

Publication P2

Alan Klanac and Jasmin Jelovica. 2009. Vectorization and constraint grouping to enhance optimization of marine structures. *Marine Structures*, volume 22, number 2, pages 225-245.

© 2008 Elsevier

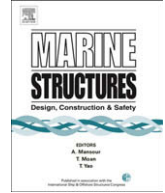
Reprinted with permission from Elsevier.



Contents lists available at [ScienceDirect](#)

Marine Structures

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



Vectorization and constraint grouping to enhance optimization of marine structures

Alan Klanac*, Jasmin Jelovica

Ship Laboratory, Department of Mechanical Engineering, Helsinki University of Technology, P.O. Box 5300/Tietotie 1, 02015TKK, Finland

ARTICLE INFO

Article history:

Received 28 September 2007

Received in revised form 31 July 2008

Accepted 31 July 2008

Keywords:

Optimization

Structures

Vectorization

Constraints

Objectives

Genetic algorithm

Fast ferry

ABSTRACT

Vectorization converts classical scalar optimization formulation, which strictly separates the objective from constraints, into a vector-based optimization, transforming constraints into objectives. Effectively, the search is not conducted anymore for a single optimum, but for a set of Pareto optima between the original objective and transformed constraints. Constraint grouping enhances handling of multiple constraints for vectorized problems, by combining several constraints within a single-objective function, thus reducing the computational time and computational difficulties of high-dimensional spaces created by vectorization. This paper formulates and investigates these two concepts with respect to design of marine structures. It analyses their effects on the possibility to improve the flexibility of optimization in a practical environment, by implementing them within a simple genetic algorithm. Obtained results of vectorization applied to realistic weight optimization problem are encouraging when compared with the results of the classical scalar form optimization, showing a significant improvement in magnitude as well as in reduced computation time needed to reach the optimum.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Optimization is an important part of modern design of marine structures. It helps designer to determine the best combination of parameter values using an appropriate search algorithm. If paired

* Corresponding author. Tel.: +358 9 451 4172; fax: +358 9 451 4173.

E-mail addresses: alan.klanac@tkk.fi, alan.klanac@as2con.com (A. Klanac).

with the holistic experiences of designer, optimization should lead to an increase in product competitiveness. Yet, optimization is regularly laid with challenges. In design of marine structures strength and stiffness formulations are higher order polynomials of design parameters, while the objective functions, e.g. weight or cost, are rarely linear. This situation leads to a non-convex feasible domain of alternatives that cannot be readily handled using the classical ‘text-book’ methods of mathematical programming, e.g. gradient-based methods, linear or quadratic programming. The possibility of premature convergence is then high, banning significant improvements. Yet, if this feasible domain is highly constrained and comprehensive, so that it can be either sequentially linearized or separated into a series of convex problems, the classic methods of mathematical programming can still be utilized. Zanic et al. [1] and Rigo [2] show on different practical applications, including thousands of constraints and hundreds of design variables, that these approaches can result in significant improvements within a favourable computational time period.

However, when problems with non-comprehensive, disconnected or discrete feasible domains occur, e.g. in the conceptual stages of design where design limitations are precisely unknown, especially for vibration-, or production-based limitations [3:588], the optimization of marine structures becomes impractical in the form stated above. But, the aid can be found in the application of multi point exploration methods, e.g. Genetic algorithms (GAs). GAs show the capability to deal with these problems as they operate solely with the objective and constraint values and not with their derivatives; see Refs. [4–10] for some successful applications in design of marine structures. GAs evaluate, compare and operate on a population of design alternatives in parallel, and this specifically allows them to tackle complicated design spaces, but it also potentially leads to a large computational time consumption due to a sheer necessity of evaluating all alternatives. And this precisely should be avoided in the early design stages, where quick computations are essential. Therefore, GAs are often employed to optimize using as few functional evaluations as possible. But, if the population size is reduced, or its diversity is not maintained, GAs will have a difficulty to map the design space properly, generating designs in the close neighbourhood of only one, often unsatisfactory local minimum. Control of the population diversity is performed via GAs control parameters, and their choice is essential for producing good results; see Ref. [5]. But this choice is difficult in practice, as real problems leave no luxury of fine tuning the control parameters. Thus, their number should be reasonably reduced, or their influence on the performance should be decoupled.

To bridge these problems and maintain the positive characteristics of GAs, Klanac and Jelovica [11], instead of devising another algorithm, formulate the concept of vectorization within a simple GA to tackle realistic single-objective structural optimization problems. Vectorization addresses conversion of constraints into objectives, turning single-objective, or scalar optimization into a special class of multi-objective or vector optimization. Hence the name: vectorization. Several authors in Refs. [12–15] discuss similar approaches, indicating improved optimization performance. The choice of using a simple GA instead of a sophisticated and complex counterpart is the result of a reduced number of control parameters existing in it. Even though a counterpart is probably more efficient in its search, the more complex the algorithm, the more parameters it involves. Hence, a desire is to raise the performance level of a simple GA strictly through vectorization.

This GA, named VOP (abbreviating ‘vectorization’ and ‘optimization’), has been applied so far in several studies. Klanac and Jelovica [16] and Besnard et al. [17] respectively present its applicability for structural multi-objective optimization and compare it with other more established commercial codes and tools, such as MAESTRO [18,19] and LBR5 [20]. Other applications of VOP in design of marine structures have also been reported, see Refs. [21,22], and all returned results up to the current standards.

In this paper, besides methodical theoretical argumentation of its causes and effects, vectorization is compounded by another concept of constraint grouping. Constraint grouping is introduced to allow reduction of the total number of objectives created when ‘vectorizing’ large-scale problems with many constraints by combining constraints into one or more additive functions. This quickens the optimization as the problem size and a number of function calls decrease. In that sense, this study shows that by vectorizing and constraint grouping the large-scale structural optimization, applying only a simple GA, cannot only approach the global optimum better, but can do so more time efficiently than if dealing strictly with the original scalar formulations.

The Chapter 2 thus discusses vectorization and introduces the constraint grouping, while in Chapter 3 the extended VOP algorithm is described, capable now of solving scalar, vectorized and constraint grouped formulations. Chapter 4 brings the results of the application of VOP to the discrete single-objective optimization problem of an 88 m long fast passenger/car ferry, while Chapters 5 and 6 close the paper with some reflections, discussion and conclusions.

2. The vectorization and constraint grouping

2.1. Fundamentals

Consider the following relaxed multicriterion, or vector optimization problem (VO) over a vector of design variables x

$$\min_{x \in X} \{f_0(x), \dots, f_j(x), \dots, f_l(x) | j \in [1, l]\}, \quad (1)$$

where any vector x describes one design alternative belonging to the set X which contains all the possible design alternatives between every variable's lower $x_{\min}$ and upper bounds $x_{\max}$. Function $f_j(x)$ is a criterion to be minimized. Obviously, X then contains both feasible, or acceptable, and infeasible design alternatives. Contrary to the typical optimization, which operates on a feasible domain of design alternatives Ω

$$\Omega = \{x \in X | g(x) \geq 0\}, \quad (2)$$

where $g(x)$ is the vector of constraints, VO operates over the infeasible domain, and does not differ between the two. Let now all but one of the criteria, $f_0(x)$, in VO be understood as constraints $g_j(x)$, where $f_0(x)$ is then the objective¹ and let one solution of VO be equivalent to the solution of the following scalar formulations (SO)

$$\min_{x \in \Omega} f_0(x). \quad (3)$$

This procedure for solving single-objective constrained problems using vector optimization is addressed as vectorization. Now, the solution of SO, the optimum x^{**} , is the minimum of the objective f_0 within the feasible domain, while in VO, due to the operation over infeasible domain, it is conditionally Pareto optimal. Therefore, by properly defining the VO and solving it for a set of Pareto optima, or Pareto frontier, it is possible to solve the SO for x^{**} . Generally, any alternative x belonging to a Pareto frontier of VO, $\hat{X}$, is weakly Pareto optimal, where $\hat{X}$ is defined as

$$\hat{X} = \{x \in X | \exists x^k, f(x^k) < f(x), \forall x^k \in X \setminus x\}, \quad (4)$$

and some alternatives are strongly Pareto optimal if

$$\exists x^k, f(x^k) \leq f(x), \forall x^k \in X \setminus x. \quad (5)$$

where f stands for a vector of criteria of Eq. (1).²

2.2. Definition of VO through constraint representation

Structural optimization regularly yields optimal design alternatives on the boundaries of a feasible domain. Therefore, minimizing the constraints in VO will most likely ensure the lower objective values.

¹ Not necessarily all but one criterion in VO need to be considered for constraints. Klanac and Jelovica [16] exploit this fact to perform optimization of both single- and multi-objective problems using a single problem formulation.

² Inequality symbol such as ' $\leq$ ' when used here with vectors considers a specific meaning that vector components on the left can separately be smaller or equal to a criterion on the right. Similar symbols used in the text ' $<$ ', ' $>$ ', ' $\geq$ ' follow the same logic.

