Genetic Algorithm Selection for Ship Concept Design

Adam Sobey(✉), Przemyslaw Grudniewski, and Thomas Savasta

Fluid Structure Interactions Group, University of Southampton, Southampton SO16 7QF, UK
ajs502@soton.ac.uk

Abstract. Engineering designers are increasingly looking to automate the design process. By utilizing tools to support decision making engineers can explore a wider range of options and reduce the time to an initial design, which can then be iterated over a number of cycles. One area which is prominent in the literature is the design of boat layouts as this is a complex part of the design which interacts with a range of different subsystems. The layout is often optimized using Genetic Algorithms and the selection of these algorithms has been shown to be important. In addition, the representation for this problem is also vital, with a balance between the continuous nature of the design space and the total size of the space. This paper therefore explores one representation, the vector-based approach, to determine how reducing the number of constraints might improve the performance. The new representation is compared over 3 problems of increasing complexity using a number of current state-of-the-art Genetic Algorithms. The results show that the reduction in constraints increases the size of the space more than it reduces the complexity, showing poorer results than previously. cMLSGA is always the top performing Genetic Algorithm and it can be concluded that the diversity focused general solvers should be promoted on similar kind of problems.

Keywords: Layout design · Evolutionary computation · Automated design

1 Marine Layout Design

Ships are increasingly complex to design. The number of objectives that designers want to consider is growing, regulation is becoming more complex, customers are more demanding due to the current economic climate and accounting for the environment is an ever increasing concern. For a practical, systems-level, approach to design this means that the designers job is becoming increasingly difficult due to the number of variables that need to be considered. This is both the manner in which the variables from their own subsystem interacts with variables in other subsystems, for example stiffener height impacting on cabin volume in a yacht, but the increase in complexity in their own subsystem, for example many vessels are now looking at optimising a hull over an operating profile rather than a maximum speed. Many other industries with similar issues are now looking to software solutions to help simplify the design space. This can be in the use of data to support decision making, generation of databases and

T. Okada et al. (Eds.): PRADS 2019, LNCE 65, pp. 156–172, 2021.
https://doi.org/10.1007/978-981-15-4680-8_11

linked software to allow a freer interaction between engineers in different subsystems or automated design tools.

Automated design tools provide a way to search large design spaces, visualise results to reduce complexity and incorporate areas of expertise outside of the designer's domain knowledge. These tools are becoming increasingly prevalent, such as the use of feature selection tools to reduce the number of variables down to only the key set or optimisation tools to find the best set of designs in a given search space. Genetic Algorithms are a popular tool for the optimization process as they are flexible to a range of problems and powerful in finding solutions across large search spaces. However, a key problem in optimisation is the 'no free lunch' theory: in essence, that problems have different characteristics and that different algorithms are suited to solving these problems. If the problem is well understood then a specialist solver can be used which will have excellent performance, but if the problem is not known then a general solver must be used that inevitably has poorer performance. For many of the very difficult problems in engineering it is likely that the general solvers will be incapable of solving the problems proposed or take many more computationally expensive function calls to do so. It is therefore vital that problems are categorized and the corresponding specialist algorithms are used or developed.

This paper therefore investigates the performance of a number of 'state-of-the-art' Genetic Algorithms on the optimisation of a general arrangement, categorising the problem and highlighting the desired characteristics required from a Genetic Algorithm. A range of top performing algorithms will be used: U-NSGA-III, MOEA/D-MSF, HEIA, BCE and cMLSGA; to determine which algorithms perform most strongly. This will be performed on a simple yacht design problem, allowing a relatively complex general arrangement but reducing computational time. In addition the importance of constraints, population size and number of objectives will be investigated. Recommendations will be given on how algorithm design needs to be developed further to solve the General Arrangement problems and how extending the problem to larger and more complex problems can be achieved.

2 Review of Boat Layout Optimization

The automated optimization of a boat layout requires a method to represent the layout and then a method to optimize the layout suited to this representation. There are a number of approaches available in the literature with most of them utilizing Genetic Algorithms for the optimisation. There are four main methods in the literature for representing the layout of the vessel, each relying on different numbers of constraints, variables and objectives: intelligent ship arrangement, packing approach, grid based and vector representation.

The intelligent ship arrangement is originally developed in 2008 by Daniels and Parsons [1]. This method utilises the Agent-Genetic Algorithm to optimise the layout which is a hybrid between the agent-based approach and a Genetic Algorithm. They compare the method to an agent-based approach, which is uncommon in the literature, and a pure Genetic Algorithm [2] which is based on the original Genetic Algorithm. The population sizes used are small, 100, and the mutation rate is high, 30%. It is expected that the initial diversity of points is too low for the problem type and that a wider population would be more successful. In addition the algorithm developed appears to be based on Goldberg [3], developed in 1989, or before, meaning that the solvers used to resolve the optimisation are dated. The representation relies on a high number of constraints to reduce the design space, with a number of different representations compared, this ranges up to 521 constraints in the database. The representation relies on the use of a binary representation where the inside of the vessel is split into rectangles, with each block assigned an available compartment type. This is similar to scheduling type problems seen in the literature on which a number of benchmarking exercises of optimisation algorithms have been performed. The outputs from the optimisation are judged to be valid and to provide high quality designs, even on large problems.

