AN INTERACTIVE LAYOUT EXPLORATION AND OPTIMISATION METHOD FOR EARLY STAGE SHIP DESIGN

B Igrec, **R Pawling, G Thomas,** University College London, UK
A Sobey, University of Southampton, UK
J Rigby, BMT, UK

SUMMARY

This paper presents a novel, highly interactive genetic algorithm-based layout exploration and optimisation method for generating spatial configurations of ships in the early stages of the design process. The method draws upon the principles of design-driven architecturally centred ship design processes by enabling the naval architects to make important decisions in a hybrid design process. The method utilises a genetic algorithm-based optimisation tool to rapidly generate and evaluate a diverse set of general arrangement options. It is approached in stages where each stage comprises two steps (manual and automatic).

The new genetic algorithm-based layout optimisation tool is demonstrated by being applied to an Offshore Patrol Vessel test case. The advantages and disadvantages of the proposed tool are discussed, as well as the current limitations of the overall approach and future work.

NOMENCLATURE

CP	Computational Domain
CPF	Continuous function Penalty Factor
DBB	Design Building Block
DSS	Design Solution Space
ESSD	Early Stage Ship Design
GA	General Arrangement
GAOT	General Arrangement Optimisation Tool
ISA	Intelligent Ship Arrangement
LCG	Longitudinal Centre of Gravity
MDW	Main Design Window
NA	Naval Architect
OC	Optimisation Criteria
OE	Entity to be Optimised
OPV	Offshore Patrol Vessel
PF	Penalty Factor
SPF	Step function Penalty Factor
VCG	Vertical Centre of Gravity
VSP	Void Space Penalty factor
WF	Weighting Factor

1. INTRODUCTION

Developing a suitable General Arrangement (GA) is a critical part of ensuring the successful design of all ships, from superyachts to aircraft carriers. How different spaces, such as bridges, crew quarters, control rooms etc. are sized and located is fundamental to guaranteeing safe and effective operations. It is a complex, time-consuming and inherently iterative part of the design process since many different, and very often conflicting, requirements have to be balanced within the constraints of the vessel while ensuring that all aspects of the operations and human factors are considered.

A less detailed, high-level GA is considered for the first time in the conceptual phase of the design process, referred here as Early Stage Ship Design (ESSD). The main purpose of this initial phase is to design several alternatives in a low level of detail, which are then considered to assess their cost and performance trade-offs with the aim of elucidating design requirements [1], [2].

Traditionally, ESSD has been conducted in phases where one follows another sequentially; in general, it is perceived as a linear process with discrete sequential steps and feedbacks reflecting its inherently iterative nature, and it is often represented by a design spiral [3], [4]. Compartments are being arranged within the fixed hull dimensions towards the end of the process, typically following initial sizing [5]. All changes and adjustments require the process to go from the beginning, leading, therefore, to a significant number of iterations to be carried out making the process time-consuming and expensive [6]. Moreover, this procedure often leads to a parent ship method where a new ship is designed based on the existing one (or a class of similar designs) that closely matches the performance requirements of a new design [7]. This process can be seen to narrow down the exploration of the Design Solution Spaces (DSS), limit innovation and the creative inputs from Naval Architects (NA) [8], thus, is not particularly suitable for designing novel ships with radically different features, unconventional, unique or nonstandard designs.

Andrews [9] argues that the design of warships and most of the other commercial service vessels [10] should be driven by their internal (and upper deck) configuration and that the concept design of these types of vessels should be approached by firstly configuring spaces focusing on the vessel's primary function followed by "wrapping" the hull around afterwards [11]. This leads to the "inside-out" design approach opposing to the traditional "outside-in" approach.

These inferences led to the function-based architecturally oriented ESSD to be introduced in the latter part of the twentieth century; centering the ship architecture and allowing all traditional ESSD numerical calculations to be performed concurrently while modifying the layout configuration [8], [12]. Andrews [5] concluded that fully integrative ship design synthesis that places greater emphasis on the physical description of the ship layout will foster more creativity and novelty by enabling architectural factors to impact major decision making and allow the main requirement drivers to be considered from the start [3].

Still, the manual nature of developing GAs makes the process laborious, requiring NAs to manually readjust the whole layout many times during the DSS exploration. Moreover, only one ship can be designed at the time, which makes the generation of multiple design concepts time and resource expensive. Hence, fully manual ESSD approaches do not readily lend themselves to rapid trade-off, cost-benefit and option analysis when multiple design alternatives are required to be synthesised and evaluated [13].

Challenging markets and tight budgets for building ships drive the need for developing advanced tools that would help ship designers to rapidly produce a diverse set of design options with an aim of enabling wider and more thorough concept exploration, while at the same time reducing the time and the cost of ESSD. Despite the significant developments in the current ESSD tools and software applications, a high degree of user inputs still prevents the extensive exploration of DSS, particularly in the case of configurationally driven service vessels [13].

The aim of this paper is to introduce a project carried out at UCL Mechanical Engineering in collaboration with the University of Southampton and BMT. Its overarching aim is the development of highly interactive layout exploration and optimisation method that can be applied to configurationally driven ships for generating GAs. The project focuses on naval service vessels during ESSD. The research aims to develop a method and optimisation tool that can be easily applied to any generic manual ESSD process without any sophisticated software or synthesis systems required as a prerequisite.

The problem is approached in stages where each stage is composed of two steps. First, the General Arrangement Optimisation Tool (GAOT) is utilised to assist the designer in developing and evaluating a diverse but rational set of alternative arrangements. Then NAs examine returned alternatives manually, make changes where necessary and return the modified designs back to the tool as the initial population (solution) for the next stage. The hybrid process goes on until the satisfactory solution has been found. The purpose of the tool is to help the designer to explore the design solution space by returning design alternatives after each stage, however, the major ship design calculations and all decisions are still performed manually or by coupling GAOT with other manually operated ESSD synthesis tool/s. GAOT integrates space and system interrelationships, global preferences, constraint management, and also considers certain basic naval architecture aspects such as stability.

2. OPTIMISATION METHODS TO ESSD

Previously discussed downsides common to all manual design approaches combined with rapid development in affordable computational power over the recent years have urged researchers to start developing methods and tools that can automatically produce many design options simultaneously. Early examples of such tools are ASSET [14], [15] and PASS [16], [17] where designs can be semi-automatically generated by altering relevant design variables. However, the optimisation of GAs of configurationally driven service vessels is a challenging and complex problem typified by excessively large DSS and a great number of conflicting objectives and constraints. Thereby, it is necessary to approach the problem, not only by developing fast and efficient algorithms, but also by proposing a method suitable to "optimise" the automation/(manual input from NAs) ratio so as to cope with the complexity of a given problem (i.e. improving the design procedure while reducing the overall ESSD time).

Existence of numerous, quite often subjective and not explicitly defined optimisation criteria such as 'close to', 'separated from', or 'more or less square' is an ideal environment for fuzzy logic [18] principles to be applied. The fuzzy optimisation approach, first proposed in [19], has been applied in many arrangement design efforts [20]–[22].

These principles have also been utilised by researchers from the University of Michigan. Intelligent Ship Arrangement (ISA) method – published in [23]–[27] – has been developed as an addition to the U.S. design process with the purpose to assist NAs in semi-automatically producing spatial arrangements in ESSD. The process starts by allocating spaces into predefined and fixed zone-decks during the first step followed by arranging each one separately. It works in a hybrid environment where a genetic algorithm developed by Nick [28] – encouraging solution diversity – is coupled with a multi-agent system that provides intelligent search capabilities. The method requires an intelligent product modelling system and a central data repository (LEAPS) [29] that has been introduced as the central, coordinating database for naval ship design information created as a part of the ASSET [30] naval ship synthesis system [27]. Therefore, ISA is compatible with the U.S. navy design process, however, it cannot be straightforwardly applied to other design processes.