However, just simply minimizing the constraints in VO would lead to a general infeasibility. Therefore, the constraints need to be transformed, so that their minimization leads to zero, approaching their boundaries of feasibility. Osyczka et al. [13] and Deb [14] thus represent the constraints with the Heaviside function

$$f_j(x) = H[g_j(x)], \quad (6)$$

where

$$H[g_j(x)] = \begin{cases} -g_j(x), & \text{if } g_j(x) < 0 \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Even though the constraints are relaxed, this representation differentiates between the feasible and infeasible designs and assures that the optimum x^{**} is Pareto optimal in VO. Yet, this representation does not preserve the information on the non-negative deviation from the constraint boundary, so for the designs in the feasible domain, constraint boundaries, as seen in Fig. 1, stretch now inwards over the entire domain, with loss of crispness. Therefore, the actual positions of design alternatives in the feasible domain of the design space become now irrelevant, and designs differ solely on the magnitude of the objective. To avoid this loss of information an alternative – ‘absolute’ – representation is proposed in Ref. [11]

$$f_j(x) = |g_j(x)|. \quad (8)$$

All the information on the position of feasible designs is now preserved, but the clear distinction between the feasible and infeasible designs is dropped, and the infeasible domain in Fig. 1a now mirrors into a feasible part of design space (Fig. 1c), but, preserving the magnitude and crispness of the original feasibility boundary.

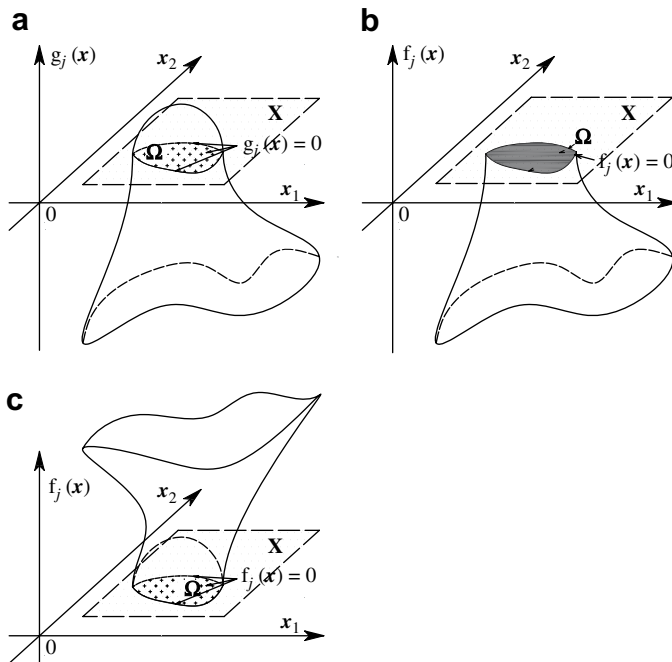


Fig. 1. Effects of constraint transformation of a) original design space applying b) Heaviside representation and c) ‘absolute’ representation.

According to the definition of Eq. (4), the Pareto frontier $\hat{X}$ contains all the global minima of the criteria in VO. But for the ‘absolute’ representation, $\hat{X}$ might then exclude the optimum x^{**} as there exist infeasible designs which probably possess lower values of f_0 and dominate over it. But in that sense, any design alternative ‘sitting’ on a constraint, and thus having

$$f_j(x) = 0 \text{ for any } j \in [1, l] \quad (9)$$

will be at least weakly Pareto optimal and a member of $\hat{X}$. Therefore, if the optimum x^{**} is placed on the boundary of a feasible domain it is guaranteed a weak Pareto optimality, and is a proper solution of VO. Obviously then, by solving the VO with ‘absolute’ representation, the search algorithm will prefer those alternatives which are on the constraint boundaries as they are members of $\hat{X}$, and those which are in their neighbourhood.

Following then the conclusions of Pareto optimality of x^{**} for both Heaviside and for ‘absolute’ representation, it is possible to substitute the search for the scalar optimum in SO with the search for the Pareto front in VO. On the other hand, obtaining the overall Pareto frontier of VO might be too expensive, but also computationally difficult when involving a high number of constraints. It is worthwhile then to map only the desirable area supposed to contain the scalar optimum, and (or) reduce the size of the problem by grouping constraints.

2.3. Constraint grouping

Practical problems often carry high number of constraints and therefore demand, during optimization, a high number of function calls. Also, once vectorized such a problem possesses extremely high number of dimensions in the objective space with possibly excruciating non-linearities. Then again, the same problem could be also simply reduced by grouping some of the transformed constraints, p to q , within the aggregated function $F_p^q(x)$ of their linear sum

$$F_p^q(x) = \lambda_p f_p(x) + \dots + \lambda_q f_q(x). \quad (10)$$

Let then a VO problem be expressed as the reduced problem (VOR)

$$\min_{x \in X} \{f_0(x), F_1^{r-1}(x), F_r^{r+k}(x), \dots, f_l(x)\}. \quad (11)$$

Koski and Silvennoinen [23] prove the validity of the reduced approach for the ordinary multi-objective problems. Based on their conclusions the Pareto optimal solution of VOR is also Pareto optimal in VO, but in general the opposite is not valid. Thus, the Pareto front of VOR is only a part of the overall Pareto front of VO.

Work of Osyczka et al. [13] can be applied to further simplify VOR by grouping all of the constraints into a single aggregated function F_1^l , with $\lambda_1 = \dots = \lambda_l = 1$. The multi-objective problem in Eq. (11) reduces then to the following bi-objective problem

$$\min_{x \in X} \{f_0(x), F_1^l(x)\}. \quad (12)$$

It is interesting also to notice how the Eq. (10) of constraint grouping is equivalent to the Lagrangian dual of the constrained optimization problem. If all the constraints and the objective would be grouped

$$\min_{x \in Q} F(x) = \lambda_0 f_0(x) + \sum_{j=1}^m \lambda_j f_j(x), \quad (13)$$

and this vectorized problem is minimized to the optimum x^{**} , the following Karush–Kuhn–Tucker (KKT) necessary optimality conditions will be satisfied:

$$\begin{aligned} \nabla F(x^{**}) &= \lambda_0 \nabla f_0(x^{**}) - \sum_{j=1}^m \lambda_j \nabla g_j(x^{**}) = 0 \\ \lambda_j &\geq 0, \lambda_j g_j(x^{**}) = 0, g_j(x^{**}) \geq 0, j \in [1, l]. \end{aligned} \quad (14)$$

According to Bazaraa et al. [24], Eq. (13) represents then an unconstrained dual to SO. Therefore, the vectorized optimization problems in Eqs. (11) and (12) are partial duals to the SO, and as any constraint can be a linear combination of some other functions, both vectorized formulations, VO and VOR, are partial duals of SO. In comparison to the 'strict' dual of Eq. (13), a partial dual would contain then, besides the scalar optimum x^{**} , additional equivalent solutions. In this case they are other Pareto optimums of VO or VOR. Also, since partial or total constraint groupings are only partial duals, factors λ can be of any value for the optimum x^{**} and do not need to follow the KKT conditions. Nevertheless, the chosen values of λ would influence on the search process through their effect on formation of the reduced objective space.

To show how the reduced representation of VO affects the possibility to find the scalar optimum and to discuss on the positions of the scalar optimum within both VOR and VO, assume now three Pareto optimal points: $A = (f_0(x^A), F_1^l(x^A))$, $B = (f_0(x^B), F_1^l(x^B))$ and $D = (f_0(x^D), F_1^l(x^D))$, in the attainable space Λ^R of the reduced problem VOR in Fig. 2b, where constraints from Fig. 2b are represented with the absolute function. Let A , represent the scalar optimum x^{**} (see also Fig. 2a), and let B and D possess the properties $f_0(x^B) = \min f_0(x)$ and $F_1^l(x^D) = \min F_1^l(x)$ respectively. Hence, $f_0(x^D) > f_0(x^A) > f_0(x^B)$, and since A contains x^{**} it needs to follow that $F_1^l(x^D) < F_1^l(x^A) \leq F_1^l(x^B)$. Assume now an additional point $C = (f_0(x^C), F_1^l(x^C))$ in Λ^R , having $f_0(x^A) < f_0(x^C)$. Let C contains some k -th Pareto optimal design $\hat{x}^k$ in the objective space Λ of VO. Thus, $f_j(x^{**}) \geq f_j(\hat{x}^k)$ for at least one $j \in [1, l]$, but since F is a surjection it does not strictly follow that $F_1^l(x^{**}) \geq F_1^l(x^k)$, which means that $\hat{x}^k$ is not guaranteed to be Pareto optimal in a reduced problem. As x^{**} is equivalent to $\hat{x}^k$, both being Pareto optimal in VO, having A , as a Pareto optimum point in VOR, contain x^{**} is possible, but not certain. Actually only for the special class of VOR, where all the active constraints at optimum are not grouped, will be possible to guarantee the Pareto optimality of x^{**} in VOR, but knowing these active constraints a priori is practically impossible for realistic problems.