van Oers et al. [4] use a packing approach which is optimised using NSGA-II implemented in Matlab 2007b, which is an early version of NSGA-II which has since be updated in 2011 [5]. The code uses a population size of 500 and a 500 generations. This is the most commonly used Genetic Algorithm and shows strong diversity of search providing excellent performance as a general solver. However, the algorithm is not benchmarked against other candidate solutions. The layout is represented using a block approach which has a relatively low number of variables for each compartment, making it scalable for larger applications. The aim is to reduce the quantity of overlap detection and removal as far as possible to make the algorithm more efficient. This means a solid definition is used where the boundaries of a room are used alongside information about whether point lie in or outside an object. The available space is then searched for after the design is generated. The implementation allows $6.35 * 10^4$ evaluations in 10 min on a standard desktop computer. The designs contain a large number of voids, but the designs are judged to be feasible for the application.

Nam, Kim and Le [6] use a grid based system to represent the layout which is extended in Nam and Lee [7], both approaches use a Genetic Algorithm to determine the layout of the vessel with a four deck optimization of the superyacht shown in Fig. 1. The Genetic Algorithm used is a Constraint-Based Genetic Algorithm (CBGA). In the development and documentation of the Genetic Algorithm no other algorithms are compared against and there is no benchmarking on other problems outside of boat layouts. This makes it difficult to determine how effective the algorithm is in comparison to the state-of-the-art. However, the approach is compared favourably to a branch and leaf hybridised with simulated annealing. The problem is treated to be similar to scheduling problems seen in the optimization literature, where simulated annealing has been shown to perform poorly. The algorithm seeks to determine a layout with cabins while also looking at the position of the stairs, and therefore goes beyond just the optimization of the layout. The results of the algorithm are simple layouts, with no unused space, that provide short distances for escape via the staircases.

Finally, Sobey et al. [8] use a vector based approach to the optimisation based on Riley [9]. This approach is in 3D and utilises a large number of variables to represent each cabin, making it less scalable than the grid based method. The problem uses 49 variables to define a problem for a 24 m yacht, with 2 objectives of space utilisation and 3 constraints. They compare 10 different state-of-the-art Genetic Algorithms where the results show that cMLSGA [10] performs the best on these problems. It raises the

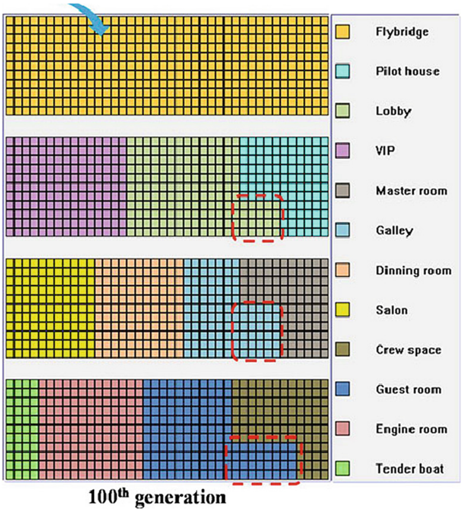

Fig. 1. Multi deck optimisation considering area requirements [7]

importance of benchmarking recent solvers and to the importance of modern Genetic Algorithms. This is because there is a separation between the performances of different algorithms, with the modern Genetic Algorithms substantially outperforming the older methods.

The four different main methods are summarized in Table 1 showing a range of representations that are available for the layout optimization. The literature shows that the methods rely on a high number of constraints, which results in design spaces that are traditionally harder for Genetic Algorithms to solve. In addition some of the methods show an escalation of variables with the number of compartments, also increasing the difficulty. More recent research shows the importance of algorithm selection and utilization of the current state-of-the-art from evolutionary computation.

Table 1. Marine Layout Optimisation

Heading level	Intelligent ship arrangement	Packing approach	Mesh representation	Vector representation
Genetic algorithm	Agent-GA hybrid	NSGA-II	CBDA	10 benchmarked with cMLSGA best performing
Variables	100	60	30	49
Objectives	3	3	2	2
Constraints	521	3	6	3
Compartments	205	30	40	7

3 Boat Layout Definition

The manner in which the boat layout is represented is important for the optimisation and so the representation presented in Sobey et al. [8] is updated to make it more efficient. Genetic Algorithms are generally designed with a poor ability to cope with discontinuities or constraints and therefore the new representation is focused on providing a continuous, unconstrained, layout representation. This representation is then compared across three different yachts to show how the increase in complexity can be handled by the Genetic Algorithm. In this case a motor yacht is optimised to maximise the space utilisation of the cabins in the hull. The motor yacht particulars for the 3 different case studies are described in Table 2.

Table 2. Motor yacht particulars.