Concurrently with, but separately from, these attempts from the U.S., van Oers [31], [32] proposed a new TU Delft (Delft University of Technology) Packing Approach. This utilises an object-driven semi-automatic 2.5D and 3D ESSD synthesis model. It relies upon parametric ship description and contains basic ship design aspects in a form of search algorithm constraints. It is performed in two steps; first, search algorithm – by utilising Non-dominated Sorting Genetic Algorithm NSGA-II [33] – generates a diverse set of feasible options by satisfying non-negotiable constraints on space and equipment location, aiming to obtain the maximum packing density, followed by manual selection of designs of interest [31]. The 2.5D version of Packing Approach is faster because it assumes simplified arrangements in the transverse direction, i.e. transverse objects are collapsed onto a centreline place which reduces object placement to two dimensions while keeping the accurate profile of the hull [34]. Later research focused on gaining insights from NAs by capturing design rationales [34], [35], introducing a new approach to allow NAs to guide the options generation [36] and applying the network theory in order to understand physical relationships in the GAs generated by the Packing Approach [37].

Nonetheless, the approach still fails to deal with the interdependences resulting from the relationships between spaces when configuration-driven ships are concerned, as concluded by Droste in the recent paper to IMDC 2018 [38]. Interactive steering introduced by Duchateau [36] is available only through manipulating with search algorithm objective function/s, while the packing rules and the search algorithm constraints stay hard-coded; which may limit the flexibility of the algorithm by hindering some of the potential benefits interactive feedback may bring.

Research at UCL has been focussed on integrating ship architecture into a holistic approach to ship synthesis, via the Design Building Block approach (DBB); first proposed by Andrews [5]. It is a graphical computer-aided holistic design method that applies principles of architecturally driven ESSD approaches in practice by combining numerical calculations with a CAD system. It has been fully realised after it was implemented into the SURFCON which comes as an additional module of the GRC ESSD tool PARAMARINE [39]. Since its development, the DBB approach has been successfully demonstrated many times and its utility and applicability has been reported in numerous publications (see [4], [8], [40]–[42]). Even though its benefits are undeniable, still, it is fully manual and requires continuous inputs from NA during DSS exploration allowing only one ship to be designed at the time.

3. SCIENTIFIC APPROACH

Early research carried out by this project focused on determining the scope and high-level features of the method. One approach was to adopt a fully automatic optimisation tool that would have more autonomy which, in order to balance the ship design, would necessarily mean addressing more than just arrangement. The result of such approach would be a set of feasible alternative ships from which the NA would choose preferred options.

However, the interactions between requirements, design solutions and cost are not known from the start; NA gains knowledge and insights during the concept exploration process. Thus, requirements may also change during the process led by the obtained knowledge and dialogues with the requirement owner causing major decisions to be also revised and the design re-worked. This is termed requirements elucidation by Andrews [1]. Accordingly, ESSD is extremely challenging to fully automate without the input of the designer in order to address these changes and newly gained insights.

Considering this key aspect, a novel hybrid approach to optimising arrangements is proposed that incorporates sufficient and appropriately implemented level of interaction between design team and GAOT throughout the whole ESSD, balancing therefore manual and automatic aspects. The method is intended to assist the intelligent user by means of GAOT rather than replacing them with the ‘intelligent’ optimisation tool that would theoretically be able to make all decisions autonomously based on inputs on the very beginning of the process.

3.1 PROPOSED METHOD

The highly interactive nature of the proposed method is outlined by the design triangle which is illustrated in figure 1. It represents the principal relations among three major factors: Main Design Window (MDW), design team and GAOT.

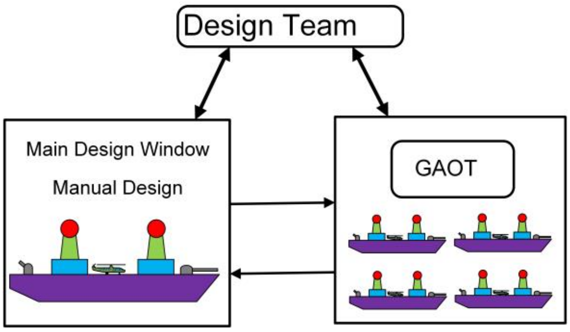

Figure 1: Design triangle

MDW typifies manually-driven design environment encompassing all necessary tools where all relevant numerical calculations are performed. MDW and GAOT have to be compatible with each other so as to allow the flow of information in both ways. Design team manages both of these systems. All tools from the MDW are manually operated and require continuous inputs and presence of NAs, whereas GAOT needs inputs only at the beginning and in between each optimisation stage.

The overall method is approached in stages where one stage is composed of two steps: manual and automatic. Each automatic step is performed by utilising genetic algorithm-based GAOT which outputs a set/s of design alternatives along with the substantial feedback about each returned design option. Returned feedback is deemed to be very important which is why a statistical analysis feature is intended to be a significant part of GAOT. The purpose of this feature is to give information that would help to read, differentiate and evaluate every design alternative. Some of the returned pieces of information are:

- The level of satisfaction of every optimisation criteria for the overall design as well as for each space to be optimised
- Statistical report that provides probability of a certain space to be found at the certain position considering all returned designs
- Rankings of options based on cumulative and selective evaluation functions
- The level of constraints violation for each OE, etc.)

Retuned designs are then manually analysed by the design team which performs additional calculations and returns modified set/s as the initial solution for the next hybrid stage.

Figure 2 depicts a high-level schematic of the method. It commences by gathering initial requirements and performing initial calculations in MDW prior to considering GA at all. Subsequently, these early insights are imported into GAOT in form of high-level blocks and set of key optimisation criteria. The main purpose of this initial phase – denoted as stage 0 in figure 1 – is to focus on the most important compartments and systems necessary for the ship to deliver its primary function and major routes. GAOT returns a certain number of overall GAs all of which contain the network of major access ways reflecting the arrangement of the first set of imported blocks.

NAs then manually consider each design alternative by means of MDW where additional numerical calculations are performed. Based on this, they modify and upgrade the returned set of designs by changing individuals separately, or combining pieces of different solutions together, which then forms the initial population for the next stage. Changes can be divided into the ones that change DSS and ones that don't. For instance, importing a new set of spaces, disassembling high-level blocks into constituent parts, fixing certain spaces, modifying requirements, Optimisation Criteria (OC) or Weighting Factors (WF) all necessarily mean changing the Computational Domain (CP), i.e. DSS. Therefore, the hybrid approach features dynamic solution space which reflects changes in the CD of certain individuals and also enables later optimisation stages to be performed in parallel. Therefore, individuals modified in a way which doesn't change the CD go into the next phase as a part of the solution forming the old population (figure 2), and the number of different variants of DSS dictate the number of new populations. This way allows for exploring multiple combinations of OC, objectives and constraints simultaneously mitigating the risk of excluding certain options and/or combinations of requirements too early in the process.