Hence, for the general class of problems the scalar optimum or any other Pareto optimal design in VO is guaranteed to be Pareto optimal in VOR only if it is contained either within B or D in Fig. 2b. And this actually holds for when representing constraints with the Heaviside function. The optimum x^{**} can be readily located because the point A in Fig. 2c will be equivalent to the point D of Fig. 2b, since according to Eq. (7) all feasible designs possess $f_j(x) = 0$, $\forall j \in [1, l]$ and are eventually stacked above the optimum. Point B strictly represents an infeasible design(s) for the Heaviside constraint representation, and probably for absolute representation, because the objectives' minima in X regularly occurs after the constraints are broken, see Fig. 2a, assuming that $Q \subseteq X$. By maintaining the relative positions of the points A – D in Λ^R this bi-objective description is directly expandable for the multi-objective VOR of Eq. (11), and it is also then expandable over and generally applicable for VO of Eq. (1).

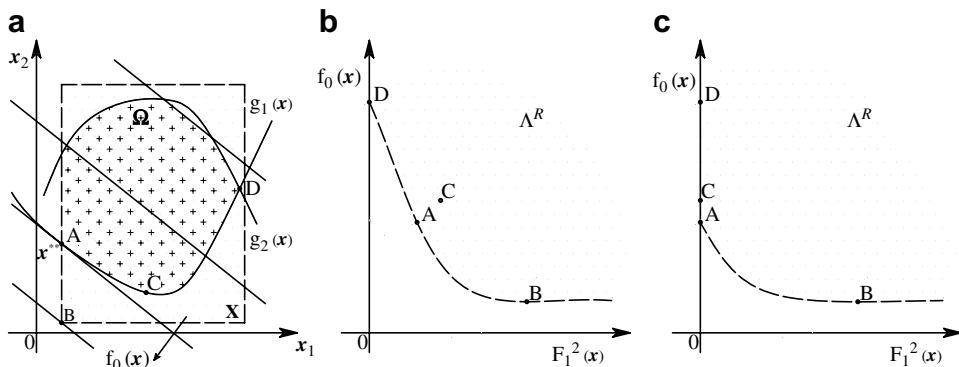


Fig. 2. a) Design and reduced objective spaces applying b) 'absolute' constraint representation and c) Heaviside representation with the characteristic points.

3. Implementation to a simple genetic algorithm

Vectorized optimization problem as presented above could be solved with any multi-objective optimization method, e.g. Ref. [15]. Here however, a simple genetic algorithm, VOP [16], is considered, which capabilities are now expanded to solve both vectorized and reduced vectorized problems. VOP is a binary coded algorithm consisting of:

- generator for the creation of a random initial population of design alternatives,
- fitness calculator for a population of design alternatives,
- weighted roulette wheel selector of design alternatives for mating pool, operating on the basis of computed fitness values,
- sub-routine executing single-point cross-over between the two consecutive alternatives and per-bit mutation,
- main routine running the search process and filtering the infeasible alternatives for the final presentation of results.

All the optimization formulations, the SO, VO and VOR can be solved with this algorithm, by simple variation of the fitness calculator. Therefore, the remainder of this chapter describes the fitness evaluation, while other parts of the algorithm are standard and well known, and can be found in e.g. Refs. [14,25].

3.1. Applying VOP to the scalar formulation SO

For SO, the fitness φ of some design alternative x , within some population of alternatives X^i in the generation i , can be considered through a standard penalty approach based on Refs. [14,25,26].

$$\varphi_1(x, R, i) = \max_{\forall x \in X^i} [\bar{f}_0(x, i) + P(x, R)] - \bar{f}_0(x, i) - P(x, R), \quad (15)$$

where $P(x, R)$ is the penalty function, defined following the Ref. [27]

$$P(x, R) = R \sum_{j=1}^l H[g_j(x)]. \quad (16)$$

The first expression in the right hand side of Eq. (15) determines the maximum of the penalized objective value. This constant is used to convert the minimization problem of SO into the problem of fitness maximization which GAs are usually meant to solve. The applied expression avoids negative fitness values and assigns zero fitness for the 'worst' design alternative. The Heaviside function in Eq. (16) is as in Eq. (7). In Eqs. (15) and (16), as can be seen, both objective and constraints are normalized. $\bar{f}_0(x, i)$ thus stands for a normalized value of the objective, calculated following the Ref. [28]

$$\bar{f}_0(x, i) = \frac{f_0(x) - \min_{\forall x \in X^i} f_0(x)}{\max_{\forall x \in X^i} f_0(x) - \min_{\forall x \in X^i} f_0(x)}. \quad (17)$$

Generally, the constraint functions in optimization of marine structures are defined in the form of the difference

$$g_j(x) \equiv a_j(x) - b_j(x) \geq 0, \quad (18)$$

where $a(x)$ indicates some structural capacity such as the critical buckling stress, or the actual plate thickness, while $b(x)$ stands for the structural demand caused by operations, e.g. hull girder stresses, or required plate thickness and profile stiffness. Constraints in that format can be then easily normalized, following the recommendations of Deb [14]

$$\underline{g}_j(x) \equiv \frac{a(x)}{b(x)} - 1 \geq 0. \quad (19)$$

All constraint values now become of the same order of magnitude which enables the use of a single penalty factor R for all constraint functions [14]. Since the objective is normalized within a unit interval, where 0 identifies the objective's minimum in a population, and 1 its maximum, the normalized constraint values will be of the same order of magnitude with the objective as well.

This fitness formula penalizes all infeasible designs based on their total normalized distance to the constraints they break, where the severity of penalization can be adjusted with the penalty parameter R . However, its proper size will depend on the problem, and should be chosen carefully. If the sum of the normalized constraint deviations in Eq. (16) is in the same order of magnitude as the normalized objective, higher values of R , e.g. already for 10 and larger, lead to a lesser chance of selecting infeasible designs for mating. A high value of R causes predominantly high fitness values for the feasible alternatives, which then sets them apart from all infeasible alternatives that significantly break constraints. Unfortunately, a high R causes also a 'levelling' of the objective function, so that the differences in its value between alternatives within one population become irrelevant. The same difference in the objective is now worth R times less in fitness than the same difference in constraint deviation. This effectively distorts the objective function so severely that it might attain artificial local optima [29] and cause premature convergence. On the other hand, a low value of R , e.g. at 1 and under, will not distort the objective, but will reduce the distinction between the feasible and infeasible alternatives, possibly yielding completely infeasible populations.

To avoid this difficult choice for the proper value of the penalty parameter, Deb [29] proposed an intelligent solution, in which the penalty parameter is not required to set apart feasible and infeasible alternatives, nor to avoid objective distortion. Following his approach, the fitness function for the problem SO can now be defined with the following function:

$$\varphi_2(x, i) = \begin{cases} \max_{x \in X^i} [f_{\max} + P(x)] - \bar{f}_0(x, i), & \text{if } x \in \Omega, \\ \max_{x \in X^i} [P(x)] - P(x), & \text{otherwise,} \end{cases} \quad (20)$$

where the penalty function is now simply a sum of the normalized negative constraint deviations

$$P(x) = \sum_{j=1}^l H[\underline{g}_j(x)]. \quad (21)$$

$f_{\max}$ is the normalized objective function value of the worst feasible alternative. In difference to the first fitness function, infeasible alternatives are compared now based only on their constraint violations, while the feasibles gain their fitness based on the objective function value. Moreover, feasible alternatives strictly possess better or equal fitness.

3.2. Applying VOP to the vectorized formulations VO and VOR

Vectorized optimization problem is solved utilizing the information from the objective spaces A and A^R . The convenient basis for fitness is then the Pareto optimality within a population of design alternatives, but in this case it is insufficient, as the interest is only in a part of an overall Pareto frontier where the optimum is expected to occur. Accounting for this, the following fitness function, φ_3 , is applied:

$$\varphi_2(x, i) = \begin{cases} \max[d(x, i)] + \frac{1}{d(x, i)}, & \text{if } x \in \hat{X}^i, \\ d(x, i), & \text{otherwise,} \end{cases} \quad (22)$$

which separates designs on the basis of attained Pareto optimality $\hat{X}^i$ for a population of design alternatives $\hat{X}^i$ within a generation i , and ranks alternatives based on the distance $d(x, i)$ to the reference point I in the objective space. Following Ref. [16], I is chosen as the set containing the minimum values of every objective attained within a population of generation i .

$$I(i) \left\{ \min_{x \in X^i} f_j(x), \forall j \in [0, l] \right\} \quad (23)$$

Since all constraints, being grouped or not, are now treated as objectives, they are not normalized according to the Eq. (19), but according to the Eq. (17), so $I = \{0\}$. Following Osyczka et al. [23] the fitness function applies the weighted Euclidean metrics as a measure, so the distance to I is found with

$$d(x, i) = \left\{ \sum_j w_j [\bar{f}_j(x, i)]^2 \right\}^{1/2}, \quad \text{s.t. } 0 < w_j < 1, \quad \sum_j w_j = 1, \quad \forall j \in [0, l]. \quad (24)$$

Similarly to the Penalty-based fitness function of Eqs. (15) and (20), fitness of Eq. (22) combines the information on the original objective and on the constraints. Furthermore, the preferred alternatives – being now Pareto optima instead of feasible – strictly possess higher fitness than the non-preferred as is the case with the Deb's fitness functions. Likewise to the same fitness function, the non-preferred alternatives are effectively forced into feasibility by the assignment of higher fitness to those alternatives which are further away from any boundaries. Besides these similarities, there are a couple of profound differences.