	24 m	50 m	140 m
Length overall (m)	24.8	50	140
Max beam (m)	6.9	11.12	23.47
Draught (m)	1.955	3.50	5.0
Displacement (Tonnes)	152.8	418.6	2658.3
LCB (from transom)	8.673	17.737	48.975
Decks	1	1	2
Compartments	7	10	57
Variables	49	70	399
Objectives	4	4	8

3.1 Cabin Geometry

The new representation is reduced to a 2D definition, as previous studies have shown that there are limited benefits to a 3D representation if structures, piping and electrical subsystems are not considered. In this case each cabin is defined using 7 variables which are shown in Fig. 2:

1. x position from the forward perpendicular (FP);
2. y position from the centreline (CL);
3. length of the compartment (L);
4. distance from the centreline to the forward most point to starboard (W11);
5. distance from the centreline to the forward most point to port (W11);
6. distance from the centreline to the aft most point to starboard (W21);
7. distance from the centreline to the aft most point to port (W22).

In addition, the length of the cabin can't be longer than the distance between two bulkheads. These variables will be optimized to minimize 4 objectives for each deck, with 4 objectives overall for the small vessels, 24 m and 50 m, and 8 objectives for the larger 140 m vessel. The 4 objectives seek to reduce the unused area for the whole deck, minimize the distance between certain cabins, minimize the distance between the

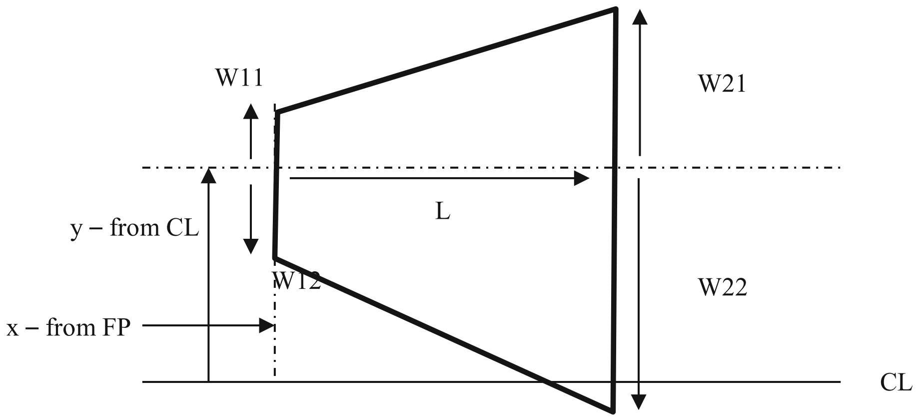

Fig. 2. Cabin geometry definition

Centre of Buoyancy and Centre of Gravity and ensure that the area of each cabin is similar to those seen for this type of vessel in the past.

The space fitness objective minimises the unused area of the deck compared to the whole deck area and is defined in Eq. (1):

$$Fitness_{space} = \frac{Deck\ area - \sum Cabins\ area}{Deck\ area} \tag{1}$$

To optimise the flow of people within the yacht the distance between cabins is minimized where 2 cabins, i and j, are defined as next to each other if one of the conditions in Eqs. (2)–(5) are satisfied:

$$x_i + l_i = x_j; \tag{2}$$

$$x_i + l_j = x_j; \tag{3}$$

$$y_i + max(w_{i11,}w_{i21)} = y_j - max(w_{i12,}w_{i22}); \tag{4}$$

$$y_j + max(w_{j11}, w_{j21}) = y_i - max(w_{i21,}w_{i22}). \tag{5}$$

If none of these conditions is satisfied, the condition showing the minimum value is returned.

To determine the trim of the vessel the centre of gravity and centre of buoyancy need to be determined. The longitudinal centre of area is found for each cabin using standard weights per unit area from data available in the literature. This is then multiplied by the actual area of the cabin to find its weights. The longitudinal centre of area is assumed to be positioned in the middle of the cabin length and the vertical centre of area is assumed to be the average of the 2 faces, shown in Eq. (6):

$$VCA = y + \frac{w_{11} + w_{12} + w_{21} + w_{22}}{4}. \tag{6}$$

The centre of mass of the cabins is assumed to be at the centre of area. The moments are taken in the transverse and longitudinal direction for each cabin and added together and divided by the total weight of cabins to find the position of the centre of gravity (CG) of the deck. The centre of buoyancy (CB) of the yacht is calculated using the sectional areas found from the yacht lines plan. The aim of this objective is to minimise the distance between CB and CG, shown in Eq. (7):

$$Fitness_{trim} = \frac{CB - CG}{CB}. \tag{7}$$

Cabin are be required to be of a certain size depending on their type. Standards areas are determine from data available in the literature and all of the cabins are required to respect these allocated areas to within a margin of 20%. The aim of this objective is to minimise the difference between the actual area and the standards area of each cabin, shown in Eq. (8):

$$Fitness_{size} = \frac{1}{nb\ of\ cabins} \cdot \sum \frac{Standard\,area - actual\,area}{Standard\,area}. \tag{8}$$

4 Genetic Algorithm Selection

The existing studies shows a limited range of Genetic Algorithms from the Evolutionary Computation literature are used within layout optimisation, all of these are applied without benchmarking to determine their performance or the characteristics required from the optimisation algorithm. However, recent research by Sobey et al. [8] shows that the layout optimisation of a boat hull is a complex optimisation problem for the Genetic Algorithms to solve, as some of the current state-of-the-art algorithms struggle to solve even simple representations of the layout. Therefore, it indicates that, the selection of the correct Genetic Algorithm is vital to ensure that the best Pareto Optimal Front is found.

