POLISH MARITIME RESEARCH 2(69) 2011 Vol 18; pp. 3-18
10.2478/v10012-011-0007-x

Multi-objective optimization of high speed vehicle-passenger catamaran by genetic algorithm

Part I
Theoretical background on evolutionary multi-objective optimization

Zbigniew Sekulski, Ph. D.
West Pomeranian University of Technology in Szczecin

ABSTRACT

Real ship structural design problems are usually characterized by presence of many conflicting objectives. Simultaneously, a complete definition of the optimal structural design requires a formulation of size-topology-shape-material optimization task unifying the optimization problems from these four areas and giving an effective solution of this problem. So far, a significant progress towards the solution of this problem has not been obtained. An objective of the present paper was to develop an evolutionary algorithm for multi-objective optimization of the structural elements of the large spatial sections of ships. Selected elements of the multi-criteria optimization theory have been presented in details. Methods for solution of the multi-criteria optimization problems have been discussed with the focus on the evolutionary optimization algorithms. In the paper an evolutionary algorithm where selection takes place based on the aggregated objective function combined with domination attributes as well as distance to the asymptotic solution is proposed and applied to solve the problem of optimizing structural elements with respect to their weight and surface area on a high speed vehicle-passenger catamaran structure with several design variables, such as plate thickness, scantlings of longitudinal stiffeners and transverse frames, and spacing between longitudinals and transversal members. Details of the computational models were at the level typical for conceptual design. Scantlings were analyzed using the selected rules of a classification society. The results of numerical experiments with the use of the developed algorithm are presented. They show that the proposed genetic algorithm can be an efficient multi-objective optimization tool for ship structures optimization.

The paper will be published in three parts: Part I: Theoretical background on evolutionary multi-objective optimization, Part II: Computational investigations, and Part III: Analysis of the results.

Keywords: ship structure; multi-objective optimization; evolutionary algorithm; genetic algorithm; Pareto domination; set of non-dominated solutions

INTRODUCTION

Hull structures are fundamental load-carrying members of modern waterborne vessels integrating the whole structural system, similarly to air, railway and other transportation vessels. Except for specific solutions of the particular types of ships (e.g. bulk carrier, container carrier, ro-ro carrier), the hull of each merchant ship is constructed according to the common, basic design that involves a thin, tightly-closed, watertight coating stiffened by a set of orthogonally intersecting stiffeners and series of bulkheads, Fig. 1.

Fore part of the ship hull is referred to as a bow or fore peak, aft part – a stern or after peak. The lowest part of the hull is called a bottom, side walls – sides, and the top part closing the hull – a deck. Consistently the bottom part of hull shell is referred to as a bottom shell, side part – side shell, deck part – deck shell. Structural components holding the watertight shell are referred to as framing. Hull framing is composed of intersecting transverse and longitudinal beams referred to as members joined with bottom, side and deck plating. Beams oriented across form transverse framing, while those oriented alongside form longitudinal framing. Vertical watertight bulkheads, dividing space limited by hull outer coating to series of sections and compartments, are referred to as watertight bulkheads: transverse and longitudinal. First ones are set in perpendicular planes in bow-stern direction and they divide hull inner space onto several compartments along ship. Longitudinal bulkheads, set parallel to that direction, divide inner volume to series of compartments in ship breadth. Watertight bulkheads are situated between bottom and upper deck ensuring floatability in case of emergency flooding of specific compartments. Apart from vertical diaphragms or bulkheads there are horizontal diaphragms assembled inside the shell referred to as tweendecks or platforms, depending on spatial range and tightness.

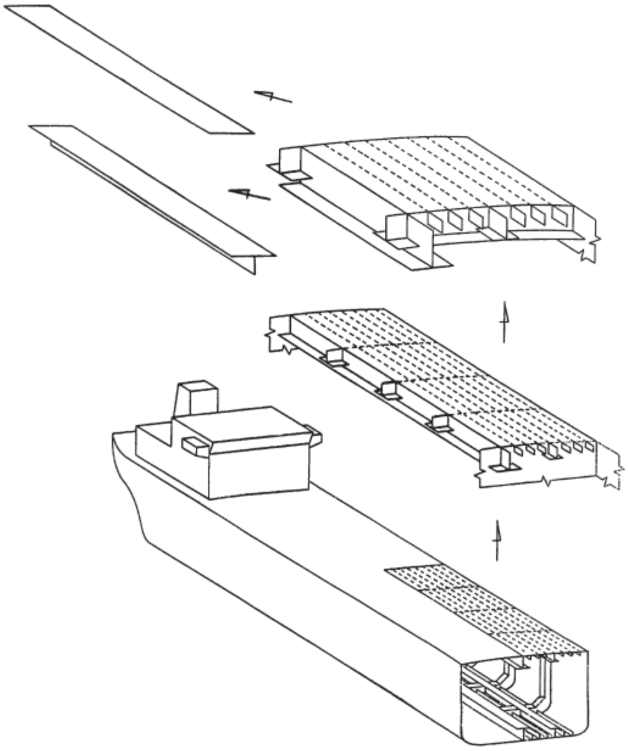


***Fig. 1.** Structural members of ship hull structure*

Structural members are divided into two basic groups. Structural members made from rolled profiles[1)] supporting only shell plating elements are referred to as secondary structural members or stiffeners. Structural members, usually of large size, prefabricated from elements carved from steel plates, supporting shell plating and secondary structural members (stiffeners) are referred to as primary structural members or frames (perpendicular oriented) or girders (longitudinal oriented). Stiffeners supporting bottom shell plating are referred to as bottom frames, whereas transverse oriented bottom primary members are referred to as floors and longitudinally oriented – bottom girders. Stiffeners supporting side shell plating are referred to as transverse or longitudinal side frames – depending on orientation. Transverse primary side members are referred to as web frames, those oriented longitudinally – side girders. Stiffeners supporting a deck are referred to as beams, transverse or longitudinal depending on the orientation, while primary members are referred to as deck transverse, if transverse, or deck girders, if running along the hull. Floors, web frames, and deck transverse are situated in the same intersections (planes) and stiffly joined form main frame.

Transversally oriented ship stiffeners are spaced in regular distances referred to as frame spacing. The frame spacing is determined in the hull structure design process and fundamentally influences its strength and operational characteristics as a whole.

Since the hull of the seagoing ship (1) is the main factor ensuring safety and structural strength, (2) has a very large weight, (3) imposes the framework for internal space arrangement and (4) dictates the basic construction strategy, ship structural design is a key factor in the ship design process. Ship hull structural design always begins with defining design objective. One is supposed to completely and fully specify tasks to fulfill by the structure being designed. It is very rare that the solution of the design process is strictly specified and no variations from initial assumptions are allowed. In most cases precise expression of tasks made in subsequent phase may influence changes in initial concept developed in foregoing phase. Afterwards the actual structural design process begins in which a proper set of component structures are selected and it is specified how they should act jointly. The structural design being developed is subject to continuous qualitative and quantitative analyses. As a result topological description of useful solutions and dimensions of particular component structures are obtained. In this way the ship hull structure is defined in all details. After an operation that is made up from certain number of iteration steps (where loops of these steps differ) a complete structural design is received as a result. After that it remains to expand the information concerning the use of proper technology. It is also needed to confirm that ship hull structure properties meet the requirements specified at the beginning, e.g. with use of numerical calculations or laboratory research. On the other hand detailed design documentation components must be prepared, according to witch the structure is to be realized.

The considered entire process of designing a ship structure is characterized by repeatability of different phases. That repeatability cannot be found in simple sequential form. For example, the analysis of solution concept can influence the task specification, which as a result, may undergo expanding or restriction. Also the design objective can change, e.g. by expanding the range of structure requirements. Structure design may be considered finished if it is presented and accepted by the builder and ordered body. Therefore designing and constructing the whole ship and its hull courses in phases and is a sequent-iterative process. Very often this process is represented in so referred to as "Design spiral" form, Fig. 2.

Thus seagoing ship design process has the sequent-iterative course including concept, structural and detailed design where many operations are repeated with simultaneous increase in work detail and gradual approaching from initial assumptions and basic structure ideas to detail design of structural joints. It is also a hierarchic process with descending structure from more general requirements to detailed design of structural regions. In most general form structure designing is a choice of material and its spatial arrangement in form of structure components (decks, bulkheads, hull sections), which are made of smaller structural elements (stiffeners, plating etc.), Fig. 1. It is supposed to be performed in such a way, that with fulfilled assumed design requirements the hull will ensure safe ship operation at the least costs.

Decisions fundamental for achieving design objectives are taken at the preliminary design phase because at this phase basic ship characteristics are being decided and it is here that the basic risk of making fundamental errors in designing the vessel. Errors made at this phase will influence utilitarian and operational characteristics and actually will not be able to be compensated in later optimization of details on lower hierarchic levels of structure. Not removed will cause the highest losses. At the same time quick and accurate decisions bring the highest benefits and decisively influence total success of the enterprise that is bound with investing substantial amount of money in building a new ship. In this paper it was assumed

[1)] Steel profiles are produced in rolling process, while profiles made from aluminum alloys are produced through extrusion.

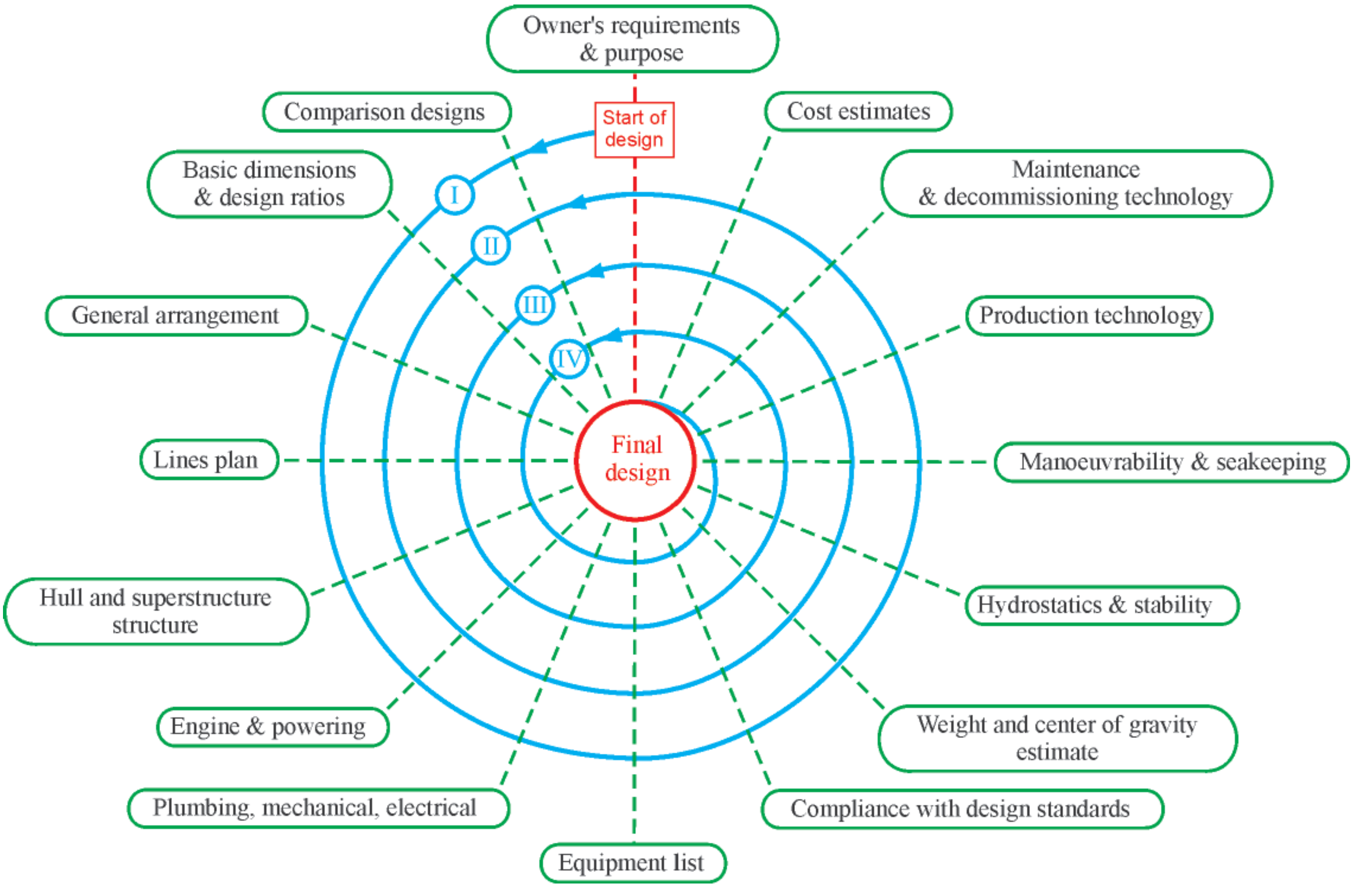

Owner's requirements & purpose
Comparison designs
Start of design
Cost estimates
Basic dimensions & design ratios
Maintenance & decommissioning technology
I
II
III
IV
General arrangement
Production technology
Lines plan
Final design
Manoeuvrability & seakeeping
Hull and superstructure structure
Hydrostatics & stability
Engine & powering
Weight and center of gravity estimate
Plumbing, mechanical, electrical
Compliance with design standards
Equipment list

Fig. 2. *Ship design as the process of achieving the next design levels in several approximations - ship design spiral; designation of ship design phases:* ***I)*** *concept design,* ***II)*** *preliminary design,* ***III)*** *contract design,* ***IV)*** *basic design,* ***V)*** *detailed engineering design,* ***VI)*** *as fitted documentation or delivery documentation*

that the research will be conducted in a field proper for ship preliminary design.

