization. Eng. Optim. 34 (5), 461–484.

Najafi, M., Faraji, S., Bazargan, K., Lilja, D., 2020. Energy-efficient pulse-based convolution for near-sensor processing. In: IEEE Intern. Symp. on Circuits and Systems. ISCAS.

Nick, E., 2008. Fuzzy Optimal Allocation and Arrangement of Spaces in Naval Surface Ship Design (Ph.D. thesis). University of Michigan.

Oduguwa, V., Roy, R., 2002. Bi-level optimisation using genetic algorithm. In: Proceedings 2002 IEEE International Conference on Artificial Intelligence Systems. ICAIS 2002, IEEE, pp. 322–327.

Pawling, R., 2007. The Application of the Design Building Block Approach to Innovative Ship Design (Ph.D. thesis). UCL (University College London).

Pawling, R., Andrews, D., 2018. Seeing arrangements as connections: The use of networks in analysing existing and historical ship designs. In: Marine Design XIII. CRC Press, pp. 307–326.

Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J., Mistree, F., 2000. Validating design methods and research: the validation square. In: ASME Design Engineering Technical Conferences. pp. 1–12.

Peng, C., Yang, Y., Wonka, P., 2014. Computing layouts with deformable templates. ACM Trans. Graph. 33 (4), 1–11.

Poli, R., Kennedy, J., Blackwell, T., 2007. Particle swarm optimization. Swarm Intell. 1 (1), 33–57.

Pruinelli, L., Stai, B., Ma, S., Pruett, R., Simon, G., 2019. A likelihood-based convolution approach to estimate event occurrences in large longitudinal incomplete clinical data. In: 2019 IEEE International Conference on Healthcare Informatics. ICHI, IEEE, pp. 1–8.

Roth, M., 2017. Analysis of general arrangements created by the TU Delft packing approach making use of network theory. (M.Sc. Thesis). Delft University of Technology.

Takken, E., 2009. Concept design by using functional volume blocks with variable resolution. (M.Sc. Thesis). Delft University of Technology.

Van Oers, B., 2007. Development and implementation of an optimisation-based space allocation routine for the generation of feasible concept designs. In: 6th International Conference on Computer and IT Applications in the Maritime Industries. COMPIT'07.

Van Oers, B., 2011a. Designing the Process and Tools to Design Affordable Warships. Technical Report. NATO-RTO-MP AVT-173, Workshop on virtual prototyping of affordable military vehicles, Sofia, Bulgaria.

Van Oers, B., 2011b. A Packing Approach for the Early Stage Design of Service Vessels (Ph.D. thesis). Delft University of Technology.

Van Oers, B., Stapersma, D., Hopman, H., 2008. Issues when selecting naval ship configurations from a Pareto-optimal set. In: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. http://dx.doi.org/10.2514/6.2008-5886.

Van Oers, B., Takken, E., Duchateau, E., Zandstra, R., Cieraad, S., Van den Broek-de Bruijn, W., Janssen, M., 2018. Warship concept exploration and definition at The Netherlands defence materiel organisation. Nav. Eng. J. 130 (2), 63–84.

Joan le Poole is a Ph.D. candidate at Delft University of Technology in the field of early stage navel ship design, and focuses on the application of interactive design tools and the reuse of design rationale in multi-actor decision-making processes, applied to naval ship layout design. Previously he work as a researcher on the WARGEAR project. He completed his Bachelor and Master studies in Marine Technology at the same university in 2016 and 2018 respectively.

Etienne Duchateau holds the position of Sr. Naval Architect and Team Leader Ship Concept Design at the Maritime Systems Division of the Defence Materiel Organisation. Etienne is responsible for concept exploration studies for future warships for the Royal Netherlands Navy. In addition, he is involved with ship design tool development and research into early stage ship and fleet effectiveness simulations. In 2011 Etienne graduated as Naval Architect at Delft University of Technology. In 2016 he obtained his Ph.D. also at Delft University of Technology for his research on an interactive steerable concept exploration tool for early stage ship design.

Bart Van Oers has a Ph.D. in Ship Design (2011) and a MSc. in Ship Hydrodynamics (2005) from Delft University of Technology. He works for the Maritime Systems Division of the Netherlands Defence Materiel Organisation since 2009. Since 2020, Bart heads the Bureau of Lifecycle Modelling, which as the DMO's forward design group is responsible for conducting concept studies for future warships for the Royal Netherlands Navy including the estimation of their operational effectiveness and lifecycle cost. He can be contacted at: b.j.vanoers@xs4all.nl.

Hans Hopman is professor Ship Design and was head of the Department of Maritime and Transport Technology at Delft University of Technology. His research focuses on the improvement of the design process for complex ships and marine systems integration. He started his academic career in Delft in 2006 after having worked nearly 25 years for the Directorate Materiel Royal Netherlands Navy where he was involved in many naval procurement programmes.

Austin Kana is an Assistant Professor in the Department of Maritime and Transport Technology at Delft University of Technology. His research focuses on developing design methods and tools to help address early stage complex ship design problems, and is the daily supervisor of Joan le Poole. He received his Ph.D. in Naval Architecture and Marine Engineering at the University of Michigan in 2016.