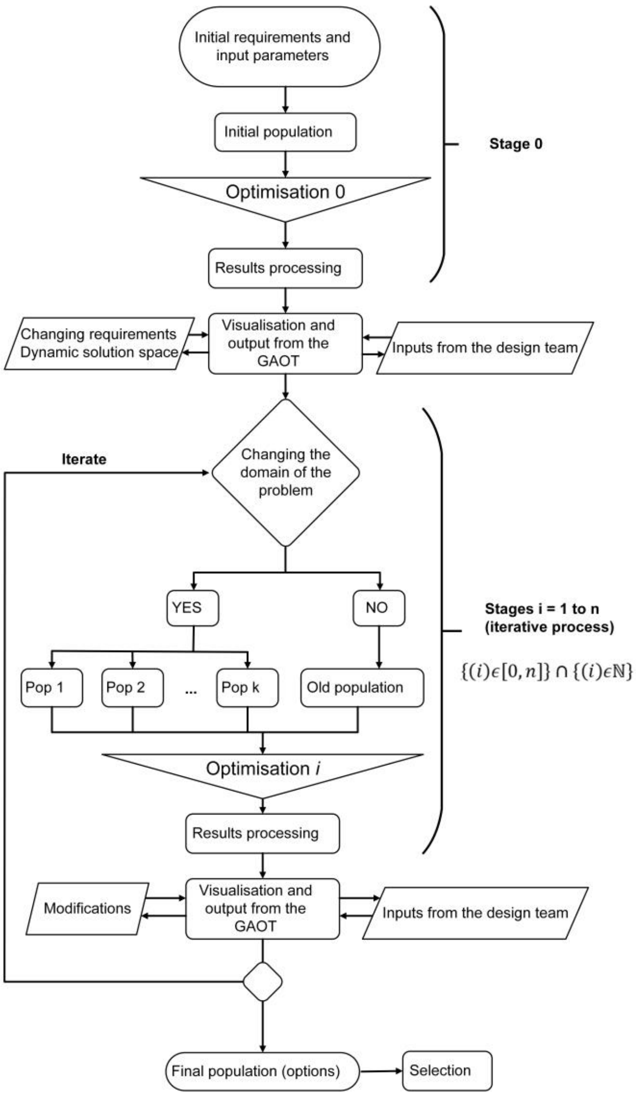

Figure 2: Schematic of the proposed method

Within a stage, every subprocess, representing a unique population, is considered by the algorithm separately, without crossing over (DSS is separately defined for each subprocess, i.e. population); all *k+1* subprocesses run in parallel forming the overall algorithmic process within a hybrid stage (figure 2). From the perspective of the algorithm, every subprocess can be performed in single or multi-objective manner, where fitness functions can be differently defined for every subprocess, and also, all subprocesses don't need to have the same number of algorithmic objectives. This means that various naval architecture objectives, combination of WFs and other OC can be differently combined and grouped to form fitness functions for different subprocesses; since there is generally a great number of naval architecture objectives and OC, each of which is also strengthened by a unique WF, all of them cannot be modelled as separate fitness functions, hence, some of them have to be combined within the same fitness function.

However, between stages, design team can combine solutions from all populations, add new ones or obliterate ones that are deemed not to be attractive any more. The number of stages (denoted with *n* in figure 1) is not predetermined; the process goes on until the satisfactory and limited set of design options has been found, which is then filtered manually.

Between stages Entities to be Optimised, (OE) that are deemed satisfactorily arranged can be fixed, which means they are excluded from the CD, or can be kept inside with revised WFs. OEs are all objects manipulated by the algorithm such as spaces, systems, compartments, passageways, etc. Strong WFs for global preferences allow only slight deviations from their previously found 'optimal' positions, shapes and areas. Typical examples for this are high priority entities arranged in the initial stage. This is schematically depicted in figure 3.

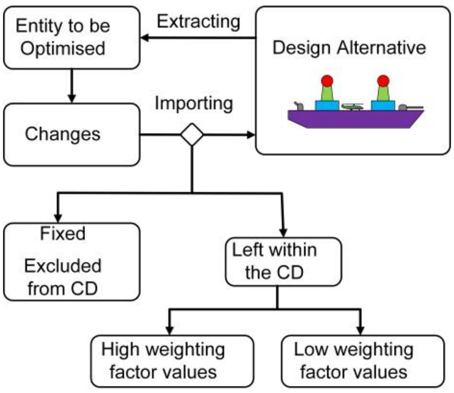

Figure 3: Types of changes to OEs

3.1 (a) Considering Major Components

Major routes and watertight bulkheads are approached in a slightly different way. Both are considered for the first time in the initial stage taking into account only high-level blocks. A network of major passageways is modelled as a collection of nodes which are related to arranged high-priority spaces. One approach implies fixing them after the stage 0, (exclude them from the CD), while another approach allows them to deviate from the original shape (by skewing and distorting the network) so as to adjust to more detailed GAs in successive stages. The level of adjustment (deviation from preferred routes) can be regulated by numerical values of WFs. Watertight bulkheads and hull dimensions follow the same principle. In that their deviation from a specified value can be regulated or limited by the user; these settings can be differently defined for different populations. Any prior decision/s can be reversed or changed in any of the succeeding stages.

3.2 LAYOUT OPTIMISATION TOOL

GAOT solver is based on a genetic algorithm method. Genetic algorithms, first introduced by John J. Holland [43], belong to the group of evolutionary algorithms [44]. It is a stochastic optimisation technique inspired by Darwin's principles of natural selection and survival of the fittest individuals. The solution converges towards the optimum by repeatedly modifying a population of individuals, represented by chromosomes. The process starts by creating an initial population that is then evaluated and ranked against an objective function. In the next step, the algorithm selects individuals from the current generation to be parents and uses them to produce children for the next generation (crossover and mutation operators are performed). During the selection, only the fittest individuals survive, while the rest of them are discarded by the algorithm. The population evolves towards an optimum solution over consecutive generations by repeating the above steps, imitating thereby biological evolution [45], [46]. The current version of GAOT is encoded within MATLAB programming language where MATLAB Genetic Algorithm Toolbox has been utilised [47], [48].

There are two types of constraints supported by the tool, hard and soft ones. Hard implies defining a constraint function where constraints are considered separate from objectives. In a soft constraint, penalty functions are used, where all constraints, along with other OC, are incorporated within the fitness function in form of Penalty Factors (PF) which are then minimised throughout the process. Two types of PFs are supported. Step function PFs (SPF) are modelled as a step function, where, if the constraint is violated, always the same penalty value is returned. The other option is to adopt sliding or Continuous function PFs (CPF), where, the more the constraint is violated, the higher the penalty value is. This way allows for soft modelling of all constraints; implemented penalty function calculates the degree of a constraint violation, and correspondingly returns the penalty factor value. This approach allows semi-feasible individuals to survive and take part in the crossover process; the algorithm gradually eliminates semi-feasible individuals through the selection improving, thereby, the chance of finding better solutions and converging towards promising results faster. On the contrary, if only hard constraints had been used, only feasible individuals (no constraint violation at all) would have been survived to the next generation, reducing population diversity and increasing the probably that the algorithm would diverge and fail to the inability to find any feasible individuals.

3.2 (a) Overlap Management

Overlap management secures rational and realistic geometrical properties of returned layouts by:

- Suppressing overlapping between OEs
- Suppressing overlapping between OEs and predefined entities
- Restraining all OEs to stay inside the CP, i.e. penalising OEs that exceed deck borders
- Penalising OEs that cross through watertight bulkheads (the network of main passageways is an exception)

Overlapping is addressed by penalising the overlapping area, however the use of a soft constraint means that small overlaps will be permitted if they lead to a more optimal overall arrangement.

3.2 (b) Modelling Constraints

The same way of modelling is adopted for constraints, objectives and other design rules, all of which are denoted as OC. They are modelled in a form of functions, where each function is characterised by a PF, WF and the function shape *f*, all of which can be specified by the user. All expressions are generally taking a form of *PF = WF * f(x')*, where *x'* is a decoded subset of design variables relevant for the given PF. Function shape enables defining a variety of different preferences for different OC. They can be 1D, 2D or even 3D fields, and also can be continuous or discrete.

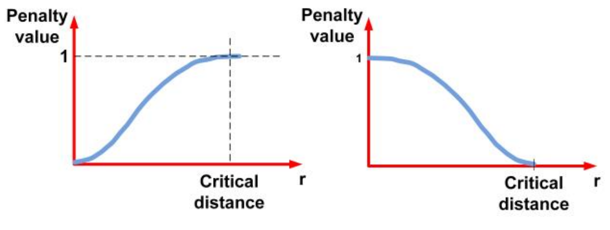

Figure 4: Example of penalty function shapes

For example, if a space should be located close to the stern, or other general longitudinal position, this can be addressed by defining a global position PF with respect to the *x* coordinate. On the other hand, if the certain compartment prefers to have a porthole (captain's cabin for instance), this can be addressed by appropriately defined global preference PF with the respect to y to ensure it is located outboard. Left hand-side of figure 4 shows the example of 1D continuous penalty function curve for space's global location preference. The function is normalised in a way that if the space is positioned at the target position, penalty value is zero, and it grows with the distance reaching the value of one for some critical distance. The right side of the same figure depicts an example of the penalty curve for the separation. Minimum separation in terms of Euclidean distances is prescribed to be minimum distance. Every distance between two considered compartments that is greater or equal to minimum distance is not penalised, otherwise, penalty value rises and reaches one if compartments are positioned next to each other. To reflect the fact that the ship comprises multiple decks, Manhattan distance system is also encoded into GAOT.