Considering dimensioning of structural elements, the basis for the structural design are either requirements of classification societies (e.g. Det Norske Veritas, Lloyd's Register, Germanischer Lloyd) published as classification rules, containing simplified formulas for evaluating structural loads and dimensioning structural elements resisting these loads, or more rational approach based directly and fully on the structural mechanics. Regardless of occasionally formulated criticism of the classification rules as a tool not meeting the designers expectations in case of designing innovative solutions, they are, however, a set of recommendations derived from the good practice and their application in case of conventional ships is fully satisfactory. In the present paper the author assumes that dimensions of structural elements are determined only according to the requirements of a classification society enabling quick and automatic dimensioning of many structural variants by an optimization algorithm.

A primary objective of the ship structural optimization is to find the optimum positions of structural elements, also referred to as topological optimization, shapes (shape optimization) and scantlings (sizing optimization) of structural elements for an objective function subject to constraints. Formally, selection of structural material can also be treated as a part of the optimization process (material optimization). The topological optimization means searching for the optimal existence and space localization of structural elements while the shape optimization is searching for the optimal shape of a ship hull body. The sizing optimization can also be expressed as a process of finding optimum scantlings of structural elements with fixed topology and shape. Selection of the structural material is usually not an explicit optimization task but is rather done according to the experience and capability of a shipyard. Systematic optimization procedures for the selection of structural material are applied directly in rare cases.

Moreover the design of such a complex object as seagoing ship structure is a solution of the multi-objective optimization task including many optimization criteria often counteracting each other, e.g. small hydrodynamic resistance vs. large cargo deadweight, high structural strength and reliability vs. low structural weight. This requires a comprehensive search of the solution space, without a capability to select one solution unequivocally selected as the best one, as it is in the single-objective optimization tasks. This is because multi-objective optimization does not yield an unequivocal determination of the single variant proposed for further development, but a set of compromise solutions (infinite in general), which is used as a basis for taking the final design decision consisting in a selection of a solution (or solutions) to be further developed. a task of the multi-objective optimization is thus the appropriate identification of the set of "best possible compromises" or the single "best possible solution" as a result of the multi-objective seagoing ship structure design process.

Due to its high complexity, in spite of rising research and computational resources the multi-objective optimization of the seagoing ship structures is still held back by a number of obstacles hindering its application in practice, and the attempts with respect to this problem can be judged to be marginal. Most authors assume that an outcome of the multi-objective optimization task is a set of the Pareto-optimal solutions, while it is impossible to point the objectively best one among them. Classical multi-objective optimization algorithms allow for finding in the best case a single solution in the single algorithm run, which makes them unsuitable for multi-objective optimization tasks involving the determination of the Pareto-optimal solution set. At the same time, the evolution-based algorithms for example allow for the determination of this set in the single algorithm run thanks to the fact that they process not single solutions but usually the large set of potential solutions which in their consecutive steps gradually evolve to a Pareto-optimal set.

A practical example of the application of the developed computational tool is presented, featuring the multi-objective optimization of the structure of fast passenger-vehicle ferry model design named Auto Express 82 m. a task of the two-objective optimization of the ship structure minimizing its weight and surface area for cleaning and painting has been formulated. Number of optimization objectives has been limited to two for the sake of simplicity of graphical presentation of results and their analysis. The precision of the developed computational model has been limited to the level typical for the preliminary phases of the design process for structures of a similar type. The formulated constraints account for structural strength values estimated using procedures laid down in classification rules. A computation application has been built for the solving of so formulated task, being based on a specialized genetic algorithm code. Appropriate models of the ship structures have been built and computational investigations carried out. The obtained results have led to the conclusion that the genetic algorithms can be considered as a method allowing for the solution of the topology-size multi-objective optimization problems formulated for the ship structures in the concept design stage.

Because the prepared paper has considerable sizes it was divided on three parts:

- Part I: Theoretical background on evolutionary multi-objective optimization,
- Part II: Computational investigations,
- Part III: Analysis of the results,

for the publication in the consecutive volumes of the journal.

BASIC CONCEPTS OF MULTI-OBJECTIVE OPTIMIZATION

From mathematical point of view, the multi-objective optimization can be defined in the general way as a procedure consisting in selecting an element of the set on the basis of relations establishing some order in this set. In reference to the ship design the elements of this set are in general the representations of particular problem solutions, such as ship structural variants, various types of ship main propulsor control ensuring that a specified aim of control is attained (e.g. lowest ship operation cost) etc. This set, referred to as "set of possible solutions", is a subset of solution space V_x. As we know, the set of such solutions is limited by the introduction of various constraints and such a constrained set is than called "set of feasible solutions Φ". For obvious reasons set of feasible solutions Φ is also subset of solution space V_x, and each element of this space is a vector of design variables $x \in V_x$, Fig. 3. Solution space V_x may be a functional space or Euclidean space, if all its coordinates are numbers:

$$x = [x_1\ x_2\ ...\ x_i\ ...\ x_n]^T \in V_x \qquad (1)$$

The case where the solution space is a n-dimensional Euclidean space $\Re^n$ is most often encountered in practical applications. In the further part of the paper the solution space is an $\Re^n$ space.

An objective of the multi-objective optimization problem solving can be formulated in the following way: find a combination of design variable values $x = [x_1\ x_2\ ...\ x_i\ ...\ x_n]^T$, which optimizes at the same time all components of a given objective function vector $f(x) = [f_1(x)\ f_2(x)\ ...\ f_s(x)\ ...\ f_S(x)]^T$. With a possibility to impose constraints on the variability ranges of design variables. It is also assumed that all the functions occurring within the problem are real ones, and the number of constraints is finite. Taking into account the demand or computational resources and their cost, another requirement may be formulated, that the selection made could be implemented at the lowest possible cost. Exact definition of the meaning of term „optimize" has crucial significance in case of multi-objective optimization problem. In the further part of the paper this concept is going to be discussed in more detail. The general mathematical formulation of a multi-objective optimization problem can be presented as follows:

- for design variables:

$$x = [x_1\ x_2\ ...\ x_i\ ...\ x_n]^T: \\ x_{i,min} \le x_i \le x_{i,max},\ i = 1, 2, ..., n \qquad (2a)$$

- optimize:

$$f(x) = [f_1(x)\ f_2(x)\ ...\ f_s(x)\ ...\ f_S(x)]^T \\ s = 1, 2, ..., S \qquad (2b)$$

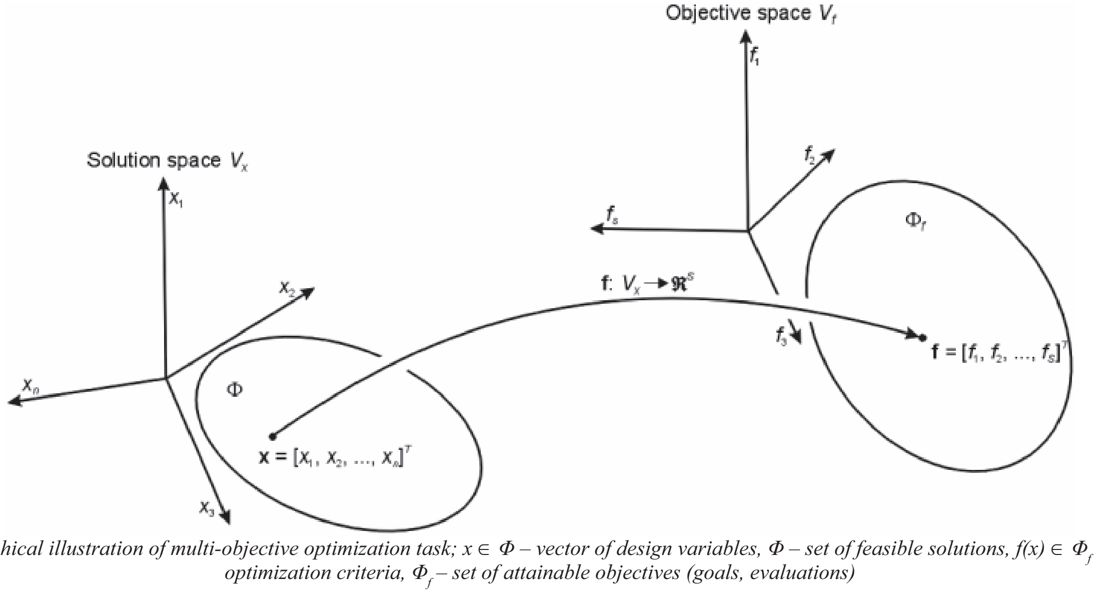

Objective space V_f
f_1
f_2
f_s
f_3
Φ_f
f: $V_x \rightarrow \Re^S$
$\mathbf{f} = [f_1, f_2, ..., f_S]^T$
Solution space V_x
x_1
x_2
x_n
x_3
Φ
$\mathbf{x} = [x_1, x_2, ..., x_n]^T$

Fig. 3. Graphical illustration of multi-objective optimization task; $x \in \Phi$ – vector of design variables, Φ – set of feasible solutions, $f(x) \in \Phi_f$ – vector of optimization criteria, Φ_f – set of attainable objectives (goals, evaluations)

- subject:

$$h_k(x) = 0 \quad k = 1, 2, ..., m_1 \qquad (2c)$$

$$g_j(x) \geq 0 \quad j = 1, 2, m_2 \qquad (2d)$$

where:

$f(x)$ – a single column objective function vector S,

x – a single-column independent variable vector n (design variables),

$h_k(x)$ and $g_j(x)$ – constraint functions

$x_{i,min}$ and $x_{i,max}$ – respectively and upper and lower limit of variables.

Eq.s $h_k(x) = 0$ – referred to as equality constraints, while inequalities $g_j(x) \geq 0$ – inequality constraints.

A solution of multi-objective decision making problem originally formulated by Francis Ysidro Edgeworth in 1881 [Edgeworth (1881)] and generalized then by Vilfredo Pareto [Pareto (1896)] is the commonly accepted measure of quality in the multi-objective selection problems. It is now referred to as Edgeworth-Pareto optimum or Pareto optimum. According to the definition, a design of the seagoing ship structure may be called Pareto-optimal under the condition that there are no other variants of the structure which are better with regard to at least one criterion while at the same time being equally good with regard to all the other optimization criteria. This means that the Pareto-optimal structural variant cannot be improved without simultaneous worsening of at least one criterion. Pareto-optimal designs are also referred to in literature as being non-dominated ones, trade-offs, noninferior or Pareto-efficient. The variant of the ship structure is not Pareto-optimal if there is any other variant improving at least one criterion while at the same time not worsening the values obtained for the remaining ones. Such variants are also called dominated ones or inferior ones.

