HAL
open science

A New Method for Decision Making in Multi-Objective Optimization Problems

Oscar Augusto, Bennis Fouad, Stéphane Caro

To cite this version:

Oscar Augusto, Bennis Fouad, Stéphane Caro. A New Method for Decision Making in Multi-Objective Optimization Problems. Pesquisa Operacional, 2012, 32 (3), pp.331-369. 10.1590/S0101-74382012005000014 . hal-00914025

HAL Id: hal-00914025

https://hal.science/hal-00914025v1

Submitted on 4 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pesquisa Operacional
© 2012 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope

A NEW METHOD FOR DECISION MAKING IN MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Oscar Brito Augusto[1*], Fouad Bennis[2] and Stephane Caro[3]

Received June 21, 2010 / Accepted March 12, 2012

ABSTRACT. Many engineering sectors are challenged by multi-objective optimization problems. Even if the idea behind these problems is simple and well established, the implementation of any procedure to solve them is not a trivial task. The use of evolutionary algorithms to find candidate solutions is widespread. Usually they supply a discrete picture of the non-dominated solutions, a Pareto set. Although it is very interesting to know the non-dominated solutions, an additional criterion is needed to select one solution to be deployed. To better support the design process, this paper presents a new method of solving non-linear multi-objective optimization problems by adding a control function that will guide the optimization process over the Pareto set that does not need to be found explicitly. The proposed methodology differs from the classical methods that combine the objective functions in a single scale, and is based on a unique run of non-linear single-objective optimizers.

Keywords: multi-objective optimization, multi-attribute decision making, engineering design optimization.

1 INTRODUCTION

Many engineering design problems are multi-objective in nature as they often involve more than one design goal to be optimized. These design goals potentially impose conflicting requirements on the technical and economic performance of a given system design. To study the trade-offs between these conflicting design objectives and to explore design options, an optimization problem with multiple objectives has to be formulated. To facilitate the exploration of different multi-objective formulations, we can incorporate Pareto optimality concepts into optimization algorithms that require the designer's involvement as a decision maker.

Edgeworth (1881) was the first academic to define an optimization problem involving multiple criteria. The problem, elaborated in the context of two consumers called P and π can be defined

*Corresponding author
[1]Escola Politécnica da Universidade de São Paulo. E-mail: oscar.augusto@poli.usp.br
[2]École Centrale de Nantes, Institut de Recherche en Communications et Cybernétique de Nantes.
E-mail: fouad.bennis@ec-nantes.fr
[3]Institut de Recherche en Communications et Cybernétique de Nantes. E-mail: stephane.caro@irccyn.ec-nantes.fr

as: "*it is required to find a point* (x, y) *such that, in whatever direction we take an infinitely step,* P *and* π *do not increase together, but, while one increases, the other decreases*".

A few years later, in 1896, Pareto (1971) establishes the optimum for n consumers: "*we will say that members of a collectivity enjoy maximum ophelimity in a certain position when it is impossible to find a way of moving from that position very slightly in such a manner that the ophelimity enjoyed by each of the individuals of that collectivity increases or decreases. That is to say, any small displacement in departing from that position necessarily has the effect of increasing the ophelimity which certain individuals enjoy, and decreasing that which other enjoy, being agreeable to some and disagreeable to others*".

Since then, many researchers have been dedicated to developing methods to solve this kind of problem. Interestingly, solutions for problems with multiple objectives, also called multi-criteria optimization or vector optimization, are treated as Pareto optimal solutions or Pareto front, although as Stadler observed (1988), they should be treated as Edgeworth-Pareto solutions.

On the other hand, thanks to the evolution of computers, the optimization of large scale problems has become a common task in engineering design processes. Computers, ever faster, with higher capacity of data storage and Internet connection capability, are revolutionizing the traditional ways of developing engineering designs. Currently, engineers can rely on a wide range of design alternatives and optimization methods that allow systematic choices between alternatives when they are based on measurable criteria. Used properly, these methods can improve or even generate the final solution of a design, as shown in Giassi *et al.* (2004), which held multi-objective optimization of a vessel hull shape with collaborative design method where the computational tools for the analysis of alternatives used in the cycles of optimization were distributed in three different sites. With this approach they demonstrate that if the optimization is conducted in a distributed way it becomes a powerful tool for the concurrent design.

Rao *et al.* (2007) designed a disk for high-pressure turbine aircraft engines, treating the problem as multi-objective by considering the manufacturing cost and fatigue life as functions to be optimized. Bouyer *et al.* (2007) performed the optimal design of a mechanical transmission system, *Slide-O-Cam*, which turns rotary into translation motion and replaces the conventional rack-pinion systems, a problem with four objective functions. In both instances, the optimal solutions were presented by their Pareto fronts.

Plenty of methods for solving design problems with more than one objective are found in scientific and technical literature. However, since the multi-objective problems have no single solution but a set of points called optimal solutions, the question that arises is *how to choose one alternative among those contained in the set of solutions*. The answer is not simple unless another criterion is added to help choose the alternative that should be deployed.

In this paper we propose a different alternative to solve the problem of optimizing a set of objective functions that will be divided into two groups: one containing only one function that will be called *control function* and one in which the p functions will be called the *performance functions group*. With this strategy, the optimization problem is formulated as single-objective, optimizing

the control function over the Pareto set that would result from the optimization problem constituted by the group of performance functions if this multi-objective optimization problem had been solved. However, since this task is not performed, the Pareto front is not an output of the optimization process and the computational task is significantly smaller than the traditional multi-objective optimization algorithms.