Generic function shapes are multiplied by WFs in order to address the importance of certain OC linked to certain OEs. WFs can take a value of zero or any other positive real number. Table 1 provides more detail on the constraints used in the prototype tool and proposed method.

3.2 (c) Combining Constraints

In general, all OC can be expressed as a combination of 2D matrices, vectors and scalars whose elements are PFs. PFs describing relations among OEs are stored in a 2D matrix A_{ij}, where both indices *i* and *j* correspond to OEs. Matrix A_{ij} enables capturing both influences entity *i* has on entity *j* and also entity *j* has on entity *i* (*i* to *j* and *j* to *i*). Since there can be multiple relationships between entities, matrix A_{ijt} stores *t* A_{ij} matrices where index *t* reflects the collection of relations driven by certain ship design aspect, i.e. overall ship OC such as survivability, maintainability, cost, accessibility, etc.

On the other hand, detail OC linked to the entity itself such as global preferences, relations between OEs and predefined entities or OEs and fixed objects are stored in penalty vectors B_i. Fixed objects can be, for instance, the propeller, bow, stern, mast or any other fixed *x*,*y*,*z* position in the CD. Matrix B_{im} stores *m* penalty vectors. Finally, certain OC such as LCG, trim, VCG or Void Space PF (VSP) are scalars, and *n* scalar PFs are stored in *n*-dimensional vector C_n. In general, all fitness functions are taking a form defined by equation 1.

$$F = \sum_t \sum_j \sum_i A_{ijt} + \sum_m \sum_i B_{im} + \sum_n C_n \qquad (1)$$

The basic principle is to add all relevant PFs to form one algorithmic objective function. For running the GAOT in a multi-objective manner, a number of fitness functions can be defined each of which is composed of differently defined and combined three terms from equation 1. PFs from overlap management are encompassed by the third term from equation 1. A high-level list of OC is summarised in table 1, while table 2 further breaks down OC expressing global and relational preferences for OE. Preferences can be linked to the entity itself or can be expressed as a relation between two entities.

<table>
<tr><th>Type of OC</th><th colspan="2">Description</th></tr>
<tr><td>Basic NA</td><td colspan="2">Intact stability, simple weight calculation, trim, VCG, LCG, hydrostatics (GM = KB + BM – KG), bulkheads are considered</td></tr>
<tr><td>Global preferences
Preferable X,Y,Z positions of compartments and systems</td><td>Global positions (target positions) of all compartments and systems on board</td><td rowspan="2">Accessibility
Operability (operation)
Survivability
Maintainability
Supportability
Costing
Noise, safety, comfort
Motions
Topological requirements
Functional dependencies (function)
Mission
Physical connections between systems – system runs – connectivity
Visibility</td></tr>
<tr><td>Compartment interdependencies and interrelationships</td><td>Relationships between all effected compartments and systems based on different optimisation criteria</td></tr>
<tr><td>Overlap penalty factors</td><td colspan="2">Overlapping between two spaces their self
Overlapping between the space and the system
Overlapping between empty space necessary for a certain system to work properly and any other object (for instance, radar arcs, required arcs for weapons etc.)
Overlapping between the deck perimeter and space
Overlapping between the main passageways and other objects in the model</td></tr>
<tr><td>Compartment shape</td><td colspan="2">The general shape of the compartment or system
length to the width aspect ratio
width to the length aspect ratio</td></tr>
<tr><td>Compartment size</td><td colspan="2">Area of compartments and systems (volume of the object can be calculated from the area – rough estimation of the volume)</td></tr>
<tr><td>Void space penalty factor</td><td colspan="2">Space utility measure. Defined as the ratio or difference between the overall hull deck area and all non-overlapping area occupied by entities inside the hull.</td></tr>
</table>

Table 1: High-level list of optimisation criteria

The structure of the GAOT is shown in figure 5. The user manipulates five levels of inputs from the MDW. The first level implies specifying all OEs and all predefined/fixed entities which are excluded from the CD. The algorithm varies areas, shapes and *x*,*y*,*z* positions of centroids of spaces by means of design variables. Two-dimensional horizontal *xy* coordinate system is originated at transom stern, where *x* coordinate axis is aligned with the centreline of the vessel facing towards the bow, while the *y*-axis is perpendicular to *x* and is oriented from starboard to port side of the deck. *X* and *y* variables are continuous, while *z* positions denote the deck on which the entity sits on and take only integers.

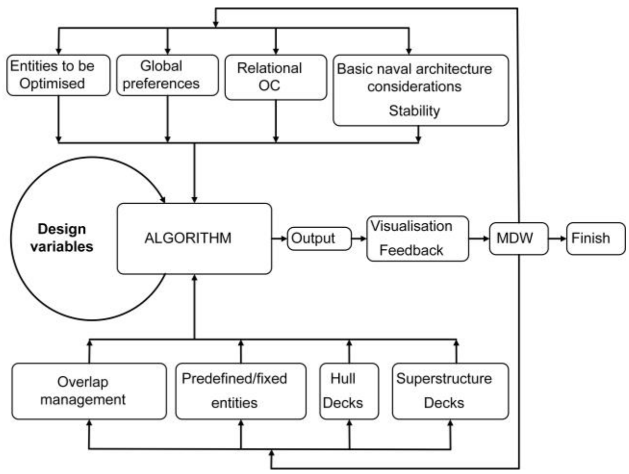

Figure 5: Structure of the GAOT

The second level of inputs encompasses specifying parameters for defining hull, watertight bulkheads and superstructure. The current version of the tool manipulates the shape of the superstructure taking into account the maximum number of superstructure decks, but doesn't support hull dimensions, the number of decks and bulkheads to be manipulated by the algorithm, hence, they have to be predefined. However, these can be manually changed for any of the parallel populations between two hybrid stages. WFs and the shape of penalty functions for global preferences, relational OC, overlap management and basic ship design considerations are specified within the third, fourth and fifth level of inputs.

All OC can be switched on or off. Parameters for all OC can be separately defined or can be imported from built-in databases. If any of elements of OC that is switched on is not specified, GAOT automatically assumes default settings from the database or assigns zero. The same principle is applied for deck shapes. They can be specified separately in the GAOT, or imported from the deck shape database. The custom way of defining all OC parameters, adding, removing or freezing OE allows for GAOT to be used in both a flexible approach to requirements elucidation, and in the less flexible mode when the design had been already locked by a specific and inflexible set of requirements given by the requirement owner.

Driving factor	Constraint	Type of constraint	Description
Topology related	Connectivity	Relational	Defines if two considered spaces need to be connected (share the same bulkhead)
	Adjacency	Relational	Defines if two affected spaces require to be positioned adjacent to each other or to other entity
	Separation	To space itself	For the given space, defines all separation requirements involving all entities of relevance
	Global location	To space itself	Defines a target position of a space
	Relative location	Relational	Defines preferred distance (min or max) between two spaces – can be modelled as the Euclidian or Manhattan distance
Geometry related	Area	To space itself	Defines target area of the space
	Volume	To space itself	Defines target volume for space
	Shape	To space itself	Defines shape and relevant aspect ratios
	Accessibility	Relational	Specifies requirements pertaining to access from one space to another
	Path	Relational	Specifies the restrictions on the path between two spaces (this can involve other spaces from the list)

Table 2: Global preferences and relational optimisation criteria

3.3 LIMITATIONS OF THE CURRENT VERSION OF THE TOOL

GAOT is still in development and only the proof of concept version of the tool is presented in this paper.