Using the concept of domination formulated by Pareto we can say that a multi-objective optimal solution is each solution which has no other feasible solutions dominating it. We say that the solution x_1 dominates (is better than) the solution x_2, see Fig. 4, if the following two conditions are satisfied:

$$f_s(x_1) \leq f_s(x_2), \text{ for all } s = 1, 2, ..., S \qquad (3a)$$

$$f_{s'}(x_1) < f_{s'}(x_2), \text{ for at least one } s' = 1, 2, ..., S \qquad (3b)$$

There is then no such a solution in the set of feasible ones for which the value of all criteria would be "better" than their respective values for any multi-objective optimal one. In other words, the multi-objective optimal solution is such feasible solution, for which no better solution can be found in the set of feasible solutions. The word "better" should be understood here in the sense of Pareto domination.

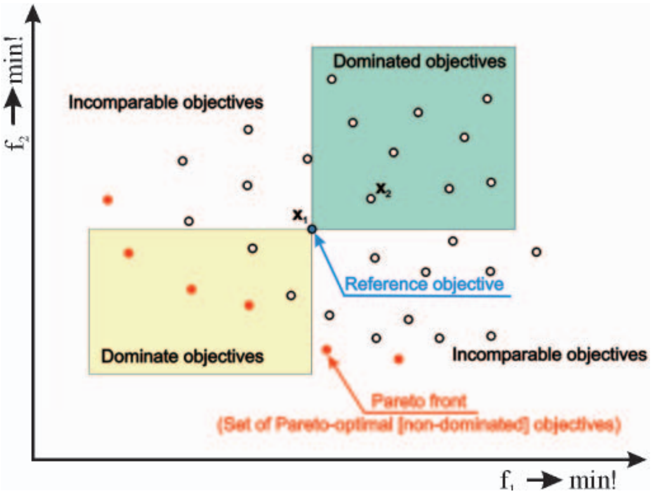
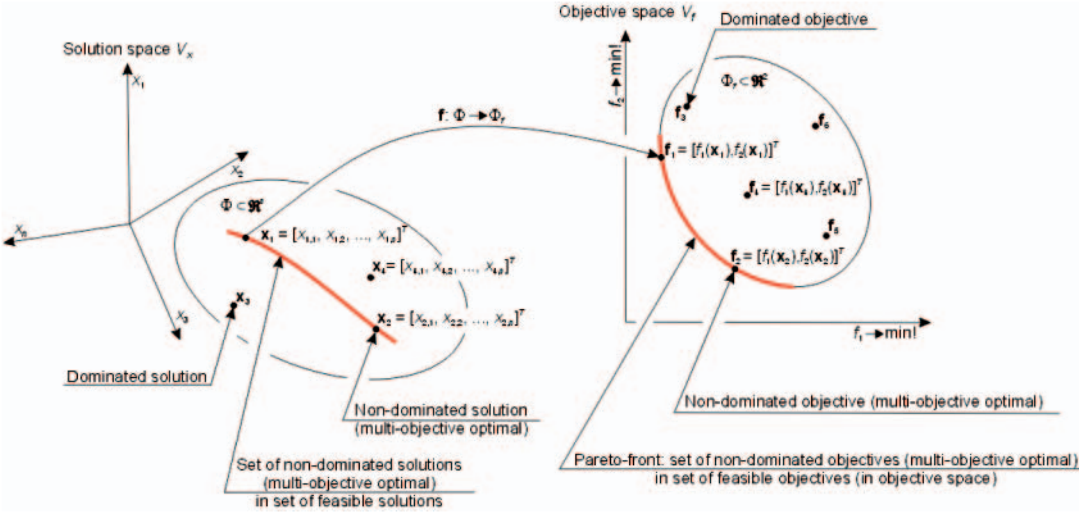


Fig. 4. Graphical illustration of possible relations between solutions in the objective space; the objectives dominated by a reference objective and the objectives dominating it are highlighted; objectives non-dominated by any other ones belonging to the set constitute a set of Pareto-optimal objectives; $f_1 \rightarrow min!, f_2 \rightarrow min!$

The concept of Pareto domination allows for introduction of the two-value measure of quality for solutions of the multi-objective optimization problem. It allows for dividing the set of feasible solutions into two subsets, Fig. 4: (1) subset of dominated solutions Φ_{fd}, and (2) subset of non-dominated solutions, Φ_{fnd} which may be considered to be the solution of a multi-objective optimization problem. Two-value of this measure does not allow for a further evaluation of a feasible dominated solutions set[2), and, particularly, does not allow for relative estimation of distances between dominated solutions and the set consisting of non-dominated solutions (set of Pareto-optimal solutions) by any other feasible solution. In spite of this, the relation of Pareto domination is the one most often used for the definition of multi-objective optimal solution. In the further



Solution space V_x
x_1
x_2
x_n
x_3
$\Phi \subset \Re^n$
$\mathbf{x}_1 = [x_{1,1}, x_{1,2}, ..., x_{1,n}]^T$
$\mathbf{x}_4 = [x_{4,1}, x_{4,2}, ..., x_{4,n}]^T$
$\mathbf{x}_3$
$\mathbf{x}_2 = [x_{2,1}, x_{2,2}, ..., x_{2,n}]^T$
Dominated solution
Non-dominated solution (multi-objective optimal)
Set of non-dominated solutions (multi-objective optimal) in set of feasible solutions
$\mathbf{f}: \Phi \rightarrow \Phi_f$
Objective space V_f
Dominated objective
f_2 → min!
$\Phi_f \subset \Re^k$
$\mathbf{f}_3$
$\mathbf{f}_6$
$\mathbf{f}_1 = [f_1(\mathbf{x}_1), f_2(\mathbf{x}_1)]^T$
$\mathbf{f}_4 = [f_1(\mathbf{x}_4), f_2(\mathbf{x}_4)]^T$
$\mathbf{f}_5$
$\mathbf{f}_2 = [f_1(\mathbf{x}_2), f_2(\mathbf{x}_2)]^T$
f_1 → min!
Non-dominated objective (multi-objective optimal)
Pareto-front: set of non-dominated objectives (multi-objective optimal) in set of feasible objectives (in objective space)

Fig. 5. Graphical illustration of line of non-dominated objectives in the objective space and set of non-dominated solutions in solution space; ***f*** *- vector objective function,* $\boldsymbol{\Phi}$ *- set of feasible solutions,* $\boldsymbol{\Phi}_f$ *- set of feasible objectives;* $f_1 \rightarrow min!, f_2 \rightarrow min!$

[2)] Most detailed analysis of set of feasible solutions in relation to domination. Relationship of domination does not „see" many details of the evaluation space. The only details it can "see" whether the solution is dominated or non-dominated.

part of the work, when talking about domination relation, we shall then understand it to be the relation of Pareto domination, and the earlier used phrase “optimize vector objective function f” shall be understood as a command: find the Pareto-non-dominated solutions within the feasible solutions set.

A basic feature of multi-objective optimal solutions is the fact that there are many (or even infinite number) of them exist in practical problems. In the case of the feasible set is continuous, and is a subset of $\Re^n$, then the set of feasible objectives shall also be continuous and a subset of $\Re^2$ (as a result of two quality criteria). As a result of the analysis of feasible objectives set we can obtain not several multi-objective optimal points but the whole curve of multi-objective optimal objectives, presented in Fig. 5.

SOLVING OF MULTI-OBJECTIVE OPTIMIZATION PROBLEM: HOW FIND SET OF NON-DOMINATED SOLUTIONS

Monographies and books on the general problems of multi-objective optimization are e.g.: [Eschenauer et al. (1990)], [Statnikov et al. (1995)], [Sen and Yang (1998)] and [Stadler (1998)]. Rare works focusing on multi-objective optimization of ship structure include: [Shi (1992)], [Das (1993)], [Das et al. (1993)], [Trincas et al. (1994)], [Ray and Sha (1994)], [Sen and Yang (1995)], [Jianguo and Zuoshui (1996)] and [Parsons and Singer (2000)].

The methods used for the solving of tasks of this type, outlined in monographies and currently applied in practice, my be divided into two basic groups: (1) classical methods, and (2) methods inspired by natural systems, evolutionary methods in the peculiarity. [Cohon (1978)], [Stadler (1988)], [Statnikov et al. (1995)] discuss classical methods, which include two basic methods used for the solving such tasks: (1.1) optimization problems are solved with regard to all optimization criteria taken individually one by one while the remaining criteria are included in the set of constraints, (1.2) a substitute optimization criterion is formed of the adopted ones as a linear combination of the original component criteria multiplied by the appropriately selected weight coefficients, and then the optimization problem is solved with regard to such a newly formed aggregate criterion. For case (1.2) a series of calculations is usually carried out for variously adopted values of weight coefficients, and the best among the found solutions is taken as the solution of the problem. The methods based on a aggregation of the vector objective function have been used in wide-ranging applications also in the methods of evolution-based multi-objective optimization, as they allow for the use of well developed single-objective optimization algorithms. Fundamental disadvantages of methods from this group are: (a) seeking only a single point on the non-dominated solutions front and resulting necessity to make numerous calculation runs for the single optimization task, (b) sensitivity of some solutions to the shape of non-dominated solutions front, and (c) the fact that expert knowledge is required at the beginning to specify the weight coefficients used for component optimization criteria.

Classical methods used for the solving of multi-objective optimization problems based primarily on the aggregation of vector objective functions are easy to implement but ineffective in many cases. However, evolutionary multi-objective optimization algorithms developed in the recent years have been proven to be highly effective in this regard [Deb (2001)], [Osyczka (2002)], [Sarker et al. (2002)], [Abraham et al. (2005)] and [Coello Coello et al. (2007)].

Not numerous but highly promising results in the field of genetic algorithms use for multi-objective optimization tasks have been obtained lately including also the results in the field of ship structures: [Okada and Neki (1992)], [Hutchinson et al. (1998)], [Kitamura et al. (2000)], [Klanac et al. (2009)] and [Sekulski (2010)]. [Jang and Shin (1997)] have applied the Evolutionary Strategy method for the multi-objective optimization of ship structures.

Special evolutionary multi-objective optimization methods can also be applied as far as genetic algorithms are concerned: VEGA – [Schaffer (1985)], HLGA – [Hajela and Lin (1992)], FFGA – [Fonseca and Fleming (1993)], NPGA – [Horn et al. (1994)], NSGA – [Srinivas and Deb (1995)], RWGA – [Murata and Ishibuchi (1995)], MOBES – [Binh and Korn (1997)], SPEA – [Zitzler and Thiele (1998)], MOMGA – [Veldhiuzen (1999)], PAES – [Knowles et al. (1999)], NSGA-II – [Deb et al. (2000)], SPEA2 – [Zitzler et al. (2001)]. Fundamental advantages of these methods are: (1) effective search of solution space and (2) capability to illustrate the non-dominated solutions front in a single simulation run. Excellent presentation of evolutionary methods of multi-objective optimization can be found in recently published monographs [Deb (2001)], [Osyczka (2002)] and [Coello Coello et al. (2007)]. The principal elements of these algorithms are: (1) selection strategies based on the Pareto-domination relation, (2) niching strategies to preserve diversity in the consecutive populations, (3) elitist strategy to ensure survival of non-dominated solutions in the time of evolution. Methods based on aggregation of the objective function are considered less effective and are closed and no more developed stage of the history of the evolutionary multi-objective optimization. Despite it, we can, however, claim that the algorithms employing aggregation of the objective functions are efficient algorithms transient from the classic methods to the advanced algorithms employing the Pareto-domination relation for the variant selection. The researchers have reported for several years that if the number of the optimization criteria is greater than 3, the methods based on the domination relation turn to be ineffective since together with the increase of the number of optimization criteria the number of non-dominated variants decreases reducing the effectiveness of the selection operator [Hughes (2003)], [Purshouse and Fleming (2003)], [Jaszkiewicz (2004)] and [Hughes (2005)]. The aggregation methods have been found promising again with the hope to: (1) developing more simple and intuitive algorithms than algorithms based on the domination relation, obtaining expertise on the multi-objective ship structural optimization, (2) developing effective algorithms for problems with a large number of the optimization criteria. Of course, it needs to be considered what number of criteria is practically justified. With regard to capability of processing data by human beings and capability to work out decisions it seems that the number of the optimization criteria in practical problems should be between 5 and 7.