Applying this strategy to a multi-objective optimization problem requires the use of an algorithm that can solve a single-objective problem and work with the functions involved. The result will be a single solution. However, the methodology differs significantly from other classical algorithms as there is no need to know a value function relating the objectives that articulates de designer preferences.

After this brief introduction, the remaining of the paper is organized into six sections. The first formulates a general multi-objective optimization problem and defines the nature of optimal solutions from the Pareto perspective and the necessary conditions to be met. The second presents a brief literature review on multi-objective optimization problem algorithms. In the sequence, it presents the main contribution of this work, a new strategy to solve the optimization problem with multiple objectives. Next, it goes on to the implementation of the strategy to solve three problems with increasing levels of complexity, and finally presents the conclusions on the work done.

Nomenclature

AIS	artificial immune systems
AC	ant colony optimization algorithm
DM	decision maker
$f_i(\mathbf{X})$	ith objective function
$f_c(\mathbf{X})$	control function
$f_i^o(\mathbf{X})$	minimum value of the ith objective function
$f_i^{\max}(\mathbf{X})$	maximum value of the ith objective function
tol_f	tolerance for equal values of *rms* in successive generations
$\mathbf{f}(\mathbf{X})$	objective functions vector
$\mathbf{f}_p(\mathbf{X})$	performance functions vector
$\mathbf{f}^o$	ideal point, vector with all objective functions' minima, in criterion space
GA	genetic algorithm
$g_j(\mathbf{X})$	jth inequality constraint function
$h_i(\mathbf{X})$	ith equality constraint function
k	number of objective functions
KKT	Karush-Khun-Tucker
l	number of equality constraint functions
m	number of inequality constraint functions
MOOP	multi-objective optimization problem
NSGA II	non-dominated sorting genetic algorithm, version two

n	dimension of the design space
p	number of performance functions
PS	particle swarm optimization algorithm
rms_n	root mean square of all objective functions at the nth generation
$\mathbb{R}^k$	function or criterion space
$\mathbb{R}^n$	decision variables or design space
SA	simulating annealing optimization algorithm
$\mathbb{S}$	feasible region in the design space
VEGA	vector evaluated genetic algorithm
x_i	ith decision variable
$\overline{x_i}$	average of the ith decision variable included in the optimal set
$\mathbf{X}$	decision variable vector
$\mathbf{X}^*$	non-dominated solution of a multi-objective optimization problem
$\mathbf{X}_{\text{inf}}$, $\mathbf{X}_{\text{sup}}$	lower and upper bounds of the design space
$\mathbf{X}_{extended}$	extended vector for unknowns with $\boldsymbol{X}$, $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$
α_i	weighting factor for ith objective function gradient in KKT condition
$\boldsymbol{\alpha}$	vector of α_{is}
ω_i	weighting factor for the ith objective function
ω_i^*	normalizing weighting factor for the ith objective function
ε_j	upper bound for the jth objective function in the ε-restricted method
λ_j	weighting factor for jth inequality constraint gradient in KKT condition
$\boldsymbol{\lambda}$	vector of λ_{js}
σ_i	standard deviation for the ith decision variable included in the optimal set
μ_i	weighting factor for ith equality constraint gradient in KKT condition
$\boldsymbol{\mu}$	vector of μ_{is}
$\boldsymbol{\nabla}$	gradient operator

2 MULTI-OBJECTIVE OPTIMIZATION PROBLEM

Multi-objective optimization problems (MOOP) can be defined by the following equations:

$$\text{minimize:} \quad \mathbf{f}(\mathbf{X}) \tag{1a}$$

$$\text{subject to:} \quad g_i(\mathbf{X}) \leq 0, \quad i = 1, 2, \ldots m. \tag{1b}$$

$$h_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, l. \tag{1c}$$

$$\mathbf{X}_{\text{inf}} \leq \mathbf{X} \leq \mathbf{X}_{\text{sup}} \tag{1d}$$

where

$$\mathbf{f}(\mathbf{X}) = \left[f_1, f_2, f_3, \ldots, f_k\right]^T : \mathbf{X} \to \mathbb{R}^k$$

is a vector with the values of objective functions to be minimized. $\mathbf{X}$ is the vector containing the design variables, also called decision variables, defined in the design space $\mathbb{R}^n$. $\mathbf{X}_{\text{inf}}$ and

$\mathbf{X}_{\text{sup}}$ are respectively the lower and upper bounds of the design variables. $g_i(\mathbf{X})$ represents the ith inequality constraint function and $h_j(\mathbf{X})$ the jth equality constraint function. The three equations (1b)-(1d), define the region of feasible solutions, $\mathbb{S}$, in the design space $\mathbb{R}^n$. The constraints $g_i(\mathbf{X})$ are of type "*less than or equal*" functions in view of the fact that "*greater or equal*" functions may be converted to the first type if they are multiplied by -1. Similarly, the problem is the "*minimization*" of the functions $f_i(\mathbf{X})$, given that functions "*maximization*" can be transformed into the former by multiplying them by -1.