First of all, the use of inverse expression to calculate the fitness of the preferred alternatives pressures the algorithm, due to the quadratic ranking, to avoid selecting any non-preferred alternatives. Usually, such strong preferences are avoided as they cause reduction in population's diversity. But since alternatives are ranked now on the basis of both the original objective and constraints, any design alternative with similar distance will be equally preferred independently if it has low original objective value, or its deviation from the constraint boundaries is small.

The second, key difference to the Penalty-based fitness, relates to the physical characterization of the fitness function in Eq. (22). The algorithm is now, with the choice of weighting factor, instructed to search in the particular parts of the objective space where it is expected to find the original objective's optimum x^{**} . In that sense, for the problems adopting Heaviside representation, the weighting coefficient w_0 of the objective function should be taken as

$$0 \approx w_0 \leq w_j, \quad \forall j \in [1, l], \quad (25)$$

since the relative position of the optimum is known. This concentrates the search within the neighbourhood of the point A , seen in Fig. 2c, close to all the feasible designs, where $f_j(x) \approx 0$, $\forall j \in [1, l]$. But if $w_0 = 0$, the objective's values bear no influence of fitness, hence the minimum constraint on w_0 .

In Fig. 3 the characteristic points, A – D from Fig. 2a and b, are extended with several more to illustrate the biased search for the VO problems applying the 'absolute' constraints representation. Point A is the feasible objective minimum, and as it 'sits' on the boundary of a constraint g_2 (see window), it is also Pareto optimal. Point B , which is infeasible and has the lowest value of the objective for the considered points, is again Pareto optimal due to its position on a boundary of the constraint g_1 . Point D is also Pareto optimal, and has the highest objective value of the Pareto front.

It is inefficient then to focus the algorithm to search strictly for one active constraint boundary, as there might be many to investigate. However, the focus could be shifted more towards a particular area of the objective space, where the values of the objectives are small. GA should then, through generations, notice active boundaries and map the points there as Pareto optimal. The higher fitness will be then given to design alternatives with lower values of objectives. Therefore, an opposite strategy can be applied than for Heaviside representation, in which the importance of minimizing the objective is much greater than minimizing the constraints. This can be formalized as

$$w_0 \gg w_j, \quad \forall j \in [1, l], \quad (26)$$

As the area in focus consists of low objective values, the amount of obtained feasible designs can also be low. As this will inevitably impede the optimization, the fitness function in Eq. (22) can be modified by preferring, besides the Pareto optimality, the feasibility of design alternatives, see Eq. (27). This way Pareto optimal, but infeasible alternatives will be prevented to enter into next generation.

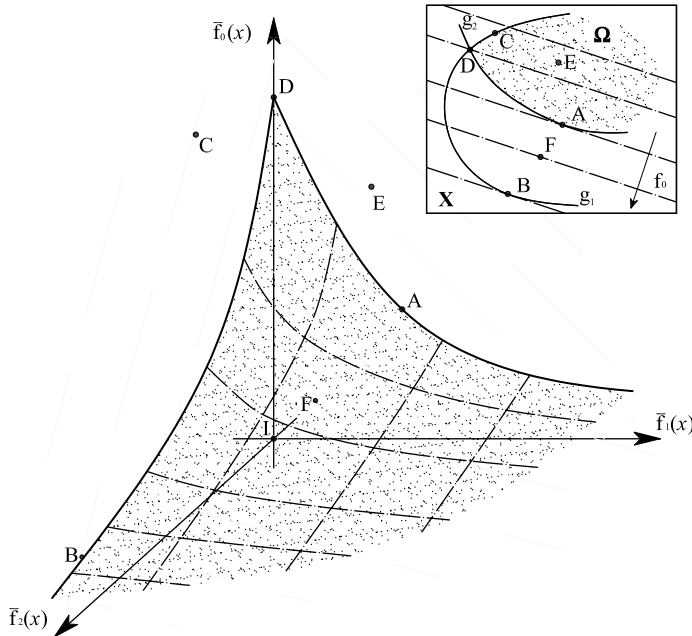


Fig. 3. An objective and a design (in window) space for a vectorized single-objective constrained problem. Notice the correspondence between the objectives f and $\bar{f}_0$, and the constraints g_1 and g_2 and objectives $\bar{f}_1$ and $\bar{f}_2$. Pareto frontier is indicated by a dotted surface.

$$\varphi_4(x, i) = \begin{cases} \max[d(x, i)] + \frac{1}{d(x, i)}, & \text{if } x \in (\Omega \cap \hat{X}^i), \\ d(x, i), & \text{otherwise.} \end{cases} \quad (27)$$

For the VOR problems involving 'absolute' representation, the choice of the w_0 follows that of VO with the same constraint representation. However, as shown in the previous section, the Pareto optimal alternative in Λ is not guaranteed the Pareto optimality in Λ^R . Therefore, applying the Pareto optimality criteria to separate preferred alternatives could be too strict in selecting feasible designs with low objective values. Since distance function is critical for utilizing the information from the objective space, a final fifth fitness function can be considered in the following form:

$$\varphi_5(x, i) = \begin{cases} \max[d(x, i)] + \frac{1}{d(x, i)}, & \text{if } x \in \Omega, \\ d(x, i), & \text{otherwise,} \end{cases} \quad (28)$$

4. Optimization of a fast ferry

A practical example of minimum weight design of a fast ferry (see Refs. [11,16,17]), seen in Fig. 4, is revisited to illustrate the presented vectorization and constraint grouping concepts. Six novel problem formulations are analyzed, as presented in Table 1, combining vectorization and constraint grouping (VO and VOR), and their results are compared with the classical scalar formulation. The scalar formulation is solved considering two different fitness functions φ_1 and φ_2 , as given in Table 2. The idea behind these expanded calculations is also to realize the effects of constraint representation and constraint grouping, as well as the role that the loss of information on both feasibility and infeasibility affects the optimization process and algorithms performance.

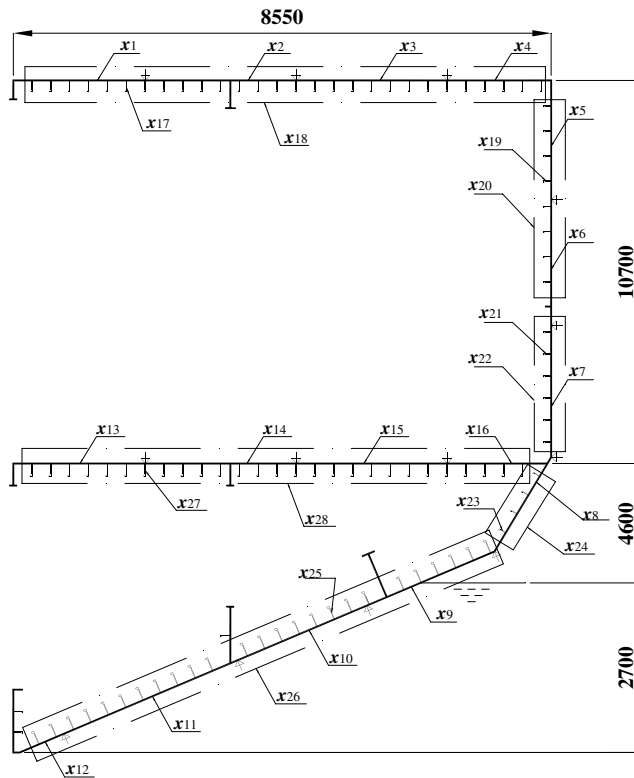


Fig. 4. A half of the ferry's midship section with marked design variables x .

4.1. The structural design model

Ship is assumed to be in the early concept stage of structural design. It is considered in a fully loaded condition, for both crest and hollow landings, with the respective design bending moment amplitudes of $M_{CREST} = 143,778$ kNm and $M_{HOLLOW} = -157,572$ kNm. The axle load of 1.0 t/axle for the car deck, at 4600 mm from the keel, is applied on the tyre print areas of 115×88 mm. The load on the passenger deck is taken as for the weather deck following the assumption that the superstructure does not contribute to the global strength of the ship. Other local loads, such as water pressure are applied according to the Rules [30]. Applied aluminium alloys 5083 and 6082, are used respectively for the plating and stiffeners, with the yield strength of 106 MPa and 84 MPa. Young modulus of $E = 70$ GPa and the Poisson coefficient of $\nu = 0.28$ are the same for both alloys.