Due to the following practical problems (1) lack of information about the actual localization of non-dominated solutions set, and (2) necessity to deploy significant computational resources to solve the multi-objective optimization problem, main effort in the practical evolution-based multi-objective optimization is directed at determining the acceptable approximation of Pareto set instead of accurate composition of this set. With regard to this it can be assumed that in practice the result of multi-objective optimization process is a set of non-dominated solutions called shortly the approximation of Pareto set and not the exact Pareto-optimal solutions set. Practical formulation of multi-objective optimization problem and of attained results should follow this guideline.

Opposite to single-objective optimization problems, where the objective function and fitness function are often the same,

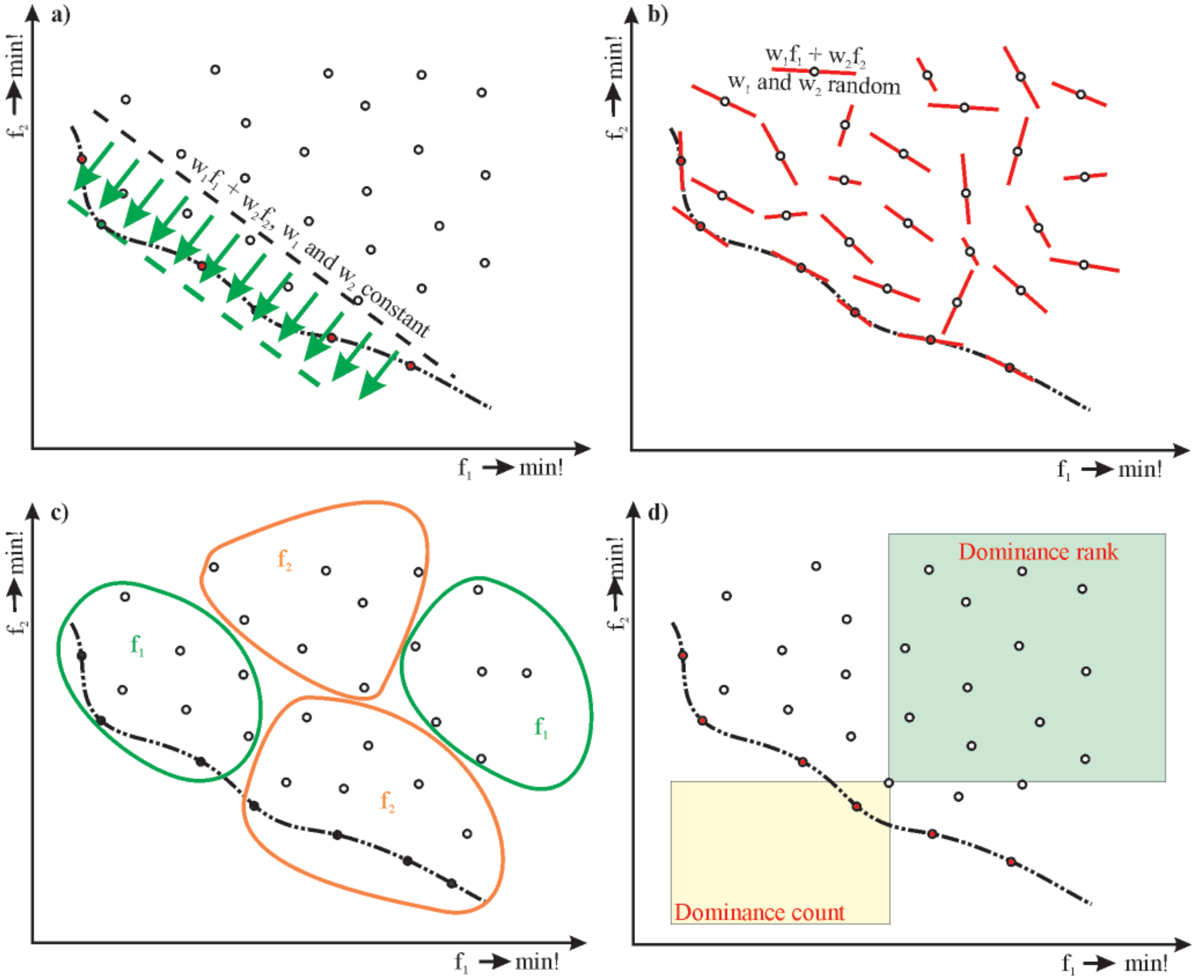


***Fig. 6.** Graphical illustration of selected strategies for taking into account the particular optimization criteria used in the multi-objective optimization algorithms: **a)** selection with respect to the scalar objective function with fixed weights of optimization criteria, **b)** selection with respect to the scalar objective function with random weights of optimization criteria, **c)** division of the variant set into sub-sets and selection in each of them with respect to single criteria, **d)** selection with respect to attributes of Pareto domination; $f_1 \rightarrow min!, f_2 \rightarrow min!$*

the evolution-based algorithms of multi-objective optimization feature both the fitness function and the selection process taking into account a number of criteria which are included in a single fitness function. From this point of view these methods can be in general divided by the type of fitness function used for calculations into the following classes: a) selection with respect to the scalar objective function with fixed weights of optimization criteria, b) selection with respect to the scalar objective function with random weights of optimization criteria, c) division of the variant set into sub-sets and selection in each of them with respect to single criteria, Fig. 6.

First proposal of Fig. 6a, stemming from classical methods used for the determination of compromise surface, consists in summing the criteria up and formulating a single, parameterized objective function. Parameters of this substitute objective function are fixed during the optimization run, which allows for finding the one non-dominated solution. The multi-objective optimization problem is reduced to the single-objective problems, with the criteria being usually referred to as objective functions. The simplest concept is the introduction of objective function F as a linear combination S of partial optimization criteria f_s:

$$F(x) = \sum_{s=1}^{S} w_s f_s(x) \tag{4}$$

where:

w_s – coefficients determining the weights given to particular criteria.

Next proposals, illustrated on Fig. 6b and Fig. 6c, are based on the weighted sum of optimization criteria, where weight coefficients represent the values changing in the process of evolution [Hajela and Lin (1992)], [Ishibuchi and Murata (1996)]. Weight coefficients of the substitute objective function change in a specific way during the optimization run, which allows for finding the non-dominated solutions set instead of a single compromise solution.

Methods based on selection with respect to the scalar objective function with random weights of optimization criteria Fig. 6b employ numerical procedures for setting random values of weight coefficients w_s. The simplest and most frequently applied implementation of the method is setting random values of uniform distribution in range [0, 1].

Methods based on selection according to single criteria as it is illustrated on Fig. 6c consist of mechanisms switching between the criteria during the selection phase. In each case where the algorithm commences the execution of reproduction, some criterion (potentially other) decides which member of population is going to be copied to the set of variants earmarked for crossbreeding. For example, [Schaffer (1985)] proposed an algorithm, where the population is divided in advance to identical parts and then a different, single criterion is used on the members of each of groups one by one and [Kursawe (1991)] proposed a different method consisting in the random selection of a single optimization criterion to be used in the next step of selection process, with probabilities to be set by the user or randomly adjusted during the evolution.

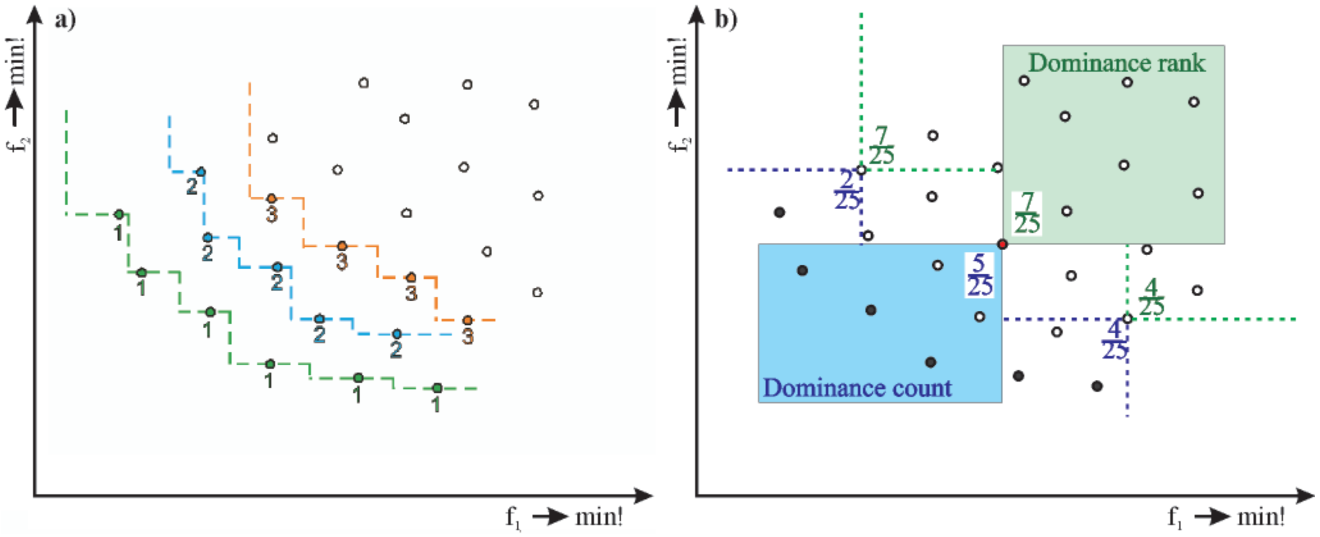


Fig. 7. Graphical illustration of basic concepts used for taking into account of variant domination in the multi-objective optimization algorithms: ***a)*** *domination depth,* ***b)*** *dominance rank, and dominance count; for non-dominated solutions (Pareto-optimal) dominance count equals zero; dominance rank for feasible solutions not dominating any other solution equal zero; $f_1 \rightarrow$ min!, $f_2 \rightarrow$ min!*

Assigned to feasible variants two-argumental domination attribute is a rather general information and does not refer to inner structure of feasible set. Particularly it does not provide knowledge e.g. about (1) what number of variants is dominated by every of feasible variants, (2) what number of solutions is dominated by a given not-dominated variant, (3) how distant is a given dominated feasible variant from Pareto front, etc. Detail study of feasible set structure enables the use of additional knowledge and elaboration of very refined and useful tools supporting desirable convergence of computing algorithm. Different detailed solutions using this concept can be found in bibliography, for example: [Fonseca and Fleming (1993)], [Zitzler and Thiele (1999)], [Deb (2001)], [Osyczka (2002)], [Abraham et al. (2005)], [Coello Coello et al. (2007)].

The concept of calculating the fitness of variants while accounting for the Pareto-type domination, Fig. 6d, has already been proposed by [Goldberg (1989)] and since then taking the advantage of partial order being present in the population, Fig. 7a, has been facilitated by many various methods developed specifically for purpose. Basing on the concept of dominance depth proposed by [Goldberg (1989)], the feasible variants appearing in the consecutive generations are divided into consecutive fronts of non-dominated variants located deeper and deeper in the feasible set, or in other words, further and further from Pareto front. Other proposals are based on the concept of dominance rank, Fig. 7b, which is based on use of the number of variants dominated by a selected variant for the calculation of fitness function value [Fonseca and Fleming (1993)]. Also a dominance count (which is a number of variants dominating a selected variant), Fig. 7b, may be taken into account. For example SPEA strategies [Zitzler and Thiele (1999)] and SPEA2 [Zitzler et al. (2001)] for the calculations of fitness function are based on both approaches, i.e. the rank and the dominance count. Disregarding the specific strategy applied, in case of these methods the value of fitness depends on the characteristics of the variants remaining in the population, in contrast with strategies based on aggregation of objectives or the selection of variants on the basis of less or more arbitrary selected single criteria, where the adaptation values are independent from the characteristics of other variants in the population.