2.1 Pareto optimal

The notion of "optimum" in solving multi-objective optimization problems is known as "Pareto optimal." A solution is said to be Pareto optimal if there is no way to improve one objective without worsening at least one other, *i.e.*, the feasible point $\mathbf{X}^* \in \mathbb{S}$ is Pareto optimal if there is no other feasible point $\mathbf{X} \in \mathbb{S}$ such that $\forall i, j$ with $i \neq j$, $f_i(\mathbf{X}) = f_i(\mathbf{X}^*)$ with strict inequality in at least one condition, $f_j(\mathbf{X}) < f_j(\mathbf{X}^*)$.

Due to the conflicting nature of the objective functions, the Pareto optimal solutions are usually scattered in the region $\mathbb{S}$, a consequence of not being able to minimize all the objective functions simultaneously. In solving the optimization problem we obtain the Pareto set or the Pareto optimal solutions defined in the design space, and the Pareto front, an image of the objective functions, in the criterion space, calculated over the set of optimal solutions.

2.2 Necessary condition for Pareto optimality

In fact, optimizing multi-objective problems expressed by equations Eqs. (1a)-(1d) is of general character. The equations represent the problem of single-objective optimization when $k = 1$. According to Miettinen (1998), such as in single-objective optimization problems, the solution $\mathbf{X}^* \in \mathbb{S}$ for the Pareto optimality must satisfy the Karush-Kuhn-Tucker (KKT) condition, expressed as:

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{X}^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(\mathbf{X}^*) + \sum_{i=1}^{l} \mu_i \nabla h_i(\mathbf{X}^*) = 0 \tag{2a}$$

$$\lambda_j g_j(\mathbf{X}^*) = 0 \tag{2b}$$

$$\lambda_j \geq 0 \tag{2c}$$

$$\mu_i \geq 0 \tag{2d}$$

$$\alpha_i \geq 0; \quad \sum_{i=1}^{k} \alpha_i = 1 \tag{2e}$$

where α_i is the weighting factor for the gradient of the ith objective function, calculated at the point $\mathbf{X}^*$, $\nabla f_i(\mathbf{X}^*)$. λ_j represents the weighting factor for the gradient of the jth inequality constraint function, $\nabla g_j(\mathbf{X}^*)$, and is zero when the constraint function associated is not active,

i.e., $g_j(\mathbf{X}^*) > 0$. μ_i represents the weighting factor for the gradient of the ith equality constraint function, $\nabla h_i(\mathbf{X}^*)$.

The set of Eqs. (2a) to (2e) form the necessary conditions for $\mathbf{X}^*$ to be a Pareto optimal. As described by Miettinen (1998), it is sufficient for the complete mapping of the Pareto front if the problem is convex and the objective functions are continuously differentiable in the $\mathbb{S}$ space. Otherwise, the solution will depend on additional conditions, as shown by Marler and Aurora (2004).

3 LITERATURE REVIEW IN METHODS FOR SOLVING MOOP

Some researchers have attempted to classify methods for solving MOOP according various considerations. Hwang & Masud (1979) and later Miettinen (1998) suggested the following four classes, depending on how the decision maker (DM) articulates preferences: no-preference methods, *a priori* methods, *a posteriori* methods, and interactive methods.

In no-preference articulation methods, the preferences of the DM are not taken into consideration. The problem can be solved by a simple method and the solution obtained is presented to the DM which will accept or reject it.

In *a priori* preference articulation methods, the hopes and opinions of the DM are taken into consideration before the solution process. Those methods require that the DM knows beforehand the priority of each objective transforming the multi-objective problem in a single-objective problem where the function to be optimized is a combination of objective functions.

In posteriori preference articulation methods no preferences of the DM are considered. After the Pareto set has been generated, the DM chooses a solution from this set of alternatives.

In interactive preference articulation methods the DM preferences are continuously used during the search process and are adjusted as the search continues.

Before explore the basics of some MOOP algorithms, it is opportune to define the ideal point

$$\mathbf{f^0} = \left[f_1^o, f_2^o, f_3^o, \ldots, f_k^o\right]^T \in \mathbb{R}^k$$

being f_i^o the minimum of $f_i(\mathbf{X})$, $\mathbf{X} \in \mathbb{S}$ and $i = 1, 2, \ldots, k$. In general, $\mathbf{f^0}$ is unattainable, *i.e.*, such a point in the criterion space does not map to a point in the design space.

In many cases it is advantageous to transform the original objective functions. This is especially true with scalarization methods that involve *a priori* articulation of preferences. As the objective functions usually have different scales, it is necessary to transform them in a non-dimensional. The relation

$$f_i^s(\mathbf{X}) = \frac{f_i(\mathbf{X})}{f_i^o} \tag{3}$$

can be used provided that $f_i^o \neq 0$. Such transformation results in a non-dimensional objective function with a lower limit of one and unbounded upper limit.

Alternatively, a most robust approach would be

$$f_i^s(\mathbf{X}) = \frac{f_i(\mathbf{X}) - f_i^o}{f_i^{\max} - f_i^0} \tag{4}$$

with $f_i^{\max}$, the maximum value of $f_i(\mathbf{X})$ for $\mathbf{X} \in \mathbb{S}$. This operation is referred as normalization as the transformed function is bounded into the zero-one interval.