Table 1

Considered problem formulations, with characterization, applying vectorization and constraint grouping – VO and VOR.

Constraint representation	Absolute	Heaviside
No constraint grouping	VO-1	VO-2
Partial constraint grouping	VOR-3	VOR-4
Complete constraint grouping	VOR-5	VOR-6
Loss of information	Separation of feasible and infeasible domain	Rate of feasibility

Table 2

Considered problem formulations, with characterization, applying classical scalar optimization – SO.

Consideration of the infeasible alternatives for mating with respect to feasible domain	Separation feasible from infeasibles	Distortion of the objective	Penalty parameter	Formulation
In a wider neighbourhood	Non-existing	Minor	$R = 1$	SO-R1
In a close neighbourhood	Significant	Significant	$R = 10$	SO-R10
In the immediate neighbourhood	Major	Major	$R = 100$	SO-R100
In a wider neighbourhood	Strict	No distortion	–	SO-DEB

Design variables include the scantlings of all the longitudinal elements except the girders, as well as the spacing of the longitudinal stiffeners. Table 7 lists all 28 design variables with minimal and maximal bounds. Generally, the minimum plate thickness of 5 mm is chosen due to a possible severe increase in deformations during the welding of thinner plates. Same wise, the minimum longitudinal spacing is selected at 200 mm.

The objective function f_0 is defined through the sum of the areas $A_i(x)$ of all 29 longitudinal elements, or element groups, in one half of the midship section, including as well the girders

$$f_0(x) = \sum_{i=1}^{29} A_i(x) \quad (29)$$

Constraints are formalized, through the minimal requirements for the thickness and the size of longitudinal stiffeners, given in the rule requirements of DNV for high speed, light craft and naval surface craft [28], namely: part 3 Chapter 3 Section 5, paragraphs B100 and C100 and Tables B1 and C1. In total, there are 24 constraints, 17 linear and 9 non-linear, as presented in Appendix I in their explicit form.

4.2. The GA model

Based on the particular number of constraints, the vectorized problems VO-1 and VO-2 contain in total 25 objectives, while the problems VOR-3 and VOR-4 reduce this to five, as the constraints become grouped according to their physical connotation. The constraint groups are a) the global stresses in the hull girder F_G , b) the required minimal hull sectional modulus F_z , c) the required minimal thicknesses F_t and d) the required minimal sizes of longitudinals and their spacing F_z

$$\left. \begin{aligned} F_G(x) &= \sum_{j=1}^2 f_j(x) \\ F_z(x) &= \sum_{j=3}^4 f_j(x) \\ F_t(x) &= \sum_{j=5}^{18} f_j(x) \\ F_z(x) &= \sum_{j=19}^{24} f_j(x) \end{aligned} \right\} \quad (30)$$

bearing in mind that such grouping is only one possibility. The problems VOR-5 and VOR-6 transformed the multiple objective formulation into an bi-objective. Table 3 presents then the respective number of function calls per one generation needed to determine the considered fitness function.

Variables are binary coded with 4 bit long strings, based on their integer representation, with the step of 0.5 mm for the plate thickness, 0.7 cm^2 for the size of longitudinals and 10 mm for their spacing. A population of 50 design alternatives, or individuals, is created within each generation following the randomly generated initial population. Individuals' chromosomes, or augmented binary strings of all variable values, are mated with the probability of 0.8 using the randomly selected single-point cross-over between the two consecutive individuals in the mating pool. Subsequently, the individuals'

Table 3

Number of functional calls for the considered formulations per design.

	VO-1	VO-2	VOR-3	VOR-4	VOR-5	VOR-6	SO-Rx ^a	SO-DEB
Nr. of calls:								
-objectives	25	25	5	5	2	2	1	1
-constraints	24	0	24	0	24	0	24	24
Fitness function	$\varphi 4$	$\varphi 3$	$\varphi 5$	$\varphi 3$	$\varphi 5$	$\varphi 3$	$\varphi 1$	$\varphi 2$

^a This column considers SO-R1, -R10 and -R100 scalar formulations.

chromosomes are mutated bit-wise with a probability of mutating one bit per chromosome, i.e. 0.009. The applied mating and mutation probabilities follow the recommendations given in Refs. [14,31].

Table 4 lists the applied weighting factors in the considered formulations. VO-1, VOR-3 and VOR-5 use high weighting factors w_0 for f_0 of 0.5, 0.8 and 0.96 respectively, so that the biased search of Pareto optimal designs targets the lowest attainable values of f_0 . VO-2, VOR-4 and VOR-6 on the contrary use the small weighting factor w_0 of 0.05 to bring the search closer to the axis of the original objective. The actual values of the weighting factors have been chosen arbitrarily.

4.3. The results

To provide for better insight into their performance consistency, the considered problem formulations are each run 10 times for 500 generations, every time using different seed for random numbers. The runs are recorded and statistically described. The following 11 measures are applied to capture the particular performance:

- The minimum of the objective function $f_0(x^{**})$ and its generation $\text{gen}_{x^{**}}$ for the best run,
- The difference between the $f_0(x^{**})$ and $f_0(x^{\text{ref}})$ in %,
- The mean μ_{x^*} and dispersion σ_{x^*} of the fittest designs x^* for the best run,
- The objective function value $f_0(x_{1\%}^*)$ of the top 1% designs $x_{1\%}^*$, and their generations of attainment $\text{gen}_{x_{1\%}^*}$,
- The mean $^{10}\mu$ and dispersion $\sigma_{\mu_{x^*}}$ of the μ_{x^*} for all 10 runs,
- The mean $\mu_{f_0(x^{**})}$ and dispersion $\sigma_{f_0(x^{**})}$ of the $f_0(x^{**})$ for all 10 runs,
- The objective function value $f_0(x_{1\%}^{**})$ of the fittest designs $x_{1\%}^{**}$ within 1% of the $f_0(x^{**})$, and their generations of attainment $\text{gen}_{x_{1\%}^{**}}$,

Table 5 presents the measures for the best runs possessing the lowest value of the objective, while Table 6 presents the measures for all the 10 runs. Furthermore, Figs. 5 and 6 respectively illustrate the optimization history for the best runs of vectorized and scalar formulations, specially presenting some of the interesting measures and additionally indicating multiple fittest designs $x_{1\%}^{**}$ within 1% of the obtained minimum objective value. 'Progress' line used in the history plots is a mere indication of dropping objective values and should not be confused with the use of elitism, which has not been applied in the algorithm.

5. Discussion

The results in Table 5 show that the applied GA managed to improve on the referenced design using all 10 considered formulations. The formulation VO-1 performed the best, by reducing the

Table 4

Objective's weighting factors for the vectorized problem formulations.

Weighting factor	VO-1	VO-2	VOR-3	VOR-4	VOR-5	VOR-6
w_0	0.5	0.05	0.5	0.05	0.96	0.05
w_j	0.02	0.04	0.125	0.238	0.04	0.95

Table 5
Optimization results for the best runs.

	VO-1	VO-2	VOR-3	VOR-4	VOR-5	VOR-6	SO-R1	SO-R10	SO-R100	SO-DEB
$f_0(x^{**})$ [m ²]	0.3791	0.3803	0.3800	0.3794	0.3811	0.3795	0.3844	0.4074	0.4175	0.3896
gen _{x^{**}}	496	166	330	457	410	435	326	261	284	262
$f_0(x^{ref}) - f_0(x^{**})$ [%]	10.2%	9.9%	10.0%	10.1%	9.7%	10.1%	8.9%	3.5%	1.1%	7.7%
μ_{x^*} [m ²]	0.3919	0.3915	0.3937	0.3911	0.3931	0.3899	0.1484	0.4365	0.4588	0.4096
σ_{x^*} in % of μ_{x^*}	3.6%	3.3%	4.2%	2.9%	4.2%	2.9%	4.4%	2.8%	3.3%	3.3%
$f_0(x_{1\%}^*)$ [m ²]	0.3792	0.3808	0.3800	0.3794	0.3811	0.3795	0.3897	0.4124	0.4219	0.3906
gen _{$x_{1\%}^*$}	0.3792	0.3808	0.3811	0.3794	0.3811	0.3795	0.3892	0.4132	0.4225	0.3906
	0.3794	0.3808	0.3814	0.3796	0.3811	0.3797	0.3896	0.4132	0.4225	0.3906
	0.3799	0.3810	0.3815	0.3797	0.3811	0.3804	0.3933	0.4136	0.4238	0.3923
	497	173	331	458	411	436	325	251	285	263
	498	176	342	459	412	437	327	400	200	264
	393	177	366	426	413	429	320	249	201	265
	369	249	341	480	414	449	346	497	283	457

cross-sectional area of the midship by 10.2%. It is closely followed by VOR-4 and VOR-6 with 10.1% of improvement, but which have on average for the ten recorded runs, performed the best, on average attaining the best results, as seen in Table 6. Conversely, the worst performing formulation was the SO-R100 which failed to reduce the area for more than 1.1% in the best run. Since scalar formulations depend significantly on the value of the penalty factor R , their attained results range widely, and the optimum reducing the midship area by 8.9% was found using SO-R1 formulation. Yet the worst performing vectorization formulation of VOR-5 still outperformed it by 0.8%.