The multi-objective evolution-based optimization algorithms outlined above have been tested by other authors on simple problems of multi-objective optimization eg. [Zitzler (1999)], [Zitzler et al. (1999)], [Zitzler and Thiele (1999)]. As no systematic research into the suitability of these algorithms for the solving of optimization problems involved in the design of seagoing ship structures has been carried out so far, then the application of a particular method should be preceded by systematic research into its effectiveness in the problems involved in the design of such structures.

LOOKING INSIDE THE FEASIBLE SOLUTIONS SET: DOMINANCE RANK AND DOMINANCE COUNT

Let us recall that assignation of a two-argument dominance attribute (0 or 1) to the feasible variants allows for dividing the feasible variants set into two subsets: (0) subset of dominated variants and (1) subset of non-dominated variants, Fig. 8a. As can be seen, this information is quite general and does to refer to the internal structure of the feasible set, and in particular does not provide any knowledge about e.g. the following: (1) what number of variants is dominated by each of feasible variants, (2) what number of solutions is dominated by a given non-dominated variant, (3) how far away from Pareto front lies a given dominated feasible variant, etc. [Goldberg (1989)] proposed for ordering the feasible solutions depending on the depth of consecutive fronts of non-dominated solutions, Fig. 8b. As can be seen, the dominance attribute which in this case is the domination depth takes discrete values: 1, 2, 3, etc. More detailed analysis of the structure of feasible set allows for the use of following concepts as dominance attributes: (a) rank of feasible variants, dominance rank, Fig. 8c, and (b) feasible variant evaluation dominance count, Fig. 8d. Values of both attributes vary in continuous way over the set of feasible solutions, depending on the strategies adopted for their determination.

The dominance rank for a given variant is proportional to the number of feasible variants dominated by a given variant, Fig. 8c. The value of dominance rank is then highest for non-dominated variants approximating the Pareto front. For the variants lying outside of Pareto front (or its approximation), deeper and deeper in the feasible variants set, the values of dominance rank are a falling measure of their distance from the front. The dominance count of a feasible variant is proportional to the number of other feasible variants dominating a given variant, Fig. 8d. Its value is then lowest and equals zero in

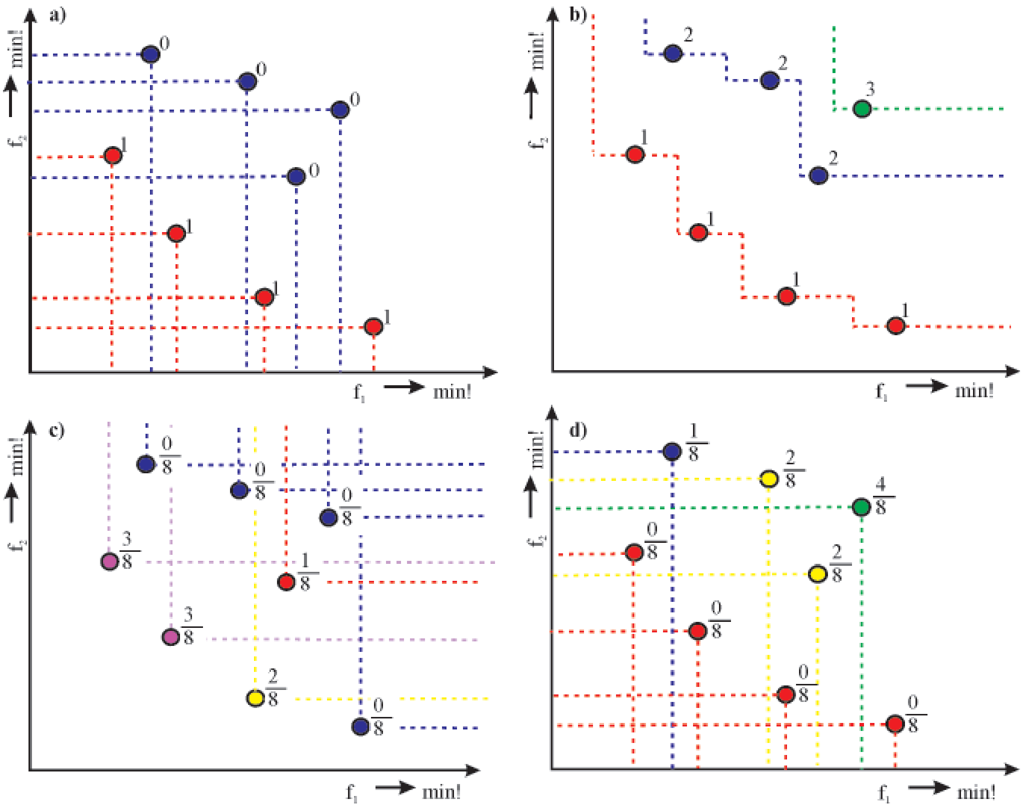


*Fig. 8. Graphical illustration of the groups of dominance attributes: **a)** classifying the feasible variants basing on Pareto-domination relation, **b)** classifying the feasible variants basing on domination depth, **c)** dominance rank (proposed strategy for the determination of a number of variants dominated by a given variant) in the objective space, and **d)** dominance count (proposed strategy for the determination of a number of variants dominating a given variant); f_1 → min!, f_2 → min!*

case of non-dominated variants which approximate the Pareto front. For the variants lying outside of Pareto front (or its approximation), deeper and deeper in the feasible variants set, the values of count are a rising measure of their distance from the front. Other schemes for dominance attributes have been proposed by e.g. [Fonseca and Flaming (1993)], [Zitzler and Thiele (1999)]. Proper application of both attributes allows for development of very subtle and useful tools supporting the desired convergence of the calculation algorithm.

Attention should be paid to the fact that the values of objective function for each variant are determined after its characteristics (evaluation of design criteria and compliance with imposed constraints) are calculated, and the distances between variants in objective space of already determined objective function values are constant and may be used as a measure of absolute mutual similarity of solutions. It is however a different story in case of dominance attribute. It is not an absolute measure of variant location in the objective space, but rather a relative measure of variant location in this space determined in relation to other feasible solutions of the set. As the feasible set continuously evolves, then the dominance attribute of its elements, feasible solutions, evolves as well, because in consecutive generations it is determined over a different set of feasible solutions, in spite of the fact that the objective function values remain unchanged. This is quite obvious: in the population initiating the simulation the first solution (non-dominated) shall be granted dominance depth 1 according to the proposal by [Goldberg (1989)]. As a result of the progress of simulation and the feasible solutions gradually approaching the Pareto optimal front, this solution is going to be overtaken by other ones and its dominance depth is going to drop lower and lower. Moreover, the difference in dominance depth between two solutions is not fixed and may not be used as a relative measure of value for two compared solutions. Of course, in each case the non-dominated solution is better than any dominated one, but it cannot be simply claimed that in case solution B has a dominance depth lower by 1 from solution A, and solution C has a dominance depth lower by 3 from solution A, then the solution B is going to be better than solution C with regard to optimization criteria. Let's analyze the example illustrated in Fig. 9, with initial set from Fig. 9a consisting of solutions A, B and C, where solution a dominates the others, but they do not dominate each other. Solution a has dominance depth equal 1 while solutions B and C have dominance depth 2. Let's then assume that two new feasible solutions D and E are added to the feasible set in the next generation as shown in Fig. 9b. Solution D is dominated by a and at the same time dominates the solutions C and E. Solution E is dominated by solutions a and D and at the same time dominates the solution C. Now, solution C obtains a new value of domination depth, equaling depth 4 (used to have depth 2 before), while the solution B retains dominance depth 2, which takes place in spite of the fact that solutions C and B did not change their locations in the objective space.

Like dominance depth neither dominance rank nor dominance count of a given solution are then objective measures of its value, yet they are practicable and widely used in practice for the evaluation of solutions sought for in the problems of multi-objective optimization.

Various particular methods based on application of these concepts of dominance attributes are found in literature, for example [Fonseca and Fleming (1993)], [Zitzler and Thiele (1999)]. Each such a method has its own particular characteristics, but the early experiences of the author and also other researchers, e.g. [Leyland (2002)] suggest that the influence of particular dominance rank strategies on the convergence of optimization process may be insignificant. In particular, while applying modern elite strategies the solutions quickly converge to non-dominated solutions set.

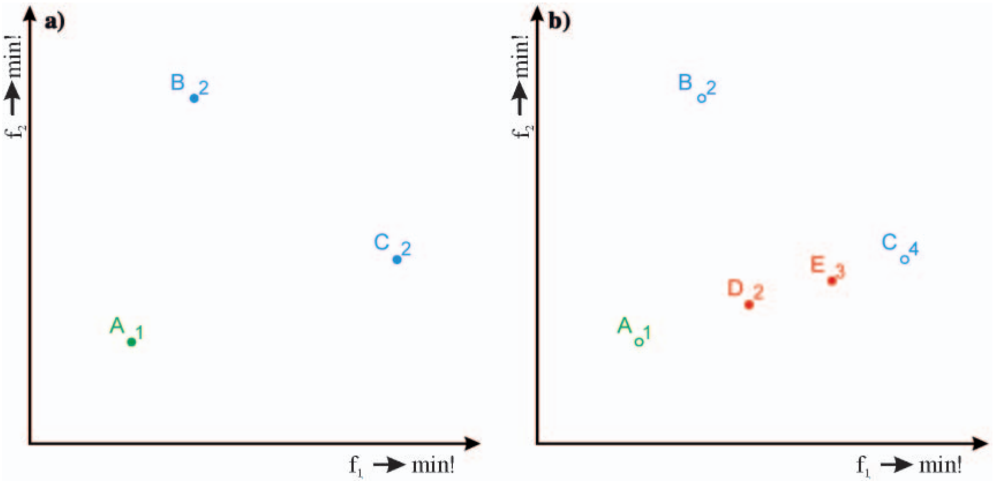


Fig. 9. Illustration of the concept of dominance depth shift without the relocation of solutions in the objective space, for min! type criteria; a) dominance depth values for solutions A, B and C in initial population (preceding one); b) extension of feasible solutions set by solutions D and E does not cause a change of dominance depth of solutions a and B but does cause a change of dominance depth of solution C without causing its relocation in the objective space: what is changed only is the relative location of this solution in relation to the remaining feasible solutions

CALCULATION TOOL FOR EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION OF SHIP STRUCTURE

The essence of the classic Darwin's theory of evolution [Darwin (1859)] is an idea that the evolution of biological organisms is driven by natural selection which affects hereditary traits ensuring survival (in the environment). According to the concept of the natural selection some organisms in certain environmental conditions manage better than others: they survive more often, have more numerous offspring and in time there are more and more of them. Therefore "the environment" selects beings better adjusted to certain situation. If environmental conditions were to change, the organism with a set of traits best suited for adapting to them, it will start to dominate. The concept describing this occurrence in evolutionary biology is adaptation or fitness, which means probability of surviving or reproduction in certain environment. There are mutations in every generation that provide the population with new variants. Natural selection sifts them: environmental severity decreases the number of "bad" ones (relatively unfit) variants and increases the number of "good" ones (relatively fit). Therefore adaptive evolution is a two-phase process in which tasks are strictly divided between mutation and selection. It is worth to notice that there are many variants in the population that can help it adapt to change in environmental conditions simultaneously. Consequently, the better the individual is adapted to environment, the faster its share in population will grow. Natural selection has very good "eyesight", which allows to perceive slight differences in adaptation between individuals. Modern theory of evolution, Neo-Darwinism, performs a synthesis of classic Darwin theory and genetics, considering natural selection for basis of species evolution, in which the key role is played by variations (mutations, also random) of genetic code.