A brief review of the current state-of-the-art in evolutionary computation is performed by splitting the algorithms into 4 broad categories: niching, decomposition, co-evolutionary and multi-level selection algorithms. Niching is exemplified by the crowding mechanism based niching technique found in the most popular Genetic Algorithm NSGA-II [11] which uses non-domination to sort the fittest solutions. It is a robust general solver with few hyper-parameters and retains a high diversity, making it the most popular algorithm in real applications. This methodology has been upgraded to a variety of versions, such as U-NSGA-III [12] that unifies NSGA-II and NSGA-III to be suitable for mono-objective, multi-objective and many-objective problems, the problem type on which it is currently the top performing algorithm.

Decomposition methods are a relatively new family where the population is divided into sub-groups that search different sub-regions of the search space. These algorithms implement a number of additional setting parameters, such as the weight vectors in MOEA/D, which greatly influence the solutions and must be optimised. MOEA/D [13] is the most popular decomposition method with a number of different variants specialised for different problem types. These methods require a priori knowledge of the objective space or they can result in extremely poor performance. However, it is hard to obtain the required knowledge of the objective space for practical optimisation problems before solving them. When solving discontinuous problems or constrained problems these algorithms struggle with the large gaps where there are no feasible solutions as the weight vectors point straight through the gaps and the individuals struggle to go around these spaces, resulting in a waste of computational power. However, these algorithms exhibit excellent convergence characteristics, dominating the benchmarking for dynamic and unconstrained problems.

The term coevolution is first introduced to describe the coexistence of plants and butterflies [14]. In the co-evolutionary approaches, multiple populations of species of individuals coexist and evolve in parallel, usually utilising distinct reproduction mechanisms. There are two currently top performing methods that utilise this approach, Bi-Criterion Evolution algorithm (BCE) and Hybrid Evolutionary Immune Algorithm (HEIA). In Bi-Criterion Evolution algorithm (BCE) [15] sub-populations operate on the same search spaces and individuals for each group are selected at each generation based on two distinct fitness indicators: the Pareto-based criterion (PC) and the Non-Pareto-based (NPC). In the Pareto-based criterion, standard Pareto dominance is utilised which rewards convergence whereas in the Non-Pareto based selection an additional indicator is introduced, based on Hypervolumes (HV) which rewards diversity of solutions. This leads to an overall improvement in diversity for the entire population, especially on many-objective cases and problems with irregular search spaces and variable linkages. However, it is still convergence dominated. A similar approach has been utilised in Hybrid Evolutionary Immune Algorithm (HEIA) [16], but in this case two distinct evolutionary computation methods are used, Immune Algorithm and Genetic Algorithm, instead of separate quality indicators. This method shows excellent performance on quite a wide range of problems, but is more convergence orientated and the performance of this method has not been evaluated on highly discontinuous problems and constrained problems where the performance is expected to be low, as it only utilises crowding distance for diversity.

Multi-Level Selection Genetic Algorithm was developed to take advantage of the recent evolutionary theories of Wilson and Sober [17]. Wilson and Sober propose that evolutionary fitness is not just dependent on the fitness of the individual but can also be dependent on the collective of individuals that it is associated with, an example might be that the survival of a wolf is not just dependent on its own fitness but the fitness of its pack. Multi-level selection Genetic Algorithm (MLSGA) was first introduced by Sobey and Grudniewski [18] and [19], in this algorithm a collective level reproduction mechanism is introduced, in addition to the individual level used in standard Genetic Algorithm, and the fitness function is split between these levels, with this algorithm being unique in this respect. The algorithm works by randomly generating an initial population which is classified into collectives according to the design variables. On the individual level, each individual is evaluated through the individual objective function and genetic operators are utilised to perform individual reproduction, and these can be selected from any of the current state-of-the-art methods. Simultaneously, each collective is evaluated using the collective objective function. There is a competition among the collectives and the worst collective(s) is eliminated. This collective is replaced by generating a copy of the best individuals from each of the remaining collectives. The process is stopped when the termination condition is satisfied. There are two main fitness evaluation methods, MLS1 and MLS2 [18]. MLS1 uses the aggregate of the individuals in the population to calculate the fitness of a collective. MLS2 calculates different objectives using a fitness defined for the collective, with MLS2R defined as being reverse. Therefore, MLS1 focuses on solutions at the middle of the Pareto optimal front and MLS2 and MLS2R enhance the search ability at the two sides of the real Pareto optimal front separately. Based on the two main methods, MLS-U, combining MLS1, MLS2 and MLS2R, can be utilised in MLSGA to maintain the diversity of the search shown in Fig. 4. This method has recently been combined with the co-evolutionary approach, co-evolutionary Multi-Level Selection Genetic Algorithm (cMLSGA), to increase its generality. So far MLSGA, and its variants, have shown to have top performance across a range of state-of-the-art multi-objective problems. The algorithm seems to thrive in environments with discontinuous fronts and constrained problems where diversity of the mechanisms is important. Multi-level selection theory is unique in that it is the only diversity first search Genetic Algorithm.