Considered scalar formulations include different level of the information on infeasible designs into their fitness calculation. If many infeasibles are considered equivalently to feasible designs, the amount of fully infeasible generations becomes high since the algorithm cannot fully distinguish between the feasible and infeasibles. For SO-R1 there were 397 completely infeasible generations, and the search appeared to be erratic, strongly dissipating generations minima x^* , as seen in Fig. 6a, with a high dispersion rate of $\sigma_{x^*} = 4.4\%$ of the mean μ_{x^*} . But in the end, a good optimum was achieved, possibly due to the minimal distortions in the objective function values. Conversely for SO-R100, where the infeasibles designs are strongly separated from the feasible, and severely impeded from further mating, there were only 3 infeasible generations. But due to a strong distortion of the objective, the optimization stalled and underperformed all other formulations. Indicatively, SO-R10 penalizes infeasible alternatives less stringently, with lesser distortion of the objective function, so the optimization behaviour is improved in comparison with SO-R100, keeping, nevertheless, a high rate of feasibility throughout generation. Its results, however, are still worse than for SO-R1. SO-DEB, on the other hand, is the only formulation which fully takes into account the information on infeasible alternatives, similarly to SO-R1, without distorting the objective, but with strict separation between the feasible and infeasible alternatives like in SO-R100. Therefore, there were only two infeasible generations in its best run, and the attained optimum has managed to improve the reference design by 7.7% as seen in Table 5.

The vectorization formulations handle the infeasible design alternatives differently. Treating of the infeasible alternatives is now enhanced through the combination of distance and fitness functions, so the negative influence of either treating infeasibles equivalently or with severe penalization to the feasibility rates or attained optima is minimized. Observing Fig. 5 and the values in Table 5, no major

Table 6
Optimization results for all the computed runs.

	VO-1	VO-2	VOR-3	VOR-4	VOR-5	VOR-6	SO-R1	SO-R10	SO-R100	SO-DEB
$10\mu_{\mu_{x^*}}$ [m ²]	0.3967	0.3983	0.3997	0.3946	0.4001	0.3932	0.4191	0.4414	0.4632	0.4114
$10\sigma_{\mu_{x^*}}$ in % of 10μ	1.3	1.7	2.0	1.0	1.9	1.0	1.0	0.8	1.3	0.7
$10\mu_{f_0(x^{**})}$ [m ²]	0.3870	0.3865	0.3892	0.3839	0.3908	0.3842	0.3904	0.4126	0.4280	0.3935
$10\sigma_{f_0(x^{**})}$ in % of 10μ	1.5	1.1	2.0	1.0	2.1	1.0	0.9	0.8	1.8	0.9

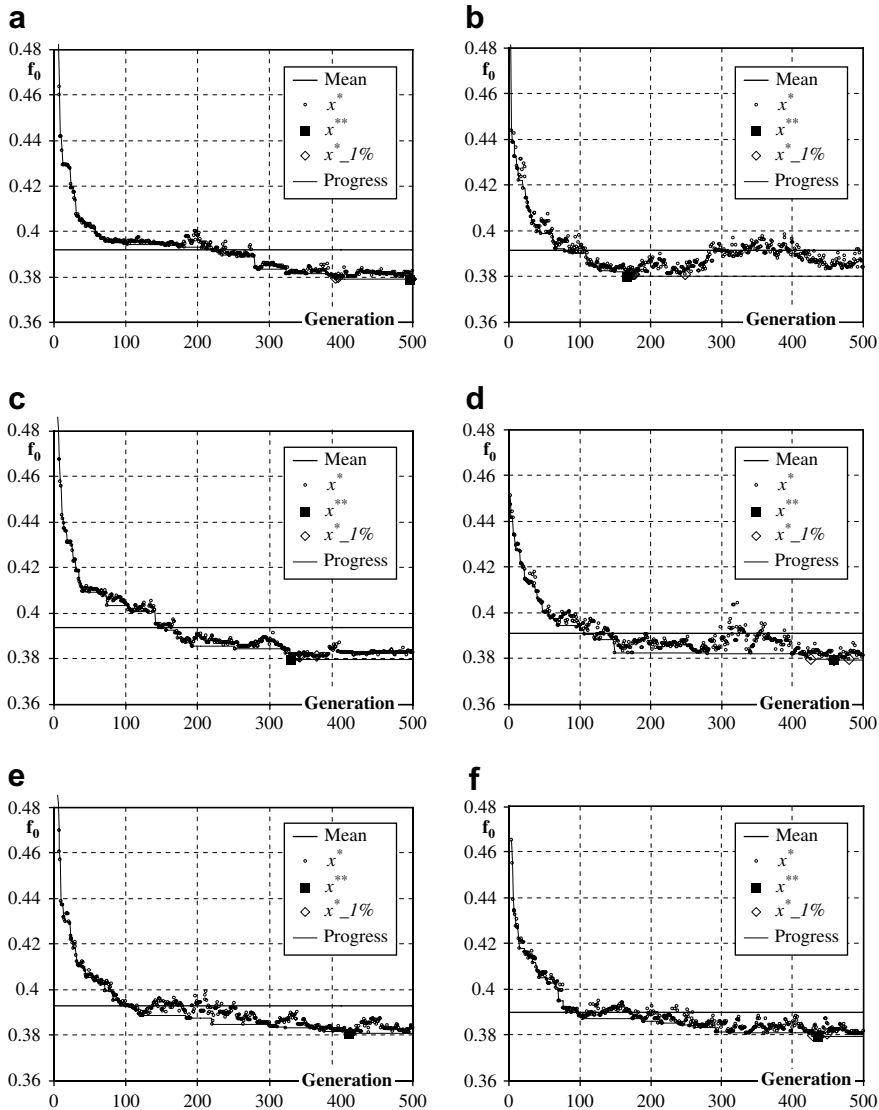


Fig. 5. Fittest design distribution for the best run of a) VO-1, b) VO-2, c) VOR-3, d) VOR-4, e) VOR-5 and f) VOR-6.

differences in the dispersion of generation optima in the best runs can be noticed. Furthermore, these results show no major sensitivity to the type of constraint representation, which leads to the conclusion that the lost information on feasible designs, occurring for VO-2, VOR-4 and VOR-6, is not significantly affecting the optimization performance. Therefore, as vectorization formulations, in contrast to scalar formulations, maintain the information on the rate of infeasibility, but are inconsistent on the rate of feasibility, it seems that the information on infeasible designs is relevant for the improvement of optimization process. But without further mathematical analysis it could be only speculated that the reason for this lies in the position of the optima in structural optimization, being predominantly on the boundaries of a feasible domain.

In addition to vectorization, the applied constraint grouping shortened the optimization time due to the reduced number of function calls and the reduced computation matrices, e.g. in the evaluation of

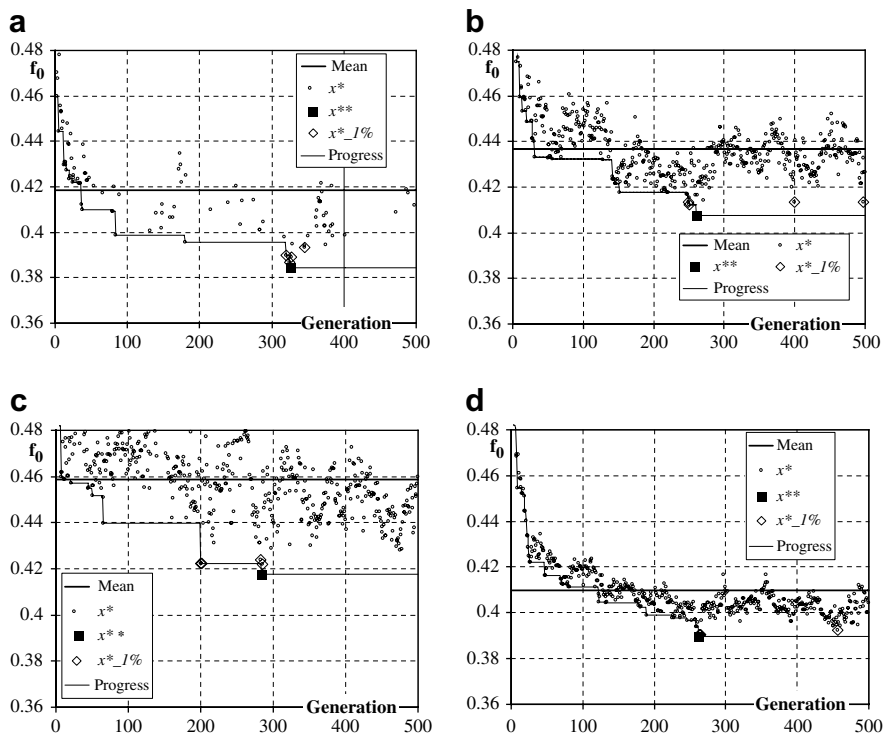


Fig. 6. Fitted design distribution for the best run of a) SO-R1, b) SO-R10, c) SO-R100 and d) SO-DEB.