- The current version of the solver is based upon the MATLAB Genetic Algorithm Toolbox that originates from mid 1990s; the overall capabilities of GAOT are limited by the capabilities of the mentioned toolbox.
- Hull dimensions, number of decks and positions of watertight bulkheads are not manipulated by the algorithm.
- Defining input parameters manually and from scratch is a time-consuming process. The design database containing default settings is partial and underdeveloped, thus, many of OC and function shapes have to be separately defined every time GAOT is used.
- A rules-based network representation of major passageways is under development but has not yet been implemented
- The current version of GAOT tracks many parameters during the algorithmic process. Every separate PF value for every individual or combinations of cumulative PFs can be tracked and saved. However, they have to be analysed manually between two subsequent stages since no statistical analysis feature has been developed yet.
- Only 2D representations of relevant decks are available as an output from the tool at the moment, and the MDW does not yet have

interactivity to allow the designer to "drag and drop" entities into place and edit their characteristics; this version of MDW requires numerical inputs for all manual editing (changing positions, areas or shapes of OEs).
- Breaking down high-level blocks by arranging each one separately is supported by the current version of the tool but it is not straightforward since a new CD has to be defined manually; future versions will automate this feature in a way to automatically adjust to the new CD and semi-automatically set up multiple subcases reflecting the prearrangement of high-level blocks.

4. PRELIMINARY RESULTS

Since the development of GAOT is still ongoing, a complete set of test cases has not yet been carried out. Two sets of tests have been conducted so far both of which were targeting a specific subset of OC. The first set of experiments focused on validating overlap violation algorithm and penalty function approach, while the second one focused on testing global preferences and relational OC.

4.1 SINGLE DECK SHIP TEST CASE

The first batch of experiments was carried out on a single deck ship. The objective was to maximise deck utility by arranging 10 spaces and a main passageway, and to avoid overlapping. The intention of these tests was to validate implemented overlap management algorithm and basic geometrical capabilities of the tool so as to make sure the tool can cope with demands for flexible shapes, areas and positions of OEs. The second objective was to prove the penalty function concept. For this purpose, four fitness functions – one for each test case – have been developed all of which contained the same subset of OC encoded by following a different approach. OC to be tested were VSP reflecting the deck utility, OC from overlap management, aspect ratio, area and shape PFs.

Compliance with deck shape, compartment shape and compartment aspect ratio OC have been defined as hard nonlinear constraints for the first two test cases; overlapping OC have been included within the fitness function in a form of SPFs and CPFs for test cases 1 and 2 respectively. Existence of hard constraints implies that the algorithm doesn't consider individuals that violate any of the constraints. On the other hand, by modelling OC as PFs, all individuals are considered, and the algorithm tries to exclude bad individuals from the subsequent generations due to their poor finesses.

Hard constraints have been excluded from the remaining two cases; all mentioned OC have been modelled as SPFs for the test case 3, whereas only CPFs have been used in test case 4. All SPFs have been modelled as step functions with the same penalty value of 10^{50}. This value was returned every time the constraint was violated regardless of the level of violation. In contrast, CPFs return penalty value based on the level of a constraint violation, WFs and the shapes of the penalty functions. Linear shapes have been selected for all CPFs.

All four test cases had VSP defined as a difference between the overall deck area and the deck area occupied by spaces. The area of each of ten spaces to optimise was restricted to be within the range of 1/5 and 1/15 of the overall deck area, while the area of the main passageway was fixed. Overall fitness value was set up according to equation 1 and divided by the overall deck area. All OC have been normalised to the penalty area, and the target fitness value was set to be 1, which corresponds to satisfying all constraints and reaching minimum objective value.

All four cases were left to run in parallel. Simulations were stopped after test case 3 and 4 reached 400 generations. All simulations were run with the population size of 20000 and with the crossover fraction (ratio between crossover and mutation children) of 0.7. Cases 1 and 2 were converging very slowly due to hard constraint definition. Both of them were stopped after case 4 reached 400 generations; during this time case 1 reached 9 generations and case 2 only 2 generations. Resulting GAs are displayed in figure 6, while figure 7 shows graphs of the overall fitness value and VSP divided with the overall deck area over generations.

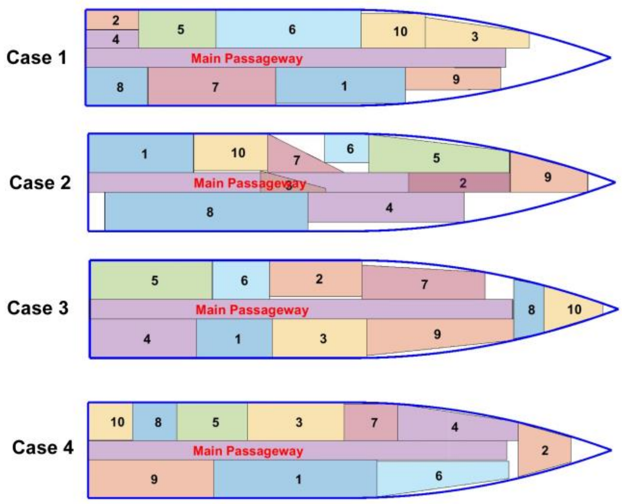

Figure 6: Results from the single deck ship test cases – 2D spatial configurations

It is clear that results from both cases 3 and four 4 have almost fully converged satisfying all geometrical, compartments shape, areas and overlapping OC. Around 98 percent of the space was utilised for both cases 3 and 4. This can be also seen from the deck utility tracking parameter from figure 7, however, WF of 6 has to be considered as well. Even though both PF approaches yielded satisfactory results, CPF approach took 4 times more computational time to reach 400 generations compared to just a few hours of iterating necessary for SPF approach. The reason for this lies in the fact that the algorithm has to calculate the overlapping area for every combination of OEs and for every individual during one generation, which takes much more time than just checking if the constraint is violated and assigning 0 or 10^{50} correspondingly.

On the other hand, Case 1 seemed to satisfy overlapping OC, and partially satisfied compartment shape criteria while the deck utility still reached reasonably high 90 percent. By contrast, during the same time, test case 2 reached only 2 generations (it was stopped in the third generation) which was not enough to suppress overlapping; compartment shape and areas were also not fully satisfied. Overall, both test cases 1 and 2 required significantly more time to start converging and yielded worse GAs.