3.1 The weighted sum method

One of the most intuitive ways used to obtain a single unique solution for multi-objective optimization is the weighted sum method. In this approach, the MOOP are converted into a scalar preference function using a linear weighted sum function of the form,

$$\text{minimize:} \quad \sum_{i=1}^{k} \omega_i f_i^s(\mathbf{X}) \tag{5a}$$

$$\text{subject to:} \quad \mathbf{X} \in \mathbb{S} \tag{5b}$$

$$\omega_i \geq 0; \quad \sum_{i=1}^{k} \omega_i = 1 \tag{5c}$$

For this method, the weights ω_i reflect, *a priori*, the designer's preferences. It is simple, but the proper selection of the weights may be a challenge itself.

The method can be used to find the Pareto set in many individual runs if the weights are consistently chosen for each run. Nevertheless, varying the weights continuously may not necessarily result in an even distribution of Pareto optimal points and a complete representation of the Pareto optimal set for non-convex problems as shown in Miettinen (1998).

3.2 The weighted metric methods

Other means of combining multiple objectives into a single-objective are based on the weighted distance metrics. Some DMs aim to find a feasible design that minimizes its distance from a pre-defined design as the representation of a designer's overall preferences. Assuming that the pre-defined design is the ideal design, the weighted p-norm as is expressed by

$$l_p = \left(\sum_{i=1}^{k} \left[\omega_i^* | f_i - f_i^o | \right]^p \right)^{\frac{1}{p}} \tag{6}$$

where,

$$\omega_i^* = \frac{\omega_i}{f_i^{\max} - f_i^o}$$

is the normalizing weight.

The MOOP is written as

$$\text{minimize:} \quad l_p \tag{7a}$$

$$\text{subject to:} \quad \mathbf{X} \in \mathbb{S} \tag{7b}$$

$$\omega_i \geq 0; \quad \sum_{i=1}^{k} \omega_i = 1 \tag{7c}$$

The parameter p may be chosen from 1 to infinite. With $p = 2$, Eq. (6) yields to the Euclidian distance metric.

A *compromise solution*, an alternative to the idea of Pareto optimality is a single point that minimizes the Euclidian distance between the potential optimal point and the ideal point.

3.2.1 The min-max method

The min-max solution was initially proposed by Lightner and Director (1981). The method try to find a feasible design that minimizes its distance from the ideal design as the representation of a designer's overall preferences. The weighted ∞-norm is used as distance metric.

In Eq. (6), the limit of l_p as p approaches to ∞ is

$$l_\infty = l_{p\to\infty} \xrightarrow{\text{yields}} \max_i \left[\omega_i^* |f_i - f_i^o|\right] \tag{8}$$

because the largest $\omega_i^*(f_i - f_i^o)$ will dominate all others when taken to the infinite power.

The min-max problem is then defined by

$$\text{minimize:} \quad \max_i \left[\omega_i^* |f_i - f_i^o|\right] \tag{9a}$$

$$\text{subject to:} \quad \mathbf{X} \in \mathbb{S} \tag{9b}$$

Given different relative weights of objectives, the min-max method is capable of discovering efficient solutions of a multi-objective problem whether the problem is convex or non-convex.

3.3 The goal programming

Goal programming, originally proposed by Charnes & Cooper (1977), is a technique which requires preference information before any efficient solution are generated. In fact, the method requires a designer to set goals, b_i for all objectives, f_i, that he wishes to achieve. It adopts the decision rule that the best compromise design should be the one which minimizes the weighted sum of deviations from the set goals, $\sum_{i=1}^{k} \omega_i^* |d_i|$, where d_i is the deviation from the goal b_i for the ith objective. To model the absolute values, d_i is split into positive and negative parts such that $d_i = d_i^+ + d_i^-$, with $d_i^+ \geq 0$, $d_i^- \geq 0$ and $d_i^+ d_i^- = 0$. Deviations d_i^- and d_i^+ denote underachievement and overachievement, respectively, where achievement implies that a goal has

been reached. Similarly, the jth-inequality or equality constraint function are treated like the objective functions setting the goals $b_j = 0$. Then, the MOOP is formulated as:

$$\text{minimize:} \quad \sum_{i=1}^{k} \omega_i^* (d_i^+ + d_i^-) \tag{10a}$$

$$\text{subject to:} \quad f_i(\mathbf{X}) + d_i^+ + d_i^- = b_i, \quad i = 1, 2, \ldots, k \tag{10b}$$

$$d_i^+ \geq 0, \; d_i^- \geq 0 \text{ and } d_i^+ d_i^- = 0, \quad i = 1, 2, \ldots, k \tag{10c}$$

$$\omega_i \geq 0 \tag{10d}$$

$$\mathbf{X} \in \mathbb{S} \tag{10e}$$

The method allows the designer to assign preemptive weights to objectives and to define different achievement levels of the goals.

3.4 The ε-constraint method

Haimes *et al.* (1971) introduced the ε-constraint strategy that minimizes the single objective function $f_i(\mathbf{X})$. All other objective functions are used to form additional constraints. The MOOP is formulated as:

$$\text{minimize:} \quad f_i(\mathbf{X}) \tag{11a}$$

$$\text{subject to:} \quad f_j(\mathbf{X}) \leq \varepsilon_j, \quad j = 1, 2, \ldots, k; \quad \forall j \neq i \tag{11b}$$

$$\mathbf{X} \in \mathbb{S} \tag{11c}$$

The definition of the limits ε_j requires knowing *a priori* the designer's preference. A set of Pareto optimal solutions can be obtained with a systematic variation of ε_j. However, improper selection of $\varepsilon_j \in \mathbb{R}$ can result in a formulation with no feasible solution.