Pareto optimality, while evidently exerting no major influence onto the performance of the algorithm. This is noticeable through Fig. 5 and Table 3 when comparing the performance and the number of functions calls between e.g. VOR-5 or VOR-6 and VO-1 or VO-2. Therefore, reducing the size of the problem through constraint grouping is a pragmatic approach, fruitful for enlarged problems extending towards hundreds of constraints. To answer on the proper choice of constraint representation is however more difficult, and should be further studied, before making any conclusions.

The vectorized formulations enable also the algorithm to find fitter designs throughout generations than for scalar formulations. See the mean values μ_{x^*} of the best runs in Table 5, but also their means for all the runs $\mu_{x^{**}}$ and the means of the optima $\mu_{f_0(x^{**})}$ in Table 6. The backing for this argumentation is based on generally low dispersion of results regarding the global behaviour of all the considered approaches, as they all show a high possibility to repeat the best obtained values, see $\sigma_{f_0(x^{**})}$ and $\sigma_{\mu(x^{**})}$. Also, the obtained improvements of the objective, at 10%, for vectorization and constraint grouping are in line with the typical improvements in weight minimization found in the literature; see e.g. Refs. [1,2,17].

Table 7 provides a comparison between the obtained minimum weight design, x_{VO-1}^{**} , taken to be the minimum of VO-1, its standardized version x_{stand}^{**} and the referenced design x^{ref} . In comparison with the reference, the optimization reduced the spacing of longitudinals, which then generally caused the reduction in plate thicknesses and in the size of longitudinals. The obtained minimum weight design, x_{VO-1}^{**} , is actively constrained by the longitudinal strength, thus some of the passenger deck strakes, as well as the lowest parts of the bottom shell are thickened in comparison to the reference, while those on the sides have been thinned. Some of the values however, should be considered as initial, due to the application of a conceptual structural model, with a simplified response and strength analysis. Furthermore, the optimization did not consider special production limitations, such as welding of adjoined plates with large difference in thickness. For these reasons, following the optimization, some of the obtained scantlings in x_{VO-1}^{**} were rearranged manually, see Table 7, to represent

Table 7

Design variables with min–max bounds and values for the reference x^{ref} , computed optimum $x_{\text{VOR-4}}^{**}$ and its standardized version x_{stand}^{**}

Design variable	Min	Max	x^{ref}	$x_{\text{VOR-4}}^{**}$	x_{stand}^{**}
Thickness of passenger deck – strake 1, x_1 [mm]	5	12.5	8	5.5	7.5
Thickness of passenger deck – strake 2, x_2 [mm]	5	12.5	8	12.5	9
Thickness of passenger deck – strake 3, x_3 [mm]	5	12.5	8	10	10
Thickness of passenger deck – strake 4, x_4 [mm]	5	12.5	8	5.5	7.5
Thickness of shear strake 1, x_5 [mm]	5	12.5	9	5	5
Thickness of side shell – strake 1, x_6 [mm]	5	12.5	8	5	5
Thickness of side shell – strake 2, x_7 [mm]	5	12.5	8	5	5
Bilge strake, x_8 [mm]	6	13.5	9	6	6
Thickness of bottom shell – strake 1, x_9 [mm]	7	14.5	10	7.5	8
Thickness of bottom shell – strake 2, x_{10} [mm]	7	14.5	11	13.5	9
Thickness of bottom shell – strake 3, x_{11} [mm]	7	14.5	12	7.5	10
Keel plate, x_{12} [mm]	8	15.5	12	15.5	15
Thickness of car deck – strake 1, x_{13} [mm]	5	12.5	8	6	6
Thickness of car deck – strake 2, x_{14} [mm]	5	12.5	8	6	6
Thickness of car deck – strake 3, x_{15} [mm]	5	12.5	8	6	6
Thickness of car deck – strake 4, x_{16} [mm]	5	12.5	8	6	6
Size of passenger deck longitudinals, x_{17} [cm ²]	5.4	15.9	9.31	6.10	6.20
Spacing of passenger deck long's, x_{18} [mm]	200	350	300	200	200
Size of upper side shell longitudinals, x_{19} [cm ²]	5.4	15.9	6.20	5.40	5.40
Spacing of upper side shell long's, x_{20} [mm]	200	350	400	230	230
Size of lower side shell longitudinals, x_{21} [cm ²]	5.4	15.9	6.20	5.40	5.40
Spacing of lower side shell long's, x_{22} [mm]	200	350	350	230	230
Size of bilge longitudinals, x_{23} [cm ²]	5.4	15.9	6.20	7.50	7.74
Spacing of bilge longitudinals, x_{24} [mm]	200	350	350	220	200
Size of bottom shell longitudinals, x_{25} [cm ²]	5.4	15.9	12.40	13.10	13.80
Spacing of bottom shell longitudinals, x_{26} [mm]	200	350	300	240	240
Size of car deck longitudinals, x_{27} [cm ²]	5.4	15.9	12.4	5.40	5.40
Spacing of car deck longitudinals, x_{28} [mm]	200	350	300	220	220
Total area of a half of the midship section [m ²]			0.4221	0.3791	0.3798
Improvements to the referenced design [%]			–	10.2	10.0

more standard distribution of material, maintaining the attained objective value. Most notable changes are seen for the strakes in both the passenger deck and the bottom. Within this process the size of the longitudinals was also standardized to fit the profile sizes available in the market [29].

After standardization, the new 'standardized' optimum x_{stand}^{**} ended up being heavier by a margin of 0.2% than the optimum $x_{\text{VO-1}}^{**}$ attained directly from optimization. Obviously, this leads to the conclusion that $x_{\text{VO-1}}^{**}$ is in the neighbourhood of other good solutions which might be more appropriate for production or maintenance requirements which were not considered originally.

6. Conclusion

This paper presented an alternative approach to single-objective optimization problems, addressed as vectorization. Using a realistic problem as an example, it has been shown that consideration of constraints as additional objectives, alongside the original objective function, can exert benefits regarding the achieved minimum weight design. By combining the concept of vectorization and a concept of constraint grouping developed in this study, six alternative formulations to single-objective optimization have been presented. These were then implemented into a simple GA and confronted in the case study with the classic single-objective formulations. The obtained results are encouraging for vectorization, as all the vectorization approaches outmatched the conventional approach, and the obtained minimum weight design was 10.2% better than the referenced design. Also, through the concept of constraint grouping it was possible to significantly reduce the number of functional calls in GA and the computational time needed to compute a designated number of generations. All this was accomplished without any damaging effects on the efficiency of the optimization process.

Beside the enhancements that vectorization and constraint grouping offer for a GA, this study contributed to the description of GAs working principles. It is evident that GA's performance advanced once handling of the information on infeasible domain is improved. Classical scalar formulation bears a brunt of this added information as it loses the ability to penalize properly the infeasible designs, but vectorized formulations handle this information in a different way, not any more through a penalty function, but through the objective space, created between the constraint and objective functions. Applying vectorization one then avoids a tedious search for the optimal penalty parameter required in the classical scalar formulation.

Prior to ending this text, it should be noted that the presented novel concepts still require stringent testing and in-depth analysis of their influence onto optimization process. Several immediate actions could be named: a) application of a more sophisticated GA, b) analysis of constraint grouping and sensitivity of the results to it, c) application of constraint grouping in multi-objective optimization and finally d) application of vectorization and constraint grouping in extremely large realistic problems with hundreds of variables and hundreds of even thousands of predominantly non-linear constraints or objectives.

Acknowledgments

This study was supported by the *MARSTRUCT* network of excellence on ship structures, funded by the *European Union*, and the *Technology Development Centre of Finland – TEKES* and Finnish shipbuilding and shipping industries through the projects *TORMAKE* and *CONSTRUCT*. *Centre for international mobility – CIMO* of Finland also supported the stay of the second author in Finland. For this help they are all gratefully acknowledged.