Genetic algorithms are an example of concepts of adaptation and evolutionary mechanism such as inheritance, mutation, natural selection, and recombination (or crossover) to solve complex optimization problems see eg. [Goldberg (1989)], [Michalewicz (1996)], [Coley (1999)] and [Man et al. (1999)], Population of rapidly reproducing creatures/individuals (representing models of optimized objects) is placed in the artificial environment (represented by optimization criteria and constraints), in which adaptation differences are exaggerated in order to speed up the evolution. The size of the population is determined in a way to make it large enough to ensure steady flow of high number of newly occurring mutations. Applied in experimental evolution selective pressure can be abnormally strong – probably much harder than in nature. Individuals with feature variants that are favorable in inhabited environment will survive longer and have more offspring than those with unfavorable features. Therefore favorable features will accumulate over time in process of inevitable "natural selection". That kind of lab procedure simulated with computers allows to quickly obtain variants of optimized objects (in our case ship structures) with desirable features and produce better and better solutions.

Genetic algorithms (GA) have already been extensively described in literature discussing their theoretical foundations, details of calculation procedures and their practical applications, so these problems are not going to be discussed here again and the reader is referred to respective literature e.g. [Goldberg (1989)], [Davis (1991)], [Michalewicz (1996)].

The genetic algorithm is typically implemented in the form of computer simulations where a population of abstract representations (called chromosomes) of candidate solutions (called individuals) to an optimization problem evolves gradually towards better solutions. Traditionally, solutions are represented in the binary system as strings of 0s and 1s but different encodings are also possible. The evolution starts from a population of completely random individuals and is continued in subsequent generations. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), modified (mutated or recombined) to form a new population which becomes current in the next generation. Procedures of creation and evaluation of the successive generations of trial solutions are repeated until the condition of termination of computations is fulfilled, e.g. forming a predefined number of generations or lack of correction of the fitness function in a number of successive generations. The best variant found is then taken as the solution of the optimization problem.

The most important point of calculation tool for multi-objective optimization of ship structure is appropriate formulation of fitness function which governs the optimization process. In the first step a single, parameterized objective function was formulated, consists in summing the objective criteria with proper weight coefficients. The weight coefficients values of this substitute objective function are taken accordingly corresponding to multi-objective optimization strategy. The simplest concept is the introduction of objective function F(x) as a linear combination S of partial objective criteria $f_s(x)$:

$$F(x)=\sum_{s=1}^{n_o} w_s f_s(\mathbf{x}) \qquad (5)$$

objective = criteria

where:
w_s – coefficients determining the weights given to particular criteria,
n_o – the number of partial optimization criteria.

The multi-objective optimization problem is then reduced to the single-objective problems, with the criteria being usually called objective functions.

Formulating of fitness function in the form of scalar substitute optimization criteria in form of Eq. 5 is a commonly accepted practice. Written in Eq. 5 partial optimization criteria were replaced by properly formulated utility functions of these criteria: $f_s(x) \rightarrow u_s[f_s(x)]$:

$$u_i(x) = \left(\frac{f_i(x)}{f_{i,max}}\right) \rightarrow \max! \Leftrightarrow f_i(x) \rightarrow \max! \qquad (6a)$$

$$u_j(x) = \left(\frac{f_{j,max} - f_j(x)}{f_{j,max}}\right) \rightarrow \max! \Leftrightarrow f_j(x) \rightarrow \min! \qquad (6b)$$

where:
$f_{i,max}$ and $f_{j,max}$ – the greatest values of respective optimization criteria anticipated in computations. These values are in practice set during test computations.

After assuming utility function in form of Eq. 6 the scalar substitute optimization criterion can be written down in form:

$$F(x)=\sum_{s=1}^{n_o} w_s u_s(\mathbf{x}) \qquad (7)$$

objective = criteria

Advantages of proposed substitutions are:

1. scalar substitute optimization criterion is maximized for any types of partial optimization criteria, that means F(x) → max!,
2. values of utility function are dimensionless and normalized to a unit, that means $u_s[f_s(x)] \rightarrow [0, 1]$.

The calculation tool developed for use in optimization of ship structure should of course allow for accounting for a series of constraints imposed by design, local strength and overall strength. On the other hand, implementation of genetic algorithms requires that the equivalent problem is formulated without any constraints. Observing that genetic algorithms do not require continuity nor the existence of derivative functions, an external penalty function has been used [Fox (1971)], [Ryan (1974)], [Reklaitis et al. (1983)], [Vanderplaats (1984)]. The augmented objective function of unconstrained maximization problem f(x), has been formulated as a penalty function:

$$f(x)=\sum_{s=1}^{n_o} w_s u_s(\mathbf{x}) + \sum_{k=1}^{n_c} w_k P_k(x) \qquad (8)$$

objective = criteria + constraints

where:
$u_s(x)$ – utility function in the constrained problem (Eq. 7),
n_o – number of optimization criteria,
$P_k(x)$ – component of penalty function for the violation of k-th constraint,
w_k – penalty coefficient for the violation of k-th constraint,
n_c – number of constraints.

To adjustment mathematical form of penalty function $P_k(x)$ with dimensionless and normalized utility functions $u_s(x)$ similar requirements were put before penalty functions. Mathematical form of the penalty function was therefore proposed in form:

$$Pk(x) = \begin{cases} e^{|\mathbf{x}-\mathbf{x}_k|}, & \text{for} \quad \mathbf{x} < \mathbf{x}_k \\ e^{(\mathbf{x}_k - \mathbf{x})}, & \text{for} \quad \mathbf{x} \geq \mathbf{x}_k \end{cases} \qquad (9)$$

where:
$P_k(x)$ – appropriate component of penalty function,
x – vector of current values of design variables,
x_k – vector of design variables ensuring the least permissible value of appropriate constrain condition, (e.g. vector containing the least required by classification rules thickness of hull plate or the least required value of section modulus of stiffener).

Assumed forms of penalty functions components ensure normalization of penalty function components to a unit and introduce „soft" selection to the search, from a point of view of a factor defined by an appropriate component of penalty function $P_k(x)$.

As the augmented objective function f(x) expressed by the relation Eq. 8 with penalty components expressed by Eq. 9 is: (1) defined, (2) single-valued, (3) ascending, having real values and positive in the search space, it has been adopted directly as the fitness function.

As we already know the scheme of multi-objective optimization proposed in Eq. 8 allows only for rough differentiation of feasible solutions with regard to domination relation in Pareto sense, see Fig. 8a, and does not account for information about how many solutions are dominated by a given solution.

For the solving of the mentioned problem the author proposed a scheme in which the feasible solutions are ranked by the number of other solutions dominated by them, relative to the number of feasible solutions in the current population. Therefore, dominance rank R_{fi} of i-th feasible solution is specified by an equation:

$$R_{fi}(i) = \frac{\sum_{j=1, j\neq i}^{N_{fi}} dm(i,j)}{N_{fi}} \qquad (10)$$

where:
dm(i, j) = 1 when i dominates j,
dm(i, j) = 0 in other cases,
i, j – indices of verified feasible solutions,
N_{fi} – number of feasible solutions in the current population.

Fig. 8c and Fig. 10a. The advantages of proposed strategy are: (1) ease of calculations, (2) standardization of dominance rank values in [0, 1] range, and (3) ascending values of dominance rank for solutions approaching the Pareto front (lying at the edge of feasible set). Thanks to properties (2) and (3) the value of dominance rank calculated in a proposed way may be directly included in the fitness function. In such a case selection is going to promote feasible solutions located close to Pareto front, while the solutions lying gradually further and further from the Pareto front are going to be promoted weaker and weaker, which is a numerical realization of selection pressure exerted on solutions located close to Pareto front and which thus enhances the exploitational performance of the algorithm. A disadvantage of the proposed strategy is computational complexity N^2.

Similarly, feasible solutions may be classified by the number of solutions dominating them, relative to the number of feasible solutions. Thus, evaluation dominance count C_{fi} of i-th feasible solution is expressed by the formula:

$$C_{fi}(i) = \frac{\sum_{j=1, j \neq i}^{N_{fi}} dm(j,i)}{N_{fi}} \tag{11}$$

where:
dm(i, j) = 1 when i dominates j,
dm(i, j) = 0 in other cases,
i, j – indices of verified feasible solutions,
N_{fi} – number of feasible solutions in the current population.

Fig. 8d and Fig. 10b. The dominance count defined in this way has the above mentioned properties (1) and (2), and property (3) ascending values of dominance count for the variants lying further and further from Pareto front (located deep inside the feasible set). Thanks to properties (2) and (3) the value of dominance count calculated in the proposed way may also be directly included in the feasible function - in such a case selection is going to promote feasible solutions located far from Pareto front, while the solutions approaching the Pareto front are going to be promoted weaker and weaker, which is a numerical realization of selection pressure exerted on solutions located far from Pareto front and which thus enhances the exploratory properties of the algorithm. Same as in the previous case, the disadvantage of the proposed strategy is computational complexity N^2.

As it has already been mentioned, the strategies for dominance ranking and the dominance count the feasible variants proposed by the author allows for their inclusion directly in the earlier formulated (Eq. 8) extended objective function of a unconstrained maximization problem f(x):

$$f(x) = \sum_{s=1}^{n_o} w_s u_s(\mathbf{x}) + w_r R_{fi}(x) + + w_c C_{fi}(x) + \sum_{k=1}^{n_c} w_k P_k(x) \tag{12}$$

combined objective = criteria + rank + count + constraints

where:
$u_s(x)$ – utility function for the constrained problem (Eq. 7),
n_o – number of optimization criteria,
$R_{fi}(x)$ – dominance rank of feasible variant,
w_r – dominance rank weight coefficient,
$C_{fi}(x)$ – dominance count of feasible variant,
w_c – dominance count weight coefficient,
$P_k(x)$ – penalty function component for the violation of k-th constraint,
w_k – penalty coefficient for the violation of k-th constraint,
n_c – number of constraints. Giving a zero value to the respective weight coefficient lets the user deactivate a given component of combined fitness function.

For indicate a single solution which may be considered to be „the best" solution of a multi-objective optimization problem and the monitoring of evolution of non-dominated solution in the direction of theoretically lowest values of optimization criteria $f_i \rightarrow 0!$ as moving the set of non-dominated solutions in the desired direction the author has then introduced a concept of a asymptotic objective (asymptotic solution), which represents an objective/solution corresponding to asymptotical values of optimization criteria: $f_1 \rightarrow 0!$ $f_2 \rightarrow 0!$[3)], Fig. 12b. Such a definition of a asymptotic solution it is possible to: (1) determine the distance between each non-dominated solution from and this point, and then choose a asymptotic closest solution, and additionally (2) monitor the evolution of set of non-dominated solutions distance from this solution during the simulation.

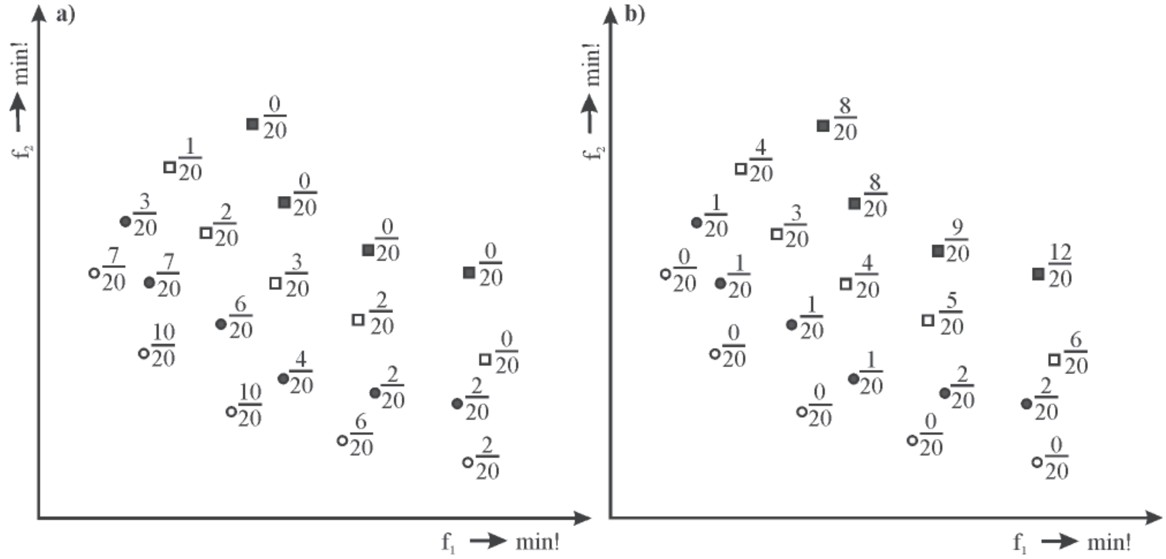


Fig. 10. *Illustration proposed strategy of value calculation of:*
a) *dominance rank,* ***b)*** *dominance count for feasible solutions; $f_1 \rightarrow$ min!, $f_2 \rightarrow$ min!*

[3)] It is then assumed that in case of min! type criteria the respective objectives (co-ordinates) are equal to zero. However, in case of max! type criteria the user shall set the value of these objectives (co-ordinates) as some known from experience values of respective criteria which are impossible to attain, but which are going to be approached asymptotically by the non-dominated solutions.