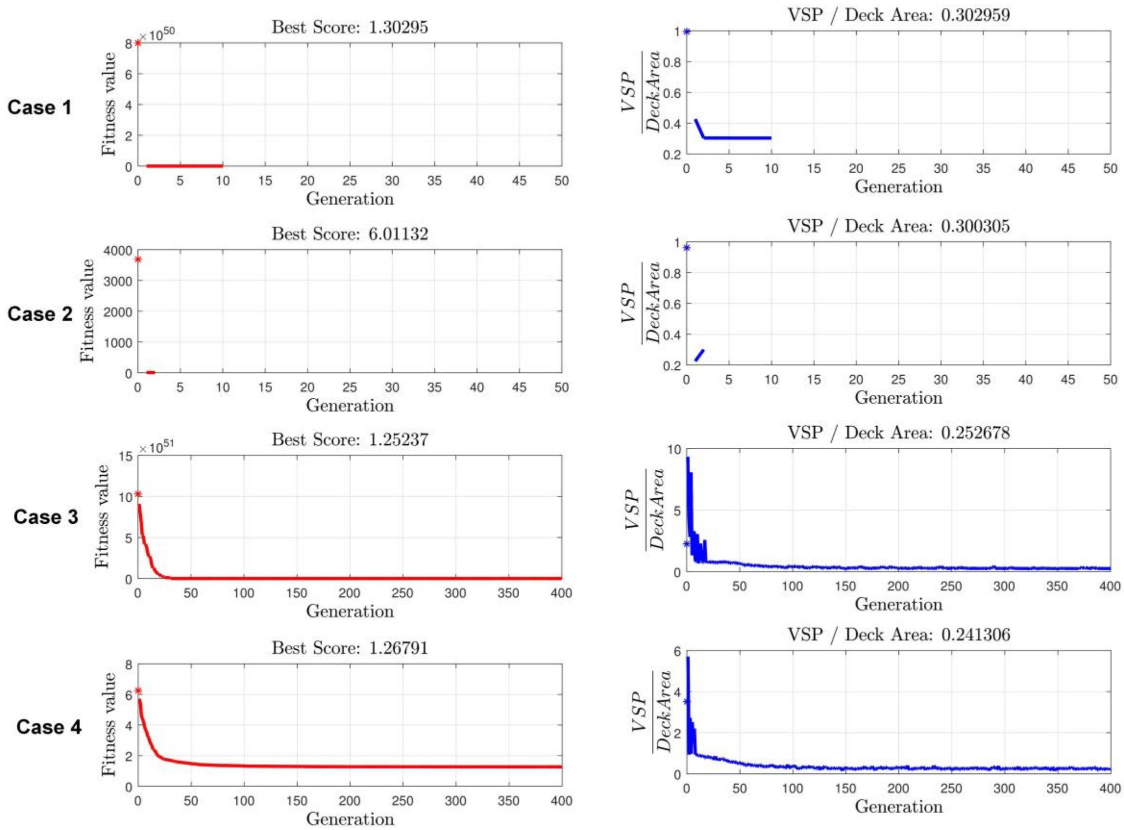

Figure 7: Some tracking parameters from single deck ship test cases

The penalty function approach proved to be superior over the use of hard constraints from both accuracy and computational time perspectives. Moreover, SPF approach happened to be more efficient for this level of complexity converging towards the solution on average four times faster. However, CPFs were tested only with linear function shapes and without the results of sensitivity analyses that would indicate the best combination of WFs. Hence, at this point, it is expected that the CPF approach will cope better with the increasing complexity. This hypothesis will be tested in the future.

GAOT utilises both SPF and CPF settings as well as the constraint function definition. Thus, it gives the user the freedom to choose the best combination of these three settings so as to adjust and optimise GAOT parameters to efficiently solving problems of different complexities.

4.2 FIVE DECK OPV TEST CASE

The second set of experiments focused on testing global preferences, relational OC and overlap management altogether, while VSP has been omitted. For this purpose, an illustrative 5-deck OPV with 50 high-level blocks on board has been selected. In order to test the ability of the algorithm to deal with global preferences, OPV was first manually designed (deck plans are shown on the right side of figure 8) to serve as a baseline. This GA contains quite a few voids; the reason for this is twofold. Only the first instance of highly important spaces have been imported, voids are in this case necessary in order to give provision for other spaces that haven't been included. Secondly, the purpose of this test was to validate global preferences and relational OC, and therefore, the case was set up to reflect this by excluding void space minimisation. Deck shapes have also been simplistically defined, assuming a wall-sided hull.

Thus, the overall aim was for the algorithm to replicate the GA from the baseline. The case was set up in a way that a target area and a range of tolerated aspect ratios have been defined for every compartment. Preferable global positions have been specified for the majority of compartments following the baseline, while the positions of the remaining OEs were restricted based on relational OC. Watertight bulkheads have been turned off and omitted from this experiment. The CPF approach has been utilised, where shapes of all penalty functions were linear. Maximum deviation from the target value was specified for each PF; this means that PF returns penalty value higher than zero only for deviations higher than the maximum tolerated. The algorithm was also set up to prefer rectangular shapes if possible.

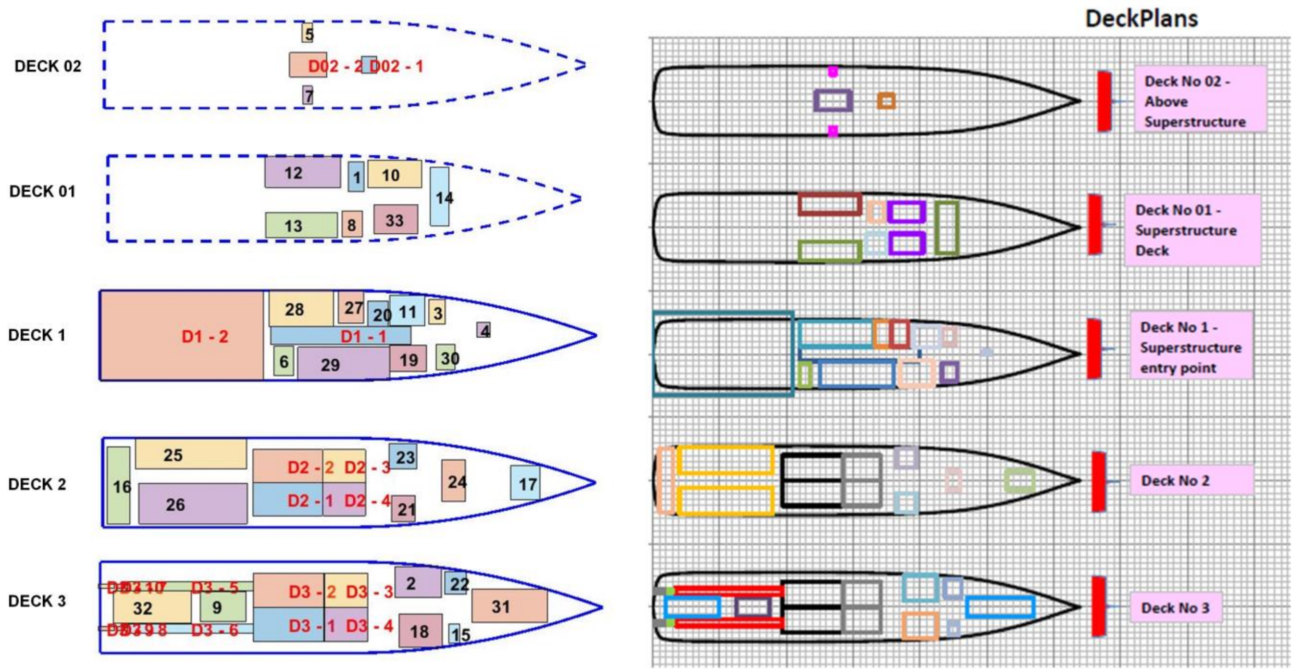

Figure 8: Five deck OPV test case

The case has been set up to have only one unambiguously defined minimum which was set up to be 0 this time. The simulation was left to run overnight and the result is shown in figure 8 left. The simulation converged towards the minimum with the less than one percent deviation from the baseline. The slight difference compared to a baseline can still be spotted; this is due to the specified maximum tolerated deviation and the range of acceptable (non-penalised) aspect ratios. However, the result of this test showed that the algorithm is able to cope with the geometrical complexity and that the relational and global preferences OC are encoded properly. The next tests will focus on case studies with all OC turned on, running the GAOT in a multi-objective manner, giving, therefore, more freedom to GAOT to fully explore the DSS, encourage diversity and yield multiple Pareto-optimal solutions in every hybrid stage.

5. FUTURE WORK

Goals for the future developments of the tool reflect the identified limitations of the current version. Development of GAOT will continue by gradually addressing limitations spelt out in subsection 3.3. Future work will focus on the following, where the sequence of bullet points also implies the priority:

- Translate the code to Python for compatibility in a research team
- Identify the best performing genetic algorithm that will be then implemented into GAOT
- Test the tool in a multi-objective manner
- Include the number of decks, hull dimensions and watertight bulkheads to be manipulated by the algorithm
- Incorporate the statistical analysis visualisation feature for providing additional feedback and evaluation of returned design options
- Include a separate stage to incorporate systems routings as well as main passageways; should come as the second hybrid stage.
- Upgrade the MDW with the drag and drop feature, and also by developing automatic layout refinement feature which will automatically improve the geometry of returned GAs.

- Upgrade the deck visualisation output to be able to toggle between the 2D representation of decks and 3D mode.
- Incorporate the generic and appropriate database containing OC settings, hull forms, list/s of generic spaces which could be then chosen and easily imported by the user, and/or modified if necessary. The idea is to gather certain collections of design rules which would be appropriately grouped (for instance, naval or commercial, type of the ship, role etc.) and ready for the designer to use/import.