3.5 Nature inspired metaheuristic algorithms

The methods for multi-objective optimization presented thus far have involved unique formulations that are solved using standard optimization engines or single-objective optimization methods algorithms. With those methods, only one Pareto optimal solution can be expected to be found in one simulation run of a classical algorithm and not all Pareto optimal solution can be found by some algorithms in non-convex MOOP.

However, other approaches such some heuristics inspired in nature process can solve MOOP getting the Pareto set directly.

3.5.1 Genetic algorithms

Genetic algorithm (GA) is a metaheuristic inspired by the Darwin evolutionist theory explaining the origin of species. In nature, weak and unfit species within their environment are faced with

extinction by natural selection. The strong ones have greater opportunity to pass their genes to future generations via reproduction. In the long run, species carrying the correct combination in their genes become dominant in their population. Sometimes, during the slow process of evolution, random changes may occur in genes. If these changes provide additional advantages in the challenge for survival, new species evolve from the old ones. Unsuccessful changes are eliminated by natural selection.

The concept of GA was originally proposed by Holland (1974) for applications into the control theories and it was quickly generalized to many different areas of engineering and sciences. The specific mechanics of the algorithm involve the language of microbiology and, in developing new potential solutions, mimic genetic operations. A *population* represents a group of potential solution points. A *generation* represents an algorithmic iteration. A *chromosome* is comparable to a design point, and a *gene* is comparable to a component of the design vector. Given a population of designs, three basic operations are applied: *selection*, *crossover*, and *mutation*. The selection operator involves selecting design vectors, called *parents*, in the current generation, which are combined together, by crossover, to form new chromosomes, called *offspring*. By iteratively applying the crossover operator, genes of good chromosomes are expected to appear more frequently in the population, eventually leading to convergence to an overall good solution. The mutation operator introduces random changes into characteristics of chromosomes. Mutation reintroduces genetic diversity back into the population and assists the search escape from local optima.

Being a population-based approach, GA is well suited to solve multi-objective optimization problems finding a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces. In addition, most of multi-objective algorithms do not require the user to prioritize, scale, or weigh objectives. Consequently, GA has been the most popular heuristic approach to multi-objective design and optimization problems. Coello (2010) maintains an updated list with more than 5000 titles of publications involving different genetic algorithms.

The first multi-objective genetic algorithm, called *vector evaluated genetic algorithm*, or VEGA, was proposed by Schaffer (1985). Afterwards, several multi-objective evolutionary algorithms were developed, as shown by Konak *et al.* (2006), including the one that is used in this work to compare problems' results, the *non-dominated sorted genetic algorithm*, or NSGA II, proposed by Deb *et al.* (2000).

In the NSGA II, before selection is performed, the population is ranked on the basis of non-domination: all non-dominated individuals are classified into one category to provide an equal reproductive potential for these individuals. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population when the selection operator is applied. Additionally, the NSGA II estimates the density of solutions surrounding a particular solution in the population by computing the average distance of two points on either side of this point along each of the objectives of the problem. This value is called crowding

distance. During selection, the NSGA II uses a crowded-comparison operator which takes into consideration both the non-domination rank of an individual in the population and its crowding distance. The non-dominated solutions are preferred over dominated solutions, but between two solutions with the same non-domination rank, the one that resides in the less crowded region is preferred. The NSGA II uses the elitist mechanism that consists of combining the best parents with the best offspring obtained. Goldberg (1989) and Deb (2001) are excellent guides for GA implementations.

3.5.2 Particle swarm

Particle swarm optimization algorithm (PS) was originally proposed by Kennedy & Eberhart (1995). It is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. Although originally adopted for balancing weights in neural networks, PS soon became a very popular global optimizer.

There are two main distinctions between PS and GA. Genetic algorithms rely on three mechanisms in their processing. In contrast, PS only relies on two mechanisms, since PS does not adopt an explicit selection operator. The absence of a selection mechanism in PS is compensated by the use of leaders to guide the search. Such set of leaders is usually stored in a different place in an external archive, with all non- dominated solutions found so far. However, there is no notion of offspring generation in PS as with evolutionary algorithms. A second difference has to do with the way in which the individuals are manipulated. PS uses an operator that sets the velocity of a particle to a particular direction. This can be seen as a directional mutation operator in which the direction is defined by both the particle's personal best and the global best of the swarm. If the direction of the personal best is similar to the direction of the global best, the angle of potential directions will be small, whereas a larger angle will provide a larger range of exploration. In contrast, evolutionary algorithms use a mutation operator that can set an individual in any direction, although the relative probabilities for each direction may be different.

The main algorithm of PS is relatively simple, it only adopts one operator for creating new solutions and its implementation is, therefore, straightforward. Its main drawback is how to control de size of the external archive that implies in the computational time expended in the optimization process.