In this study a total of 5 genetic algorithms are used to solve the presented cases: MOEA/D-MSF [20] as an improved variant of MOEA/D for imbalanced and unconstrained problems with a similar performance to MOEA/D-PSF on this problem type; HEIA [16] as an algorithm that shows high proficiency across a diverse set of problems and is a general solver with a bias towards convergence; BCE [15] which is another more recent algorithm designed as a general solver but with a stronger bias towards convergence than HEIA; U-NSGA-III [12] as the many-objective universal variant of NSGA-II [11], which is the current state-of-the-art seen in the marine literature; cMLSGA as the best general solver [10] which also provides a diversity first search and an algorithm developed to be similar to the original Genetic Algorithm, representing a solver which is still common in the marine literature.

The tests are performed over 15 separate runs, and the termination criterion is set at 300,000 function evaluations for each run on the 24 m and 50 m case studies and 50,000 for the 140 m simulations as some solvers perform the analysis slowly with the high number of objectives. The results are compared using the Hyper Volume (HV) and Inverted Generational Distance (IGD) indicators, as between them they provide comprehensive information on the convergence, accuracy, and diversity of the obtained solutions. HV is the measure of volume of the objective space between a predefined reference point and the obtained solutions which has a stronger focus on the diversity and edge points and can be calculated according to [21], where higher values indicates the better results. IGD is the measurement of the average Euclidean distance between the points in a true Pareto Optimal Fronts and the closest solution in the obtained set of solutions, and is to be minimised, where 0 indicates the perfect convergence. This metric has stronger emphasis on the convergence and uniformity of the points and for practical problems can be calculated according to [22]. Different population sizes have been evaluated and 1800 is selected as the best value for cMLSGA and 1000 for other algorithms. The crossover and mutation rates are set as 1 and 0.08 respectively, and the rest of algorithm-specific operation parameters are set as in the original publications. For all cases the objective normalization strategy taken from [13] is used.

5 Automated Design of Yacht Layouts

Three case studies are used to test the performance of the algorithms and investigate the potential complexity that Genetic Algorithms are capable of solving. A 24 m, 50 m and 140 m yacht are compared, each with increasing complexity, and the top Genetic Algorithms are benchmarked and the resulting layouts compared.

5.1 Superyacht 1 - 24 m

The 24 m superyacht forms the simplest problem with 4 objectives, 7 compartments and 49 variables. A leaderboard of the algorithms is shown in Table 3. In this case the cMLSGA is the highest performing of the algorithms showing the best performance on the IGD and HV metrics; the lowest performer is MOEA/D-MSF using both metrics for comparison. In this case the strongest convergence improving algorithm of the co-evolutionary approaches, BCE, shows a stronger performance, with HEIA performing only 4th for the IGD metric and 3rd for the HV metric. The strongest many-objective algorithm from the Evolutionary Computation literature, U-NSGA-III, performs quite poorly, despite the problem having 4 objectives. MOEA/D-MSF, which has the strongest convergence of the different algorithms, performs the worst with a similar IGD to HEIA but with a much lower HV score, reflecting its poor diversity retention.

Table 3. Performance metrics for the 24 m superyacht, values indicate the mean and standard deviation for each metric.

Ranking	IGD	HV
1	cMLSGA 0.2055 0.0303	cMLSGA 1.3157 0.1621
2	BCE 0.3193 0.0822	BCE 1.0797 0.2418
3	U-NSGA-III 0.4012 0.1742	HEIA 0.8344 0.3194
4	HEIA 0.4291 0.1507	U-NSGA-III 0.8050 0.3258
5	MOEA/D-MSF 0.4749 0.1465	MOEA/D-MSF 0.6247 0.1770

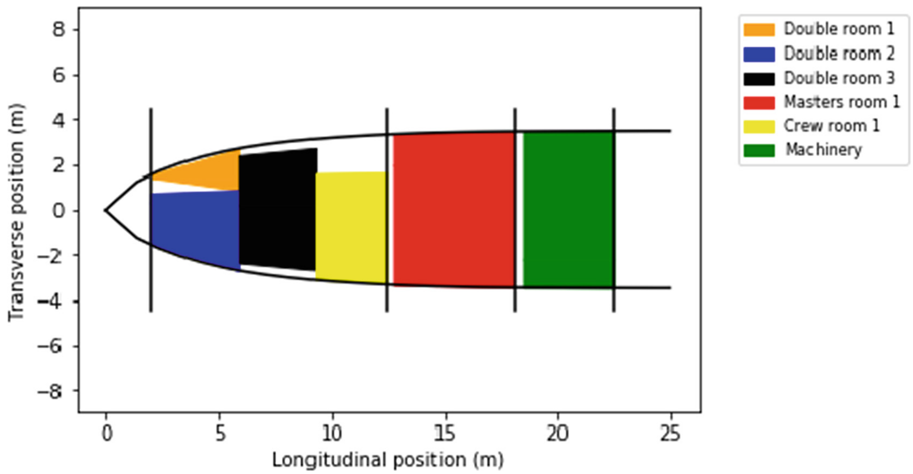

Fig. 3. Automated layout design for the 24 m yacht, black lines demonstrate the position of the bulkheads.