6. CONCLUSIONS

This paper describes progress in a collaborative project that seeks to develop highly interactive layout exploration and optimisation method for semi-automatic generation of general arrangements during early stages of ship design. A novel hybrid approach has been discussed where the design process remains under the control of naval architects with collaboration with the other disciplines (such as naval engineers, combat system engineers, human factors, system engineering, etc.). The method combines a manual design environment with a General Arrangement Optimisation Tool (GAOT). It is approached in stages where each stage is characterised by two steps. Firstly, the GAOT is utilised to automatically generate a diverse set of alternative GAs followed by manual step where the designer continues the process manually by considering returned alternatives separately and returning modified set of designs back to GAOT as the initial solution for the next hybrid stage.

The staged approach allows optimisation/design criteria to be gradually introduced during DSS exploration by incorporating newly gained insights (i.e. adding, modifying or removing certain criteria). It differs from traditional approaches where optimisation criteria are defined *a-priori* – substantially limiting the scope of DSS exploration – or *a-posteriori* by filtering and evaluating a large set of designs manually, which becomes excessively labour consuming. The method reflects the iterative nature of all ship design processes, allows for important dialogues among all interested parties to take place; it enables survivability zoning and major routing to be taken into account by the GAOT. Moreover, this approach enables for the complexity to be built gradually by introducing and arranging bigger blocks first (conglomerations of similar spaces, spaces of the same function, spaces that should go together) and breaking them down afterwards by arranging each bigger block separately. The other way to gradually build the complexity is to start by introducing high priority spaces, arranging them and finding major routes first, and introducing lower priority spaces in later stages.

The current implementation of GAOT has been presented. The tool utilises overlap management where the designer can choose between modelling constraints by means of constraint functions (hard constraints) or penalty functions (soft constraints). Penalty function approach implies defining Penalty Factors (PF) which are then minimised along with other Optimisation Criteria (OC) and objectives as a part of fitness function; this approach allows for determining the degree of satisfaction of the given constraints. GAOT supports both step and continuous function based PFs. The latter PFs are characterised by the shape of the penalty function and weighting factor that gives importance and priority to the given OC. On a high level, OC can be categorised as global preferences, relational OC and basic naval architecture. Global preferences encompass OC such as global preferable positions, target areas or acceptable shapes and are predominantly linked to the OE itself; relational OC express interdependencies between OE themselves, OE and predefined entities and OE and fixed objects; examples are separation, adjacency, proximity, etc. Void space PF describing the deck utility has been included within the set of OC as a part of basic naval architecture OC along with LCG, VCG, etc.

GAOT is still under development and only the first set of tests targeting a specific set of OC have been conducted. Results showed that the PF approach is faster and more accurate compared to hard constraints approach. Moreover, it has been shown that test case utilising step function PFs converged towards the solution significantly faster than the other three simulations indicating to be the most efficient set-up for the given complexity. It is expected that the continuous function PF approach will cope better with the problems of higher complexity. However, the presence of all three approaches gives the user a choice to choose the appropriate combination of constraint definitions within the overlapping management, which makes the tool flexible enough to efficiently adjust to the level of complexity of a given problem and to save computational time.

Overall, it can be said that GAOT successfully coped with the complexity of conducted experiments. Therefore, it can be inferred that the genetic algorithm-based approach is a viable path forward. Future work will primarily focus on translating the code into Python, identifying the best performing genetic algorithm from the variety of commercially available ones, speeding up the overall code and developing statistical analysis and visualisation output feature. Also, GAOT will be thoroughly tested in a multi-objective manner and the verification and comparison metric will be developed.

7. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding and support by BMT.

8. REFERENCES

1. ANDREWS D. J., 'Marine Design - Requirement Elucidation rather than Requirement Engineering', in *Proceedings of the International Marine Design Conference (IMDC) May 2003*, Athens, Greece, 2003, pp. 2–18.

2. ANDREWS D. J., 'Preliminary Warship Design', *Trans. R. Inst. Nav. Archit.*, vol. 136, pp. 37–55, 1994.

3. ANDREWS D. J., KEANE R. G., LAMB T., SEN P., and VASSALOS D., 'State of the Art Report: Design Methodology', in *Proceedings of the International Marine Design Conference (IMDC) 2006*, Ann Arbor, Michigen, USA, 2006.

4. ANDREWS D. J. and DICKS C. A., 'The Building Block Design Methodology Applied to Advanced Naval Ship Design', in *Proceedings of the International Marine Design Conference (IMDC) 1997*, Newcastle, UK, 1997.

5. ANDREWS D. J., 'Creative Ship Design', *Trans. R. Inst. Nav. Archit.*, vol. 123, 1981.

6. ANDREWS D. J., CASAROSA L., PAWLING R. J., GALEA E., DEERE S., and LAWRENCE P., 'Integrating Personnel Movement Simulation into Preliminary Ship Design', *Int. J. Marit. Eng.*, vol. 150, no. A1, 2008.

7. VASUDEVAN S. and RUSLING S., 'A Ship Design Tool Using Genetic Algorithms', in *Proceedings of the International Conference on Computer Applications in Shipbuilding (ICCAS) 2007*, Portsmouth, UK, 2007.

8. ANDREWS D. J., 'An Integrated Approach to Ship Synthesis', *Trans. R. Inst. Nav. Archit.*, vol. 128, pp. 73–91, 1986.

9. ANDREWS D. J., DUCHATEAU E., GILLESPE J., HOPMAN H., PAWLING R. J., and SINGER D., 'Design for Layout', in *Proceedings of the International Marine Design Conference (IMDC) May 2012*, Glasgow, Scotland, UK, 2012, pp. 111–137.

10. LEVANDER K., 'Innovative Ship design - Can Innovative Ships be designed in a Methodological Way?', in *Proceedings of the International Marine Design Conference (IMDC) May 2003*, Athens, Greece, 2003.

11. ANDREWS D. J., 'The Sophistication of Early Stage Design for Complex Vessels', *RINA Int. J. Marit. Eng.*, vol. 160, no. 1–53, 2018.

12. BROWN D. K., 'The Architecture of Frigates', in *Proceedings of RINA Warship 1987: Anti-Submarine Warfare*, London, UK, 1987.

13. DEPETRO A., 'Rapid Ship Design Evolution Using Computer Algorithms - A Scientific Approach to SEA 1180', *BMT Design & Technology*, Victoria, Australia, 2015. [Online]. Available: https://pdfs.semanticscholar.org/c4d4/454d4e7e4aecbfadf8a71c8b23c890535ad2.pdf?_ga=2.99146718.1517229703.1563206586-1020734705.1560689659 [Accessed: 02-Oct-2018]

14. DEVINE M., 'ASSET - A computer aided engineering tool for the early stage design of advanced marine vehicles', in *8th Advanced Marine Systems Conference*, 1986.

15. NETI S. N., 'Ship Design Optimization Using ASSET', *Master's Thesis, Virginia Polytechnic Institute and State University*, Blacksburg, Virginia, USA, 2005.

16. LEVIS D. R. and FORSTELL B. G., 'The Cost-Benefit of Emerging Technologies using Physics-based Ship-Design Synthesis', pp. 327–338, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.201.962&rep=rep1&type=pdf [Accessed: 08-Jul-2018].

17. LEVIS D. R. and GOUBAULT P., 'Physics-based Ship Design Synthesis', in *Proceedings of the International Marine Design Conference (IMDC) May 2003*, Athens, Greece, 2003, pp. 21–35.

18. NOVAK V., PERFILIEVA I., and MOCKOR J., 'Mathematical principles of fuzzy logic'. *Boston: Kluwer Academic*, 1999.

19. NEHRLING B., 'Fuzzy set theory and general arrangement design', in *IFIP/IFAC Fifth International Conference on Computer Applications in the Automation of Shipyard Operations and Ship Design 1985*, Trieste, Italy, 1985.