The method of calculating the distances between feasible solutions and a asymptotic solution allow for including a distance from an asymptotic solution directly in the earlier formulated extended objective function used for unconstrained maximization problem, f(x):

$$f(x) = \sum_{s=1}^{n_o} w_s u_s(\mathbf{x}) + w_r R_{fi}(x) + w_c C_{fi}(x) + \\ + w_d[1 - d_{fi}(x)] + \sum_{k=1}^{n_c} w_k P_k(x) \qquad (13)$$

combined objective = criteria + rank +
+ count + distance + constraints

where:

$u_s(x)$ – objective function for the constrained problem (Eq. 7),
n_o – number of optimization criteria,
$R_{fi}(x)$ – dominance rank of feasible solution,
w_r – dominance rank weight coefficient,
$C_{fi}(x)$ – dominance count of feasible solution,
w_c – weight coefficient of dominance count,
$d_{fi}(x)$ – distance between the feasible solution and the asymptotic solution,
w_d – distance weight coefficient,
$P_k(x)$ – penalty function component for the violation of k-th constraint,
w_k – penalty coefficient for the violation of k-th constraint,
n_c – number of constraints.

The combined fitness function used in the proposed form (Eq. 13), includes instruments which provide for effective solving of constrained multi-objective optimization ship structure problem. These are: combined fitness = objectives + rank + count + distance + constraints; objectives – represents selective pressure exerted in the direction of the desired values of optimization criteria, dominance rank as well as dominance count – represents selective pressure related to the location of an feasible variant with regard to non-dominated solutions set, measured with appropriate dominance attributes, distance – represents selective pressure related to the distance between an feasible solution and the asymptotic solution, constraints – represents the reduction of solution quality and the related reduction of selection probability caused by the violation of constraints. As the values of dominance rank, dominance count and the distance to asymptotic solution are calculated only for feasible variants, this involves also additional promotion of such variants. Proper use of the proposed components of the combined fitness function by the experienced user makes for a highly flexible and effective solving of multi-objective optimization problems based on a genetic algorithm or other evolutionary algorithms in general.

Domination of rank component is used for promoting the variants belonging to non-dominated solutions set which increases their chances for participation in reproduction and transmission of their characteristics to a larger number of descendant specimens, and thus boosts the propensity for exploration of solution space areas which contain good solutions. Variants lying far from non-dominated solutions set have lesser chances for reproduction. Such a strategy is going to reinforce the existing composition of non-dominated solutions set but it is not going to be conducive to supply of new, potentially interesting/attracting/desired solutions. It is then a strategy biased towards exploitation of areas with non-dominated solutions set variants as opposed to the exploration of a whole solution space. Domination of count component is going to have an opposite effect: reduction of reproduction probability for variants belonging to non-dominated solutions set (use of elitist strategies may however dampen this effect) and the amplification of chances for adding new variants to non-dominated solutions set, potentially having the desired characteristics. It is then a strategy biased towards exploration a whole solution space as opposed to the exploitation of areas containing variants of the desired characteristics. The choice of strategy depends on a specific problem at hand and should be preceded with testing calculations. The author proposes to use the domination of dominance count in cases where the number of variants created during a simulation run is a small fraction of a number all variants which may be created within a given optimization model, and the requirements concerning the accuracy obtained in pinpointing the compromise solution are not very high. In such a case intensive exploration of solution space boosts the chances for finding a satisfactory solution. However, in case the user emphasizes the requirement for the accuracy of a located non-dominated solution approximating the Pareto front, rank component domination is going to be recommended.

The proposed combined fitness multi-objective evolutionary algorithm (CFMOEA) using in the selection process a scalar objective function expressed in the Eq. 13, which is a weighted sum of elements representing: (1) optimization criteria, (2) ranking in relation on dominance, (3) dominance count, (4) distance from the asymptotic solution, and (5) the degree of constraint violation, shall be tested in the next parts of the paper using a fast passenger-car ferry structure as an example of a multi-objective optimization.

The computer code used for multi-objective optimization of ship structures with combined fitness function has been built on the basis of a software package for multi-objective optimization while supplementing it with a series of calculation procedures providing for combined fitness multi-objective optimization of the ship hull structure. The most important of them are the procedures for: (1) encoded of genotype and generation of the ship hull structure variants, (2) analysis of constraints, (3) analysis of feasible solutions set with regard to dominance relation, (4) control of domination, including the determination of domination attribute values, dominance rank and dominance count, and the building of non-dominated solutions set approximating the Pareto set, (5) calculations of the distance of feasible solutions to the asymptotic one, and (6) calculation of combined fitness function values, while accounting for the domination attributes and distance to asymptotic solution of feasible solutions. Block-type diagram presenting the main concepts of developed calculation software is shown in Fig. 11.

The code carries out the calculations automatically, starting from an input data set prepared by the user, which the following data regard to multi-objective optimization strategies: (1) values of switches specifying the selection of strategy which is going to be used for the calculation of fitness function values basing on optimization criteria values, and (2) the values of weight coefficients assigned to combined fitness function components. In the computer code a population of individuals of a fixed size is randomly generated. Each individual is characterized by a string of bits and represents one possible solution to the ship structure. Each new created variant of solution (an individual being a candidate to the progeny generation) is analyzed by the pre-processor. In the pre-processor binary strings of chromosomes (genotypes) are decoded into the corresponding strings of decimal values representing design variables (phenotypes). Then for the actual values of the design variables defining spatial layout of the structural elements (topology) and

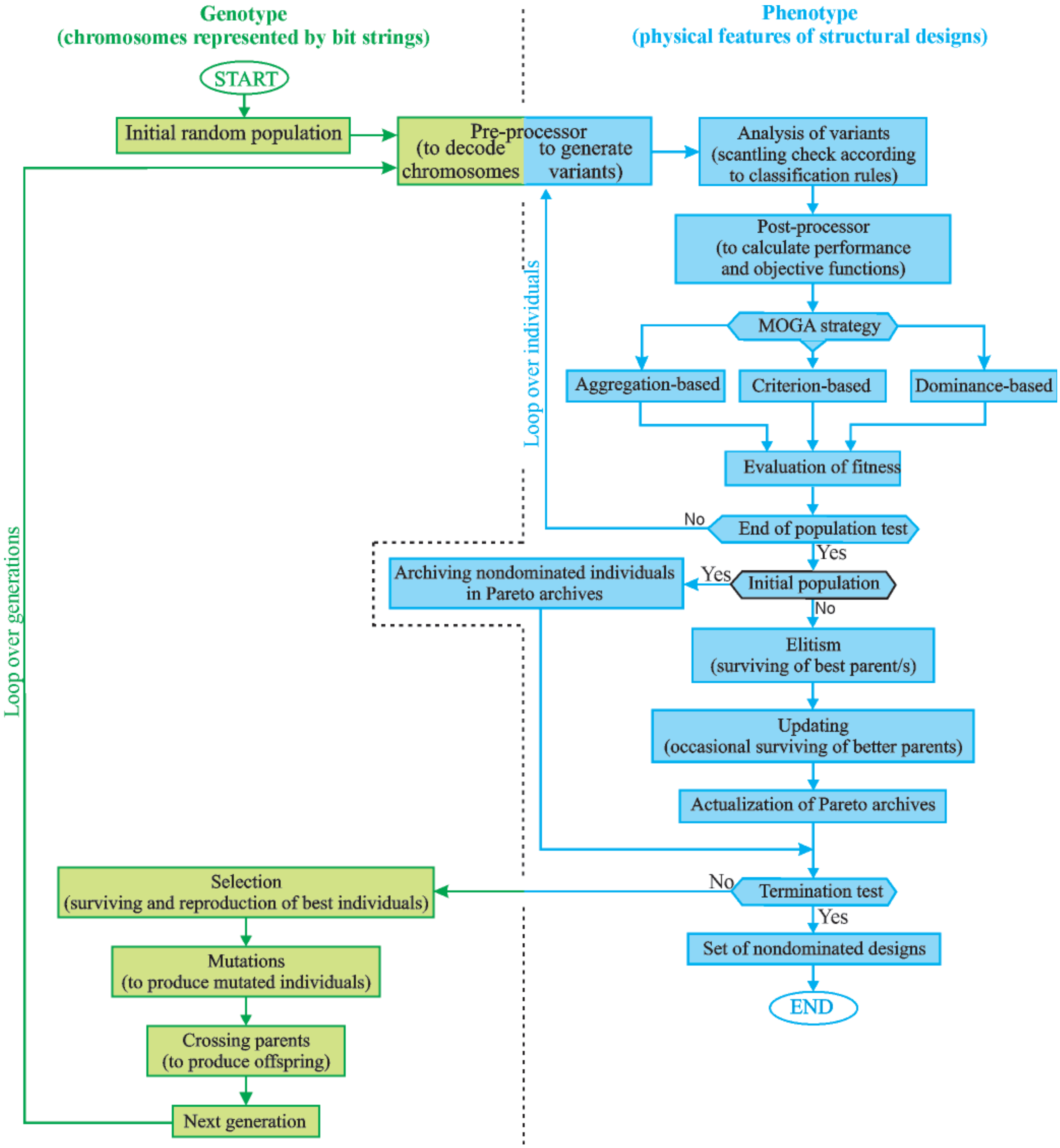

Genotype
(chromosomes represented by bit strings)
Phenotype
(physical features of structural designs)
START
Initial random population
Pre-processor
(to decode chromosomes
to generate variants)
Analysis of variants
(scantling check according to classification rules)
Post-processor
(to calculate performance and objective functions)
MOGA strategy
Aggregation-based
Criterion-based
Dominance-based
Evaluation of fitness
No
End of population test
Yes
Archiving nondominated individuals in Pareto archives
Yes
Initial population
No
Elitism
(surviving of best parent/s)
Updating
(occasional surviving of better parents)
Actualization of Pareto archives
No
Termination test
Yes
Set of nondominated designs
END
Loop over individuals
Loop over generations
Selection
(surviving and reproduction of best individuals)
Mutations
(to produce mutated individuals)
Crossing parents
(to produce offspring)
Next generation

Fig. 11. Diagram presenting the main concepts of the developed computer software for the multi-objective optimization of seagoing ship structures, based on a CFMOGA

their scantlings it is checked whether the actual configuration complies with the rules of the classification society. In the next step performance of solution is evaluated and it is checked whether the variant meets the constraints.

The domination relation is checked over a set of individuals considered to be feasible . The set of non-dominated individuals, updated for each generation, constitutes an approximation of Pareto front. At the end, the value of fitness function, proportional to the individual selection probability and its participation in the genetic operations of the next generation, is calculated. Calculation of combined fitness function value takes into account the of multi-objective optimization strategy, selected by the user. The computer software developed herein implements three selection strategies, Fig. 6: (1) strategy based on the aggregation of vector objective function, with appropriate weight coefficients (fixed values for (1a) strategy or random values for (1b) strategy), (2) strategy executing the selection of variants on the basis of randomly selected single optimization criteria, and (3) strategy based on selection with respect to attributes of Pareto domination.

The code analyses the feasible individuals of successive generations, selects the non-dominated solutions and this way builds the archive of non-dominated solutions (Pareto-optimal ones). The archive of non-dominated variants is an external set and the variants which are recorded in it do not take part in the „breeding" of successive generation solutions, unless they have managed to survive in successive populations in a natural way, remaining not destroyed and not excluded by genetic operators.