A figure representing the layout for the best point on the Pareto Front is shown below, Fig. 3. The vessel shows a rough layout with the positions of most of the rooms providing a design that is reasonable for this type of yacht. The machinery is at the back and there is a large master room. However, the crew is sandwiched between the masters room and the main rooms, which is unrealistic as the crew is likely to be away from the customers. In addition the rooms themselves are not making best utilization of the space nor are they sensible shapes, especially in the case of the double room 1. This provides a worse layout than the previous research in Sobey et al. [8] for the same number of function calls and the same boat size.

5.2 Superyacht 2 - 50 m

The 50 m superyacht provides a more complex design space with 4 objectives, 10 compartments and 70 variables. A leaderboard of the algorithms is shown in Table 4. In this case the cMLSGA is again the highest performing of the algorithms showing the best performance on the IGD and HV metrics, while MOEA/D-MSF provides the worst performance of the algorithms. The separation between cMLSGA and the other algorithms is greater on this problem, with a larger difference to BCE on the IGD metric and to HEIA on the HV metric. Since HEIA has better diversity retention than BCE it illustrates that this is becoming a factor in the performance of the Genetic Algorithm, with the larger search space requiring the successful algorithms to search more widely to find the optimal solutions. In this case U-NSGA-III is performing almost as badly as MOEA/D-MSF, despite there being 4 objectives, with a wide gap to the co-evolutionary and multi-level approaches. MOEA/D-MSF, which has the strongest convergence of the different algorithms, performs the worst again and in this case the HV metric shows a large standard deviation of 30.299, reflecting the fact that on some runs the performance was extremely poor.

Table 4. Performance metrics for the 50 m superyacht

Ranking	IGD	HV
1	cMLSGA 0.2674 0.0626	cMLSGA 1.5176 0.1669
2	BCE 0.5205 0.0473	HEIA 0.9392 0.1830
3	HEIA 0.5309 0.0627	BCE 0.8929 0.2385
4	U-NSGA-III 0.6102 0.0708	U-NSGA-III 0.7197 0.2480
5	MOEA/D-MSF 0.6202 0.1062	MOEA/D-MSF 0.7112 30.299

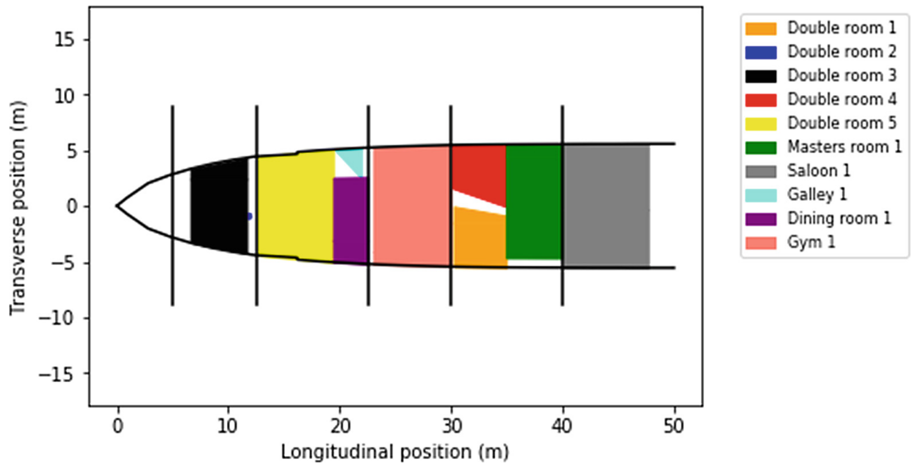

Fig. 4. Automated layout design for the 50 m yacht, black lines demonstrate the position of the bulkheads.

In this case the optimisation provides a reasonable design with most of the area covered and less available space than the previous design, shown in Fig. 4. However, the galley and dining room are small and the double room 2 is a point, hidden next to the double room 3 at the front of the vessel. In addition the size of the gym is large, taking up more space than is perhaps realistic. The galley, double room 4 and double room 1 also have unrealistic diagonal walls to the cabins. The overall design is further from a reasonable starting point for a concept design than the 24 m yacht due to the additional complexity provided by the additional rooms. Further function calls would be required to solve this problem satisfactorily.

5.3 Megayacht - 140 m

The final study is for a Megayacht design of 140 m, which is large for the leisure boat industry. However, this design also starts to represent the size and complexity expected from many commercial vessels. The optimization requires 8 objectives and 399 variables to optimize the 57 different compartments. In this case many of the solvers had started to run inefficiently, with noticeably faster performance from cMLSGA than the others. Due to this inefficiency only 50,000 function calls were made to allow the simulations to be made in a feasible time frame. The results for the different algorithms are shown in Table 5. These results show that cMLSGA provides the best convergence and uniformity with leading performance on the IGD metric. However, in this case it is only the second best performer on the HV metric. BCE is a stronger convergence based algorithm than cMLSGA and it appears that with less function call that the separation between these two algorithms is lower. In addition U-NSGA-III performs more poorly, with the worst results for IGD and second to worst on HV. This means that MOEA/D-MSF is no longer the worst performer and this might be that the convergence first nature of this algorithm is preferred in scenarios where there are less function calls available.