3.5.3 Simulated annealing

Simulated annealing optimization algorithm (SA) is based on an analogy of thermodynamics with the way metals cool and anneal. If a liquid metal is cooled slowly, its atoms form a pure crystal corresponding to the state of minimum energy for the metal. The metal reaches a state with higher energy if it is cooled quickly. Kirkpatrick *et al.* (1983) and Cerny (1985) showed that a model for simulating the annealing of solids, proposed by Metropolis *et al.* (1953), could be used for optimization of problems, where the objective function to be minimized corresponds to the energy of states of the metal. The first multi-objective version of SA has been proposed by

Serafini (1985, 1992). The algorithm of the method is almost the same as the algorithm of single objective SA. The slow convergence rate for some optimization problems is the main drawback of SA, as mentioned by Suman & Kumar (2006).

3.5.4 Ant Colony optimization algorithm

The ant colony optimization algorithm (AC) is inspired by the behavior of ants and other insects that live in a colony that in spite of the simplicity of each individual, present a high level of social organization when observed together. Some examples of ant colony's capabilities found in Dorigo *et al.* (1999) are: division of labor and task allocation, cemetery organization and brood sorting, cooperative transport and finding the shortest path between two or more locations, often between a food source and a nest.

The first AC algorithm developed was initially applied to the traveling salesman problem, Dorigo (1992). The algorithm was based on the ant colony capability to find the shortest path between a food source and a nest. The algorithm uses artificial ants that cooperate on finding solutions to the problem through communication mediated by artificial pheromone trails.

While moving on the graph associated with the problem, artificial ants deposit pheromone on the edges traversed marking a path that may be followed by other members of the colony, which then reinforce the pheromone on that path. With this kind of communication, ants have their activities coordinated. This self-organizing behavior results in a self-reinforcing process that leads to the formation of a path marked by high pheromone concentration, while paths that are less used tend to have a diminishing pheromone level due to evaporation.

This concept can be applied to any combinatorial optimization problem for which a constructive heuristic can be defined. The process of constructing solutions can be regarded as a walk on a construction graph where each edge of the graph represent a possible step the ant can take. AC is essentially constructive, as ants generate solutions by adding solution components, corresponding to the edges chosen, to an initially empty solution until the solution is complete.

3.5.5 Artificial immune systems

The human immune system has as its main task the detection of the infectious foreign elements, called pathogens, that attack us, and defend us from them, *i.e.*, its main task is to keep our organism healthy. Examples of such pathogens are bacteria and viruses. Any molecule that can be recognized by our immune system is called antigen. Such antigens provoke a specific response from our immune system. Lymphocytes of types B and T are special type of cells that play a major role in our immune system.

Upon detection of an antigen, the B cells that best recognize the antigen are cloned. Some of these cloned cells will be differentiated into plasma cells, which are the most active antibodies secretors, while others will act as memory cells. These cloned cells are subject to a high somatic mutation rate in order to increase their affinity level, *i.e.*, their matching to the antigens. These

mutations experienced by the clones are proportional to their affinity to the antigen. The highest affinity cloned cells experiment the lowest mutation rates, whereas the lowest affinity cloned cells have high mutation rates. Due to the random nature of this mutation process, some clones could be dangerous to the body and are, therefore, eliminated by the immune system itself. Plasma cells are capable of secreting only one type of antibody, which is relatively specific for the antigen. Antibodies play a key role in the immune response, since they are capable of adhering to the antigens, in order to neutralize and eliminate them.

Once the antigens have been eliminated by the antibodies, the immune system must return to its normal conditions, eliminating the in-excess cells. However, some cells remain in our blood stream acting as memory cells, so that our immune system can 'remember' the antigens that have previously attacked it.

When the immune system is exposed again to the same type of antigen or a similar one, these memory cells are activated, presenting a faster and perhaps improved response, which is called secondary response.

Based on the previous explanation of the way in which human immune system works, it can be say that, from a computer science perspective, the immune system can be seen as a parallel and distributed adaptive system. Clearly, the immune system is able to learn, it has memory, and is able of tasks such as associative retrieval of information. These features make immune systems very robust, fault tolerant, dynamic and adaptive. All of these properties can be emulated in a computer.

Artificial immune systems (AIS) are composed of the following basic elements: a) a representation for the components of the system, *e.g.*, binary strings, vectors of real numbers; b) a set of mechanisms to evaluate the interaction of individuals with their environment and with each other; such an environment is normally simulated through an affinity function, which is based on the objective functions in the case of optimization problems; c) procedures of adaptation, that indicate the way in which the behavior of the system changes over time; these procedure of adaptation consist of, for example, mutation operators.

The first direct use of the AIS for multi-objective optimization goes back to Yoo & Hajela (1999). In their work they use a standard genetic algorithm where the immune principle of antibody-antigen affinity is employed to modify the fitness value. In a first time the population is evaluated *versus* the problem objectives and different scalar values are obtained by making reference to different weighting combinations. The best individual with respect to each combination is identified as antigen. The rest of the population is the pool of antibodies. Then antibodies are matched against antigens by the definition of a matching score. The best matching antibody fitness is added by this matching score, evolving the population of antibodies to cover antigens.

4 DECISION MAKING IN MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Usually, when solving a MOOP, the DM expects, in general, to obtain its Pareto front and its Pareto set. Although it is very interesting to know these non-dominated solutions, additional

criteria are necessary to select a single solution that will be deployed. Even though useful to better understand the inter-relationships between the objectives, the Pareto front may become an obstacle since just one solution must be implemented. To make a selection, the DM will have to use an additional criterion, be it subjective or not.