Appendix I. Constraint formulations for the example fast ferry study

$$g_1(x) \equiv 0 \leq 1 - \frac{157572 \times 10^3}{77.29 \times 10^6 Z_D(x)}, \quad Z_D(x) = \frac{\sum_{i=1}^{29} (I_{\text{down}_i}(x) + I_{\text{position}_i}(x))}{10.7 - \frac{\sum_{i=1}^{29} e_i(x) A_i(x)}{\sum_{i=1}^{29} A_i(x)}}$$

$$g_2(x) \equiv 0 \leq 1 - \frac{157572 \times 10^3}{77.29 \times 10^6 Z_B(x)}, \quad Z_B(x) = \frac{\sum_{i=1}^{29} (I_{\text{own}_i}(x) + I_{\text{position}_i}(x))}{\frac{\sum_{i=1}^{29} e_i A_i(x)}{\sum_{i=1}^{29} A_i(x)}}$$

$$g_2(x) \equiv 0 \leq Z_D(x) - 2.04$$

$$g_4(x) \equiv 0 \leq Z_B(x) - 2.04$$

$$g_5(x) \equiv 0 \leq x_1 - 22.57 \times 10^{-3} x_{18}$$

$$g_6(x) \equiv 0 \leq x_2 - 22.57 \times 10^{-3} x_{18}$$

$$g_7(x) \equiv 0 \leq x_3 - 22.57 \times 10^{-3} x_{18}$$

$$g_8(x) \equiv 0 \leq x_4 - 22.57 \times 10^{-3} x_{18}$$

$$g_9(x) \equiv 0 \leq x_5 - 21.05 \times 10^{-3} x_{20}$$

$$g_{10}(x) \equiv 0 \leq x_6 - 21.05 \times 10^{-3} x_{20}$$

$$g_{11}(x) \equiv 0 \leq x_7 - 21.05 \times 10^{-3} x_{20}$$

$$g_{12}(x) \equiv 0 \leq x_8 - 26.57 \times 10^{-3} x_{24}$$

$$g_{13}(x) \equiv 0 \leq x_9 - 26.57 \times 10^{-3} x_{24}$$

$$g_{14}(x) \equiv 0 \leq x_{10} - 26.57 \times 10^{-3} x_{24}$$

$$g_{15}(x) \equiv 0 \leq x_{11} - 26.57 \times 10^{-3} x_{28}$$

$$g_{16}(x) \equiv 0 \leq x_{14} - 26.57 \times 10^{-3} x_{28}$$

$$g_{17}(x) \equiv 0 \leq x_{15} - 26.57 \times 10^{-3} x_{28}$$

$$g_{18}(x) \equiv 0 \leq x_{16} - 26.57 \times 10^{-3} x_{28}$$

$$g_{19}(x) \equiv 0 \leq (0.014x_{17})^{1.54} - 18.67 \times 10^{-3} x_{18}$$

$$g_{20}(x) \equiv 0 \leq (0.014x_{19})^{1.54} - 63.13 \times 10^{-3} x_{20}$$

$$g_{21}(x) \equiv 0 \leq (0.014x_{21})^{1.54} - 87.53 \times 10^{-3} x_{22}$$

$$g_{22}(x) \equiv 0 \leq (0.014x_{23})^{1.54} - 153.23 \times 10^{-3} x_{24}$$

$$g_{23}(x) \equiv 0 \leq (0.014x_{25})^{1.54} - 153.23 \times 10^{-3} x_{26}$$

$$g_{23}(x) \equiv 0 \leq (0.014x_{25})^{1.54} - 153.23 \times 10^{-3} x_{26}$$

$$g_{24}(x) \equiv \begin{cases} 0 \leq (0.014x_{27})^{1.54} + 827.31x_{28} - 25.37 & \text{if } x_{28} < 250 \\ 0 \leq (0.014x_{27})^{1.54} - 22.06 & \text{if } x_{28} \geq 250 \end{cases}$$

$A_i(x)$ is the cross-sectional area of a structural element or group, $e_i A_i(x)$ is its static moment of area and $I_{\text{own}_i}(x) + I_{\text{position}_i}(x)$ its vertical moment of inertia.

References

- [1] Zanic V, Jancijev T, Andric J. Mathematical models for analysis and optimization in concept and preliminary ship structural design. In: Proceedings of ninth IMAM. Naples: 2000. p. 15–22.
- [2] Rigo Ph. An integrated software for scantlings optimization and least production cost. Ship Technology Research 2003; 50(2):126–41.
- [3] Cho KN, Arai M, Basu R, Besse P, Birmingham R, Bohlmann B, et al. Design principles and criteria. In: Proceedings of 16th international ship and offshore structures congress. Southampton: 2006. p. 543–624.
- [4] Okada T, Neki I. Utilization of genetic algorithm for optimizing the design of ship hull structure. J.S.N.A. Japan 1992;171: 71–83.
- [5] Keane AJ. Experiences with optimizers in structural design. In: Parmee IC, editor. Proceedings of the conference on adaptive computing in engineering design and control 94; 1994. p. 14–27.
- [6] Nobukawa H, Zhou G. Discrete optimization of ship structures with genetic algorithm. Journal of the Society of Naval Architects of Japan 1996;179:293–301.
- [7] Zhou G. A practical discrete optimization approach for ship structures based on genetic algorithms. Ship Technology Research 1999;46(3):179–89.
- [8] Kitamura M, Nobukawa H, Yang F. Application of a genetic algorithm to the optimal structural design of a ship's engine room taking dynamic constraints into consideration. J Mar Sci Technol 2000;5:131–46.
- [9] Klanac A, Kujala P. Optimal design of steel sandwich panel applications in ships. In: Proceedings of ninth international symposium on practical design of ships and other floating structures – PRADS. Luebeck-Travemuende: 2004. p. 907–914.
- [10] Klanac A. A multicriteria redesign of the midship section of an intermodal RO-RO ship. In: Proceedings of 16th symposium on theory and practice in shipbuilding – SORTA. Plitvice: 2004.
- [11] Klanac A, Jelovica J. Vectorization in the structural optimization of a fast ferry. Brodogradnja-Shipbuilding 2007; 58(1):11–7.
- [12] Knowles JD, Watson RA, Come DW. Reducing local optima in single-objective problems by multi-objectivization. In: Zitzler E, editor. EMO 2001, Lecture notes in computer science 1993. Berlin Heidelberg: Springer-Verlag; 2001. p. 269–83.
- [13] Osyczka A, Krenich S, Tamura H, Goldberg DE. A bicriterion approach to constrained optimization problems using genetic algorithms. Evolutionary Optimization – An International Journal on the Internet 2000;2(1):43–54.
- [14] Deb K. Multi-objective optimization using evolutionary algorithms. Chichester: John Wiley & Sons; 2001.

- [15] Klamroth K, Tind J. Constrained optimization using multiple objective programming. *Journal of Global Optimization* 2007; 37:325–55.
- [16] Klanac A, Jelovica J. A concept of omni-optimization for ship structural design. In: Guedes Soares C, Das PK, editors. *Advancements in marine structures*. London: Taylor & Francis; 2007. p. 473–81.
- [17] Besnard N, Codda M, Ungaro A, Toderan C, Klanac A, Pecot F. Benchmark on ship structural optimisation. In: Guedes Soares C, Das PK, editors. *Advancements in marine structures*. London: Taylor & Francis; 2007. p. 453–63.
- [18] Hughes OF, Mistree F, Zanic V. A practical method for the rational design of ship structures. *Journal of Ship Research* 1980; 24(2):101–13.
- [19] Mistree F, Hughes OF, Phuoc HB. An optimization method for the design of large, highly constrained complex systems. *Engineering Optimization* 1981;5:179–97.
- [20] Rigo Ph, Fleury C. Scantlings optimization based on convex linearization and a dual approach – part II. *Marine Structures* 2001;14:631–49.
- [21] Romanoff J, Klanac A. Design optimization of a steel sandwich hoistable car-decks applying homogenized plate theory. In: *Proceedings of 10th international symposium on practical design of ships and other floating structures – PRADS*. Houston: 2007.
- [22] Ehlers S, Klanac A, Tabri K. Increased safety of a tanker and a RO-PAX vessel by implementing a novel sandwich structure. In: *Proceedings of fourth international conference on collision and grounding of ships*. Hamburg: 2007. p. 109–115.
- [23] Koski J, Silvennoinen R. Norm methods and partial weighting in multicriterion optimization of structures. *International Journal for Numerical Methods in Engineering* 1987;24:1101–21.
- [24] Bazaraa MS, Sherali HD, Shetty CM. *Nonlinear programming: theory and algorithms*. New York: John Wiley & Sons; 1993.
- [25] Goldberg DE. *Genetic algorithms in search, optimization and machine learning*. Boston: Addison-Wesley Longman Publishing; 1989.
- [26] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation* 1996;4(1):1–32.
- [27] Deb K. *Optimization for engineering design: algorithms and examples*. New Delhi: Prentice-Hall; 1995.
- [28] Koski J. Multicriterion optimization in structural design. In: Altrel E, Gallagher RH, Ragsdell KM, Zienkiewicz OC, editors. *New directions in optimum structural design*. New York: John Wiley and Sons; 1984. p. 483–503.
- [29] Deb K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 2000;186:311–38.
- [30] Det Norske Veritas. *Rules for classification of high speed, light craft and naval surface craft*. Høvik, 2005.
- [31] Back T. Optimal mutation rates in genetic search. In: *Proceedings of fifth international conference on genetic algorithms*. San Mateo: 1993. p. 2–8.