Table 5. Performance metrics for the 140 m megayacht

Ranking	IGD	HV
1	cMLSGA 0.7424 0.3180	BCE 1.3228 0.0773
2	HEIA 0.8531 0.2174	cMLSGA 1.2618 0.2328
3	BCE 1.4735 0.1586	MOEA/D-MSF 1.2225 0.2918
4	MOEA/D-MSF 1.5701 0.0682	U-NSGA-III 0.9545 0.2232
5	U-NSGA-III 1.6429 0.0892	HEIA 0.8629 0.3986

In this case the design still provides a relatively good packing of the boat layout, with a small quantity of unutilized space. However, it results in an extreme situation with some of the rooms filling out the area between the bulkheads and providing large cabins, Fig. 5. These cabins are often much larger than is realistic for their task, while other rooms are points with virtually no volume. This design is some distance from a feasible concept design, which is likely to be from the reduced function calls and increased complexity of the design. For the bottom deck, Fig. 6 the results are very poor demonstrating that a number of additional function calls are required.

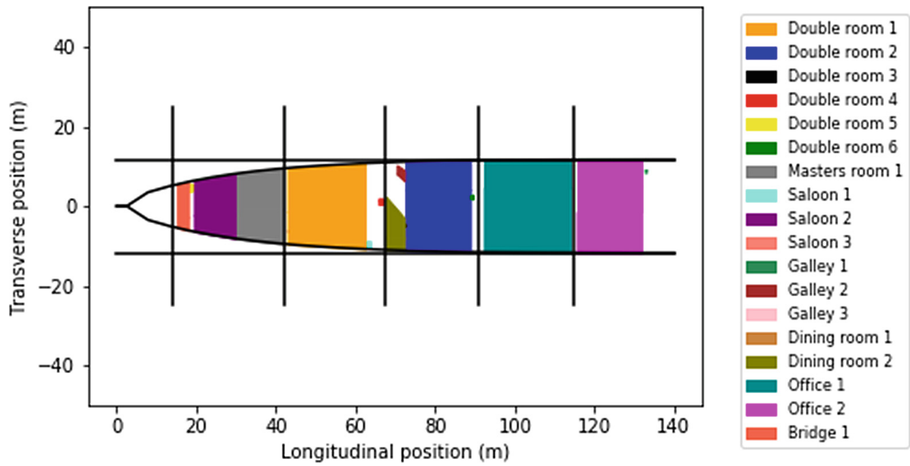

Fig. 5. Automated layout design for the 140 m yacht top deck, black lines demonstrate the position of the bulkheads.

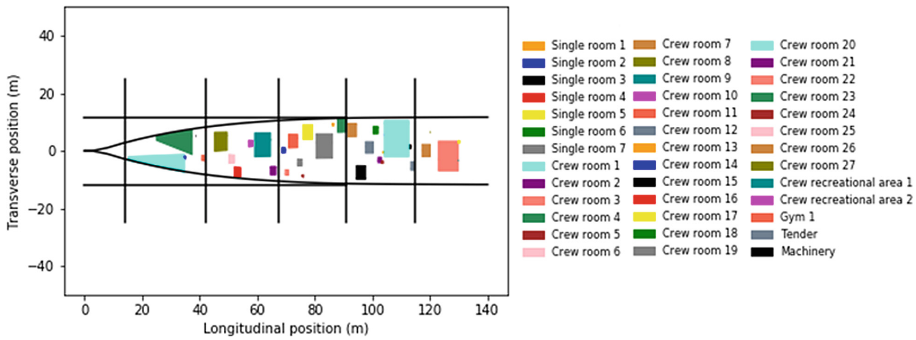

Fig. 6. Automated layout design for the 140 m yacht bottom deck, black lines demonstrate the position of the bulkheads.

6 Discussion and Limitations

The results show that the selection of the Genetic Algorithm is important to resolving the layout for the optimization problem. The state-of-the-art algorithms are already struggling to resolve the Pareto Front by the time the problem reaches 140 m, this is both to do with the representation and the difficult in solving problems of this size. cMLSGA provides the best results for all of the problems with the next best performing algorithm, BCE, also using the co-evolutionary approach which provides a good general performance. The worst performing algorithm is MOEA/D-MSF which is a convergence based algorithm and a specialist solver, which is defined as a solver with a strong performance on some problems and weak on others. This algorithm performed better on the final, 140 m, design where the function calls are reduced, potentially because of its strong convergence first mechanisms that quickly drive it to the front. Interestingly the U-NSGA-III approach performs poorly on the 140 m vessel as this is the highest performing many-objective solver showing high performance on problems with 4+ objectives in the evolutionary computation literature. It also performs poorly on the easier problems, 24 m and 50 m with 4^{th} performance on both, indicating that modern attempts at layout optimization should move to more advanced algorithms than NSGA-II.

The problem representation is adapted to reduce the number of constraints from the previous research [8]. Most Genetic Algorithms perform poorly with a large number of constraints and discontinuous spaces and so the reduction in constraints should allow for a more continuous landscape. However, due to the nature of the layout problem, it appears that the Pareto front itself is discontinuous, despite the lack of additional constraints. The poor performance compared to the previous representation of the problems shows that in some cases, a limited set of constraints may help to reduce the search space, which is highly beneficial. Fully merging the constraints into objective functions, as penalties, may lead to significant increase in complexity of the problem and outweigh the potential advantages of unconstrained optimisation. Therefore the balance between constraints is not trivial and more iterations are required to reach an optimal representation for this type of problem. In addition the designs for the 3 layouts show that the trimming and space utilization constraints are easy to solve. It is a more complex problem to ensure that the rooms are in sensible places and of a reasonable shape and size. This needs further exploration with some adaptions to the representation to provide these benefits either through constraints or through the cabin definition.