Then, why not formalize and include this additional criterion into the problem and make it a *control function* to find the single and final solution to be deployed among those that belong to Pareto front?

Based on this idea, we propose a new methodology to formulate MOOP involving non-linear differentiable functions. The method transforms the MOOP from multi-objective into single-objective that can be solved with the aid of any traditional single-objective optimization engine suitable for the problem in focus.

4.1 Proposal of a new methodology for decision making in MOOP

In the proposed method, the objective functions are divided into two groups:

a) the *control function group*, or simply, *control function*, which contains only one function.

b) the *performance functions group*, which is made up of the functions that will provide the Pareto set.

Which function will be part of each group is a designer's choice and depends on his or her experience and knowledge related to the design problem. The control function can be an additional objective function or it can be elected among the problem objective functions.

The performance functions will have an important role in the process as they will provide the Pareto set as a constraint over which the control function will search the solution that optimizes it.

Thus, the multi-objective optimization problem can be written as:

$$\text{minimize:} \quad f_c(\mathbf{X}), \mathbf{f}_p(\mathbf{X}) \tag{12a}$$

$$\text{subject to:} \quad g_i(\mathbf{X}) \leq 0, \quad i = 1, 2, \ldots, m \tag{12b}$$

$$h_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, l \tag{12c}$$

$$\mathbf{X}_{\text{inf}} \leq \mathbf{X} \leq \mathbf{X}_{\text{sup}} \tag{12d}$$

where $f_c(\mathbf{X})$ is the *control function*,

$$\mathbf{f}_p(\mathbf{X}) = \left[f_1, f_2, \ldots, f_p\right] : \mathbf{X} \to \mathbb{R}^p$$

is the vector composed of the p objective functions in the *performance functions group*.

To apply the proposed methodology, the *performance functions*, is substituted by the KKT necessary condition, Eqs. (2a)-(2e), in such way that the problem's final solution belongs to the Pareto front of the MOOP with the performance functions only.

It should be noted that the weighting factors of the Eq. (2a), *i.e.*,

$$\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \ldots, \alpha_p\right]^T, \ \boldsymbol{\lambda} = \left[\lambda_1, \lambda_2, \ldots, \lambda_m\right]^T \quad \text{and} \quad \boldsymbol{\mu} = \left[\mu_1, \mu_2, \ldots, \mu_l\right]^T$$

are not known. As unknowns in the problem, they will be incorporated into the vector of design variables, defining the extended vector of unknowns:

$$\mathbf{X}_{extended} = (\mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \tag{13}$$

Finally, the problem is formulated as a single-objective optimization problem, with the control function, $f_c(\mathbf{X})$, to be minimized and constrained by the conditions for obtaining the Pareto optimal solutions considering only the performance functions, $f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_p(\mathbf{X})$.

Mathematically, the optimization problem is formulated as:

find $\mathbf{X}_{extended}$ that

minimizes: $\quad f_c(\mathbf{X})$ (14a)

subject to:

Pareto set condition for the performance functions:

$$\begin{cases} \displaystyle\sum_{i=1}^{p} \alpha_i \nabla f_i(\mathbf{X}) + \sum_{j=1}^{m} \lambda_j \nabla g_j(\mathbf{X}) + \sum_{i=1}^{l} \mu_i \nabla h_i(\mathbf{X}) = 0 & \text{(14b)} \\ g_i(\mathbf{X}) \leq 0, \quad i = 1, 2, \ldots, m & \text{(14c)} \\ h_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, l & \text{(14d)} \\ \lambda_j g_j(\mathbf{X}) = 0 & \text{(14e)} \\ \lambda_j \geq 0 & \text{(14f)} \\ \mu_i \geq 0 & \text{(14g)} \\ \alpha_i \geq 0; \quad \displaystyle\sum_{i=1}^{p} \alpha_i = 1 & \text{(14h)} \\ \mathbf{X}_{\text{inf}} \leq \mathbf{X} \leq \mathbf{X}_{\text{sup}} & \text{(14i)} \end{cases}$$

The proposed methodology differs from those methods exposed in the items 3.1 to 3.4. Those strategies require that a complete decision-making structure of the problem is decided *a priori*. Although the proposed method is also based on *a priori* decisions, just one function shall be added or isolated as a control function. The remaining functions are incorporated into the formulation to ensure that the final solution is within the Pareto set obtained from the minimization problem of these functions, even though the Pareto set and the Pareto front is not an explicit outcome of the process.

However, it must be considered that, with the use of Eq. (2a), the methodology applies only to problems where the functions involved are continuously differentiable. Furthermore, the solution

obtained is a local optimum and it can be a global optimum for convex Pareto fronts, since Eq. (2a) is a necessary condition for the existence of an extreme point and not a sufficient condition.

5 APPLICATIONS

To validate the proposed multi-objective optimization method, it will be used to solve three examples with increasing levels of complexity, namely the optimization problem of three quadratic functions, the design of a cantilever beam and the conceptual design of a bulk carrier.

To solve the single-objective optimization problem originated by the proposed methodology, any algorithm that works with optimization problems involving nonlinear functions and constraints can be used. Due to its accessibility, the solver *fmincon* was used. It is a component of the *Optimization Toolbox* available in the application MatLab, Matworks, version 7.1.