20. CORT A. and HILLS W., 'Space Layout Design using Computer Assisted Methods', *Nav. Eng. J.*, vol. 99, no. 3, pp. 249–260, May 1987.

21. SLAPNICAR V. and GRUBISISC I., 'Multi-criteria optimization model of deck layout design', in *Proceedings of the International Marine Design Conference (IMDC) May 2003*, Athens, Greece, 2003.

22. OLCER A. I., TUZCU C., and TURAN O., 'An integrated multi-objective optimisation and fuzzy multi-attributive group decision-making technique for subdivision arrangement of Ro–Ro vessels', *Appl. Soft Comput.*, vol. 6, no. 3, pp. 221–243, Mar. 2006.

23. DANIELS A. S., TAHMASBI F., and SINGER D. J., 'Intelligent Ship Arrangement Passage Variable Lattice Network Studies and Results', *Nav. Eng. J.*, vol. 122, no. 2, pp. 107–119, Jun. 2010.

24. DANIELS A. S. and PARSONS M. G., 'Development of a Hybrid Agent-Genetic Algorithm Approach to General Arrangements', in *Proceedings of the International Conference on Computer and IT Applications in the Marine Industries (COMPIT) April 2007*, Cortona, Italy, 2007, pp. 197–211.

25. DANIELS A. S. and PARSONS M. G., 'An Agent Based Approach to Space Allocation in General Arrangements', in *Proceedings of the International Marine Design Conference (IMDC) May 2006*, Ann Arbor, Michigen, USA, 2006, pp. 673–695.

26. NICK E. K., PARSONS M. G., and NEHRLING B., 'Fuzzy Optimal Allocation of Spaces to Zone-Decks in General Arrangements', in *Proceedings of 9th International Marine Design Conference (IMDC) May 2006*, Ann Arbor, Michigen, USA, 2006.

27. PARSONS M. G., CHUNG H., NICK E., DANIELS A., LIU S., and PATEL J., 'Intelligent Ship Arrangements: A New Approach to General Arrangement', *Nav. Eng. J.*, vol. 120, no. 3, pp. 51–65, Dec. 2008.

28. NICK E. K., 'Fuzzy Optimal Allocation and Arrangement of Spaces in Naval Surface Ship Design', *PhD thesis, University of Michigan*, 2008.

29. LEAPS, 'Leading Edge Architecture for Prototyping Systems', *Naval Surface Warfare Center, Carderock Division*, Version 3.4, 2006.

30. BEYER C., CHE C., and DEVINE M., 'User manual for ASSET/MONOSC: Advanced Surface Ship Evaluation Tool / Monohull Surface Combatant Program', 1990.

31. VAN OERS B. J., STAPERSMA D., HOPMAN H., and BERTRAM V., 'A 3D Packing Approach for the Early Stage Configuration Design of Ships', in *Proceedings of the 10th International Naval Engineering Conference (INEC 2010)*, Portsmouth UK, 2010.

32. VAN OERS B. J., 'A Packing Approach for the Early Stage Design of Service Vessels', *PhD thesis, Delft University of Technology*, 2011.

33. DEB K., PRATAP A., AGARWAL S., and MEYARIVAN T., 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

34. DENUCCI T. and HOPMAN H., 'Capturing configuration rationale in complex ship design: Methodology and test case results', in *Proceedings of the International Marine Design Conference (IMDC) May 2012*, Glasgow, Scotland, UK, 2012, pp. 233–250.

35. DENUCCI T., HOPMAN H. and VAN OERS B. J., 'Capturing trade-off rationale during design selection', in *8th International Conference on Computer and IT Applic ations in the Maritime Industries (COMPIT) May 2009*, Germany, 2009, pp. 214–227.

36. DUCHATEAU E., VAN OERS B. J., and HOPMAN H., 'Interactive steering of an optimisation-based ship synthesis model for concept exploration', in *Proceedings of the International Marine Design Conference (IMDC) May 2015*, Tokyo, Japan, 2015, pp. 29–42.

37. ROTH M., DROSTE K., and KANA A. A., 'Analysis of General Arrangements Created by the TU Delft Packing Approach', in *16th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT) May 2017*, Cardiff, UK, 2017, pp. 201–2011.

38. DROSTE K., KANA A. A., and HOPMAN J. J., 'Process-based analysis of arrangement aspects for configurational-driven ships', in *Proceedings of the International Marine Design Conference (IMDC) May 2018*, Finland, Helsinki, 2018, pp. 327–337.

39. MUNOZ J. A. and FORREST C. J. M., 'Advantages of Software Integration from Initial Design Through to Production Design', in

Proceedings of ICCAS 2002, Malmo, Sweden, 2002.

40. ANDREWS D. J., 'Synthesis in Ship Design', *PhD thesis, University of London*, 1984.

41. ANDREWS D. J. and PAWLING R. J., 'A case study in preliminary ship design', *RINA Int. J. Marit. Eng.*, vol. 150, no. A3, 2008.

42. PAWLING R. J., 'The application of Design Building Block Approach to Innovative Ship Design', *PhD thesis, University College London*, 2007.

43. HOLLAND J. H., 'Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence'. *Ann Arbor: University of Michigan Press*, 1975.

44. KELNER V., CAPITANESCU F., LEONARD O., and WEHENKEL L., 'A hybrid optimization technique coupling an evolutionary and a local search algorithm', *J. Comput. Appl. Math.*, vol. 215, no. 2, pp. 448–456, Jun. 2008.

45. COLEY D. A., 'An introduction to genetic algorithms for scientists and engineers'. Singapore ; *River Edge, NJ: World Scientific*, 2010.

46. HAUPT R. L. and HAUPT S. E., 'Practical genetic algorithms', *2nd ed. Hoboken, N.J: John Wiley*, 2004.

47. CHIPPERFIELD A., FLEMING F., POHLHEIM H., and FONSECA C., 'Genetic Algorithm Toolbox: For Use with MATLAB, User's Guide'. *Department of Automatic Control and Systems Engineering, University of Sheffield*, 1994.

48. MathWorks, 'Genetic Algorithm Support', *Global Optimization Toolbox Documentation*, 2018. [Online]. Available: https://uk.mathworks.com/help/gads/genetic-algorithm.html [Accessed: 20-Mar-2018].

9. AUTHORS BIOGRAPHY

Bojan Igrec is a PhD student at UCL. He is a graduate in Mechanical Engineering from the University of Zagreb. After graduation he worked at Faculty of Electrical Engineering and Computing of University of Zagreb as a research assistant where he was engaged in the development of fibre-optic sensor system for vibration measurement applied to high power electric machinery. He joined UCL in the summer of 2017, and his main area of interest is in the development of methods and tools for computer aided layout generation within ship design, particularly during the preliminary stages. He can be reached at bojan.igrec.16@ucl.ac.uk.

Rachel Pawling is a Lecturer in Ship Design at UCL, teaching the subject of ship design to undergraduate and postgraduate students. She obtained her PhD in computer aided ship design in 2007 and has subsequently continued her research in projects funded by the EC, UK and US governments.

Giles Thomas is BMT Chair of Maritime Engineering in University College London's Department of Mechanical Engineering. A naval architect, his research focusses on the performance of ships, boats and offshore structures. Specific interests include fluid-structure interaction, hydrodynamics, full scale measurements, model testing and design. He is a Fellow of the Royal Institution of Naval Architects (RINA) and a Chartered Engineer (CEng – Engineering Council of Great Britain).

Adam Sobey is a Lecturer in soft computing applied to marine applications. He previously performed a PhD in Concurrent Engineering in the context of the leisure boatbuilding industry, focusing on automated tools for design of composite structures, including optimisation using genetic algorithms. As part of his research he has developed the Multi-Level Selection Genetic Algorithm (MLSGA) methodology, which shows top performance on a range of problems, implementing inspiration from evolutionary theory.

Jake Rigby is the Research and Development Lead and Senior Naval Architect at BMT. He is responsible for internal research, Academic Engagement and Horizon Scanning.