7 Conclusion

The automated design of vessels is of increasing interest to ship designers. However, the complexity of these problems is high and the representation of the problem is key to providing a good solution. The use of Genetic Algorithms to solve these problems is common but the algorithms that are used are often not compared to each other or outdated methodologies are utilised. In the results shown here, the best performing Genetic Algorithm is cMLSGA, which is a general solver promoting diversity over convergence. This means that, the presented layout problem, predominantly requires high diversity of the search due to its constrained and discontinuous nature. The representation used for the layout attempts to provide fewer constraints and converts these to objectives as Genetic Algorithms are generally better at solving problems with continuous surfaces. However, combining the constraints with objective functions leads to the increase of the complexity of the problem, which grows to such an extent that the results for even the 24 m problem are poor, and lower in comparison to the 3 constraints representation that has been used previously. This indicates that a small number of constraints may be preferred as they reduces the search space to one that is feasible for the algorithms to investigate and this is necessary for good performance. The recommendation is that cMLSGA should be used for boat layout optimization and that other general Genetic Algorithms with diversity first searches should be promoted on similar problems.

References

1. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**, 712–731 (2007)
2. Jiang, S., Yang, S., Wang, Y., Liu, X.: Scalarizing functions in decomposition-based multiobjective evolutionary algorithms. IEEE Trans. Evol. Comput. **22**, 296–313 (2018)
3. Lin, Q., Chen, J., Zhan, Z.-H., Chen, W., Coello Coello, C., Yin, Y., Lin, C.-M., Zhang, J.: A hybrid evolutionary immune algorithm for multiobjective optimization problems. IEEE Trans. Evol. Comput. **20**, 711–729 (2016)
4. Li, M., Yang, S., Member, S., Liu, X.: Pareto or non-pareto: bi-criterion evolution in multiobjective optimization. IEEE Trans. Evol. Comput. **20**, 645–665 (2016)
5. Seada, H., Deb, K.: U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives. In: International Conference Evolution Multi-Criterion Optimisation, Guimarães (2015)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**, 182–197 (2002)
7. Grudniewski, P.A., Sobey, A.J.: cMLSGA: co-evolutionary multi-level selection genetic algorithm. IEEE Evol. Comput., Under Review
8. While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. IEEE Trans. Evol. Comput. **16**, 86–95 (2012)
9. Wang, Z., Bai, J., Sobey, A.J., Xiong, J., Shenoi, R.A.: Optimal design of triaxial weave fabric composites under tension. Compos. Struct. **201**, 616–624 (2018)
10. Nam, J., Le, T.: Automatic interior space arrangement of mid-sized superyachts using a constraint-based genetic algorithm. J. Mar. Sci. Technol. **17**, 481–492 (2012)
11. Nam, J., Kim, D., Lee, H.: An approach to computerized preliminary design procedure of mid-size superyachts from hull modeling to interior space arrangement. Int. J. Naval Arch. Ocean Eng. **2**, 96–103 (2010)
12. van Oers, B., Stapersma, D., Hopman, H.: An optimisation-based space allocation routine for the generation of feasible ship designs. Ship Technol. Res. **56**, 31–48 (2009)
13. Deb, K.: Multi-objective NSGA-II code in C Revision 1.1.6. Softw Dev by KanGAL (2011). https://www.egr.msu.edu/~kdeb/codes.shtml. Accessed 9 Feb 2019
14. Daniels, A., Parsons, M.G.: An agent based approach to space allocation in general. In: 9th International Marine Design Conference, Ann Arbor, MI (2006)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning. Addison-Wesley, Reading (1989)
16. Sobey, A., Blanchard, J., Grudniewski, P., Savasta, T.: There's no free lunch: a study of genetic algorithm use in maritime applications. In: Conference on Computer Applications and Information Technology in the Maritime Industries, Tullamore, Ireland (2019)
17. Riley, C.: Concurrent engineering for motor yacht design. University of Southampton, Southampton (2015)
18. Daniels, A., Parsons, M.: Development of a hybrid agent-genetic algorithm approach to general arrangements. Ship Technol. Res. **55**, 78–86 (2008)
19. Ehrlich, P.R., Raven, P.H.: Butterflies and plants: a study in coevolution. Evolution **18**, 586–608 (1964)
20. Wilson, D.S., Sober, E.: Reintroducing group selection to the human behavioral sciences. Behav. Brain Sci. **136**, 337–356 (1989)

21. Sobey, A.J., Grudniewski, P.A.: Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm. Bioinspir. Biomim. **13**, 852–862 (2018)
22. Grudniewski, P.A., Sobey, A.J.: Behaviour of multi-level selection genetic algorithm (MLSGA) using different individual level selection mechanisms. Swarm Evol. Comput. **44**, 852–862 (2018)