Process of building and evaluation of descendant populations of tested variants and the updating of non-dominated variants archive is repeated until a fixed number of generations used as a simulation halt condition is reached. After a simulation is completed, the non-dominated solutions preserved in the archive are recognized to be the sought solutions of a multi-objective optimization problem.

At the end the value of the fitness function is calculated which is used for ordering the variants necessary to starting of selection. Variants are ordered with respect to this value. Knowing adaptation of each variant the random process is restarted to select variants of the successive progeny generation.

After selection the code determines randomly which genes of these whole population will mutate. This population is then mutated where small random changes are made to the mutants to maintain diversity. After that the mutate pool is created. Then decision is made how much information is swapped between the different population members. The mutated individuals are then paired up randomly and mated in the process commonly known as crossover. The idea is to derive better qualities from the parents to have even better offspring qualities. That is done by creating, with fixed probability, „cutting points" and then the parts of the chromosomes located between "cuts" are interchanged. The mating process is continued until the full population is generated. The resulting population member is then referred to as an offspring. The newly generated individuals are then re-evaluated and given fitness score, and the process is repeated until it is stopped after a fixed number of generations.

All genetic parameters are specified by the user before the calculations. The population size, number of design variables and number of bits per variable, the total genome length, number of individuals in the population are limited by the available computer memory.

The code carries out the calculations automatically, starting from an input data set prepared by the user, which has been supplemented with the following data with regard to its aggregated multi-objective predecessor: (1) values of switches specifying the selection of strategy which is going to be used for the calculation of fitness function values basing on optimization criteria values, and (2) the values of weight coefficients assigned to combined fitness function components and depending on dominance attributes and distance to asymptotic solution. The domination relations allowing for the determination of dominance rank and dominance count values for particular feasible variants are checked over a set of variants considered to be feasible. The set of non-dominated variants, updated for each generation, constitutes an approximation of the Pareto-optimal front. At the end, the value of fitness function, proportional to the variant selection probability and its participation in the genetic operations of the next generation, is calculated. Calculation of fitness function value takes into account the strategy of combined fitness multi-objective evolutionary algorithm, selected by the user. The computer software developed herein implements several multi-objective genetic algorithm selection strategies, Fig. 6. First of all, optimization is possible only with regard to optimization criteria: (1) strategy based on the aggregation of vector objective function, with appropriate weight coefficients, (2) strategy executing the selection of variants on the basis of randomly selected single optimization criteria. By assuming different from zero values of weight coefficients it is possible in optimization process to also take into consideration domination attributes as well as distance to asymptotic solution.

The values of parameters controlling the genetic operators are specified by the user before the start of simulation. Proper setting of these values is very important and requires extensive experience on the part of the user, but it is crucial for the attainment of the desired calculations convergence expressed in (1) quality of found solutions, (2) rapidity of their finding, and (3) required computation resources.

In case of multi-objective optimization problem we have to consider how to collect and present the information about the determined non-dominated solutions (Pareto-optimal) and how to archive them. It is commonly accepted to graphically present all the feasible solutions as points in the objective space. Only a part of them are going to be non-dominated (dominating, Pareto-optimal) solutions, and their set is going to be called a non-dominated set or a Pareto front containing trade-off solutions (actually we know this is going to be a set of non-dominated solutions approximating the Pareto set).

Non-dominated solutions produced during the simulation are recorded in a separate set (file) which is continuously supplemented and updated during the simulation. The solutions collected in the non-dominated solutions set may be dealt with in two possible ways: (1) set membership has no influence on the selection of individuals (egalitarian strategy) (2) individuals from the set (non-dominated solutions) enjoy a guaranteed participation in selection (elite strategy). Egalitarian strategy has been adopted in the underlying paper.

It is a well-known fact that in case many optimization criteria are used, it is going to be impossible to find a single best solution, as such a solution does not exist. In practice however the user awaits automatic or quasi-automatic determination of a single solution or a few solutions, which could be taken as a solution of the problem. Moreover, users are accustomed to the monitoring of evolution of a single value, which lets them evaluate the correctness of the calculation run, the convergence of solutions and the quality of solutions being found. In single-objective cases it is natural to monitor the values of fitness function and optimization criterion. In case of multi-objective optimization simultaneous evaluation of the evolving criteria is difficult to realization and interpretation. In order to alleviate this problem, the author has used the concept of ideal or utopia solution, well-known in literature [Cohon (1978)], [Stadler (1988)], [Statnikov and Matosov (1995)], see Fig. 12. In the generally accepted understanding an ideal point refers to the lowest values of all criteria analyzed singly and not together. It means that if f^*_1, f^*_2, … f^*_s, …, f^*_S will be used to denote the individual minima of each respective objective function, and the ideal solution (in objective space) is defined as $f^* = [f^*_1\ f^*_2\ \dots\ f^*_s\ \dots\ f^*_S]^T$. As f^* simultaneously minimizes all objectives, it is an ideal solution that is rarely feasible. In such a case however it is then possible to locate a solution closest to the ideal point (nearest to the ideal solution):

$$f^{\approx}_{m}(x) = [f^{\approx}_{1;m}(x)\ \ f^{\approx}_{2;m}(x)]^{T} \tag{14}$$

with the concept of closeness being understood here usually in the sense of Euclidean metrics where $f^{\approx}_m(x)$ is the closest solution found in the m generation.

The concept of non-dominated feasible solution nearest to the ideal objective $f^{\approx}(x)$ is sufficient to find a single solution which may be considered to be „the best" solution of a multi-objective optimization problem. It is however inappropriate for the monitoring of evolution of non-dominated solution in the direction of theoretically lowest values of optimization criteria $f_i \to 0$! as moving the set of non-dominated solutions in the desired direction may also take place with unchanging distances of these solutions from ideal solution, Fig. 13. The author has

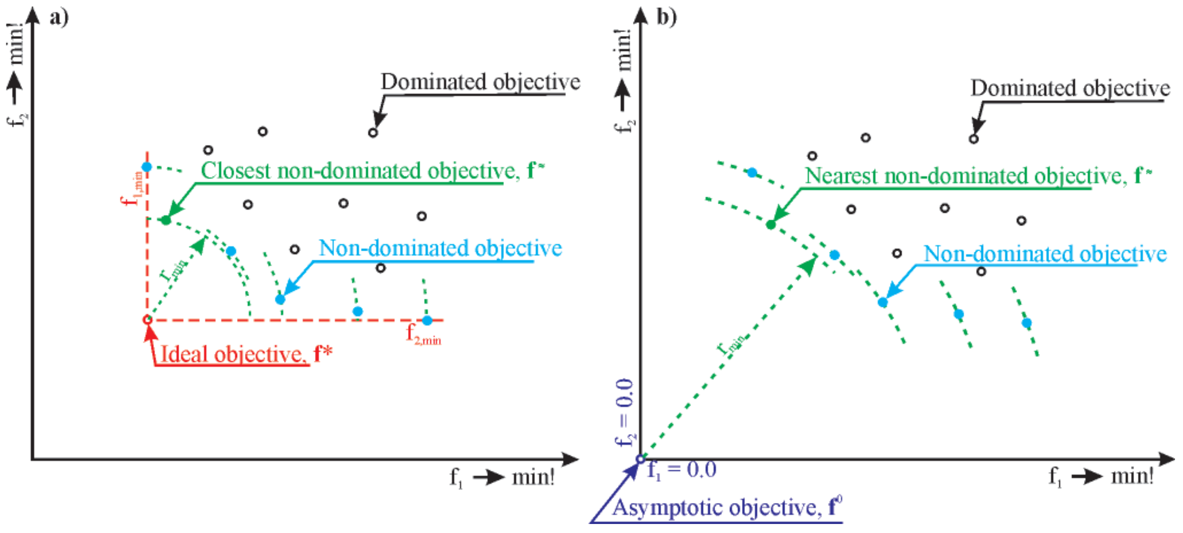


Fig. 12. a) *Graphical illustration of a non-dominated solution nearest to the ideal one f^*, for the case of min! type criteria.*
b) *Graphical illustration of the concept of a non-dominated feasible solution nearest to a asymptotic objective f^0, in case of min! type criteria*

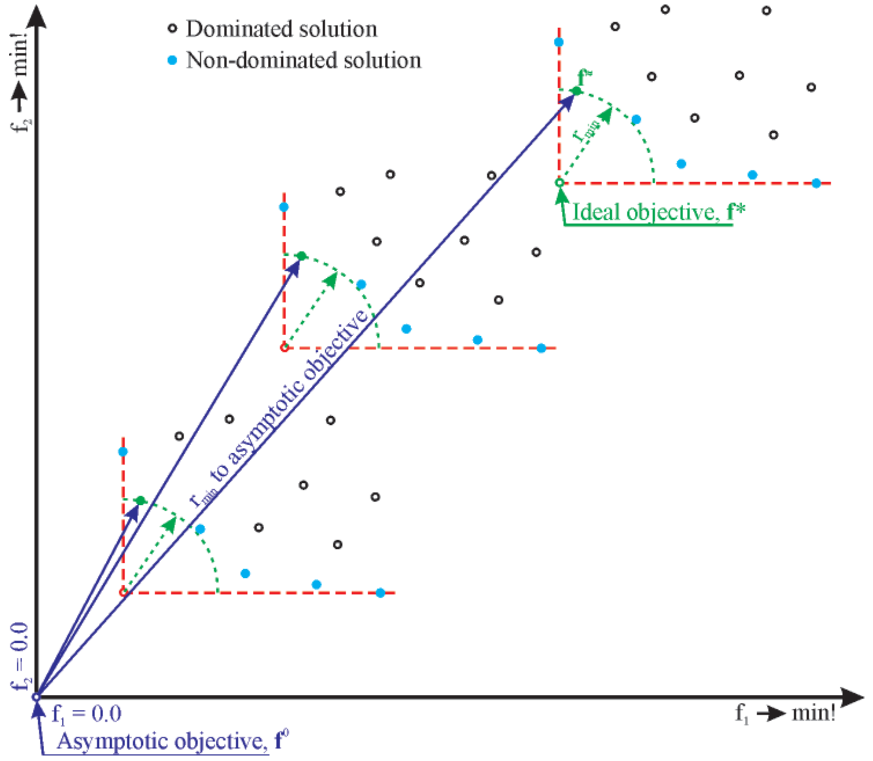


Fig. 13. Graphical illustration of the movement of solutions nearest to the ideal solution $f^{\approx}$ in the direction of asymptotic solution during the evolution, in case of min! type criteria; for constant distances from the ideal solution the distance from asymptotic one changes (evolves) instead

then introduced a concept of a asymptotic objective (asymptotic solution) f^0, which represents an objective corresponding to asymptotical values of optimization criteria: $f_1 \rightarrow 0$! $f_2 \rightarrow 0$!, Fig. 13. According to this definition, the location of ideal objective f* in the objective space is not fixed, Fig. 13, and this means that its location changes during simulation, so the distance to such a moving objective is not a good indicator of solution quality. Therefore, a definition of asymptotic objective f^0 has been adopted in such a way as to make it stationary. It is then necessary to have such a definition which would make the co-ordinates of this objective constant in the objective space, which means fixed values of optimization criteria. It is then assumed that in case of min! type criteria the respective objectives (co-ordinates) are equal to zero. However, in case of max! type criteria the user shall set the value of these objectives (co-ordinates) as some known from experience values of respective criteria which are impossible to attain, but which are going to be approached asymptotically by the non-dominated solutions. After defining the asymptotic objective f^0 let's assume that as the nearest solution $f^{\approx}(x)$ we mean nearest solution with regard to the asymptotic objective f^0.

REFERENCES

The complete list of references will be insert in the end of third part of the paper.

CONTACT WITH THE AUTHOR
Zbigniew Sekulski, Ph. D.
West Pomeranian University of Technology, Szczecin
Faculty of Marine Technology
Al. Piastów 41
71-065 Szczecin, POLAND
e-mail: zbigniew.sekulski@zut.edu.pl