The NSGA II was used to compare results and to show that the results of the proposed method lay over the Pareto set from the MOOP of the performance functions group. A minor adaption was done in the NSGA II in order to interrupt the evolution if there is no significant difference between consecutive generations. Calling fi_j the value of the ith objective function for the jth chromosome at nth generation of a population with *pop* chromosomes, the root mean square of the objective functions for the population can be defined as

$$rms_n = \sqrt[2]{\sum_i^k \frac{\sum_j^{pop} \left(fi_j\right)^2}{\left(fi_{\max} - fi_{\min}\right)^2}} \tag{15}$$

with fi_{max} and fi_{min} being the maximum and the minimum value, respectively, of the ith objective function evaluated in the population. The alternative stop criterion was defined as

$$|rms_n - rms_{n-1}| \leq tol_f \cdot rms_{n-1} \tag{16}$$

where tol_f is a small number. Sometimes, for a defined tol_f value, this criterion may cause a premature interruption of the evolutionary process. To overcome this situation, the algorithm checks if it occurs in subsequent generations before discontinue the evolution. As a default, three subsequent generations were adopted with the satisfaction of Eq. (16) as the condition to interrupt the evolutionary process.

5.1 Minimization of three quadratic functions

Consider the minimization problem defined by:

$$\text{minimize:} \quad f_1(x_1, x_2) = 3(x_1+5)^2 + (x_1+5)(x_2-2) + (x_2-2)^2 \tag{17a}$$

$$f_2(x_1, x_2) = (x_1-5)^2 - (x_1-5)(x_2-3) + (x_2-3)^2 \tag{17b}$$

$$f_3(x_1, x_2) = (x_1+2)^2 + (x_2+6)^2 \tag{17c}$$

Assume that the functions $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ integrate the performance functions group and the function $f_3(x_1, x_2)$ is the control function. Accordingly, the problem can be formulated as:

$$\text{find } \mathbf{X}_{extended} = \left[x_1, x_2, \alpha_1, \alpha_2\right]^T$$

that

$$\text{minimize:} \quad f_c(x_1, x_2) = (x_1 + 2)^2 + (x_2 + 6)^2 \tag{18a}$$

$$\text{subject to:} \quad \alpha_1 \nabla f_1(x_1, x_2) + \alpha_2 \nabla f_2(x_1, x_2) = 0 \tag{18b}$$

$$\alpha_i \geq 0; \quad \sum_{i=1}^{2} \alpha_i = 1 \tag{18c}$$

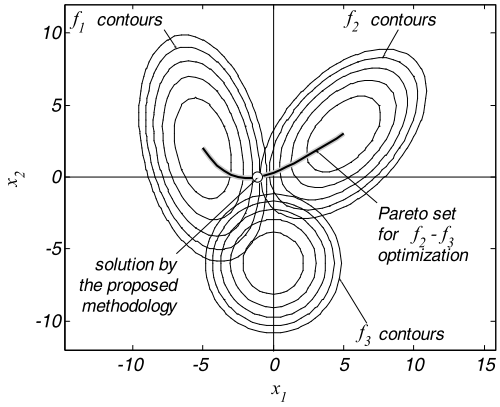
The solution of this problem, $(x_1, x_2, \alpha_1, \alpha_2) = (-1.1142, -0.0226, 0.3019, 0.6981)$, is shown in Table 1 and in the Figure (1a). The point that minimizes the control function falls into the Pareto set resulting from the MOOP formed by the performance functions. The Pareto sets shown in Figures (1a)-(1e) are not obtained by the methodology and they were computed for comparison only by a suitable algorithm.

Table 1 – Results for the minimization of quadratic functions.

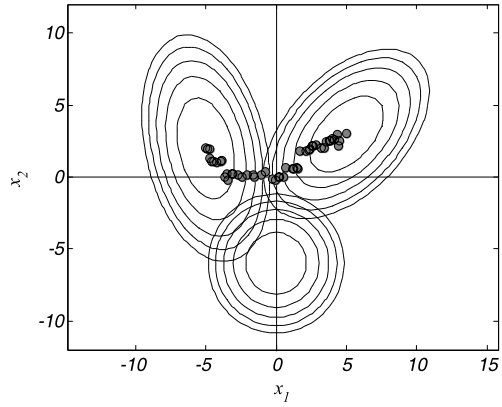
control function	method	x_1	x_2	α_1	α_2	function calls	computational time (s)
f_1	proposed	0.2955	–4.4558	0.2323	0.7677	40	0.041
	weighted	0.2942	–4.4603	0.2317	0.7683	880	0.878
	NSGA II	0.2000	–3.7026	na	na	6921*	5.591
f_2	proposed	–2.5707	–3.5394	0.3626	0.6374	47	0.051
	weighted	–2.5466	–3.5769	0.6423	0.3577	1001	0.947
	NSGA II	–2.9838	–1.3697	na	na	2739**	2.210
f_3	proposed	–1.1142	–0.0226	0.3019	0.6981	35	0.035
	weighted	–1.1242	–0.0245	0.3029	0.6971	898	0.951
	NSGA II	–0.1769	–0.3563	na	na	8434***	6.744
compromise solution		–1.9484	–1.5833	na	na	22	0.012
min-max solution		–1.5845	–0.5042	na	na	213	0.036

* 141 generations completed; ** 57 generations completed; *** 177 generations completed.

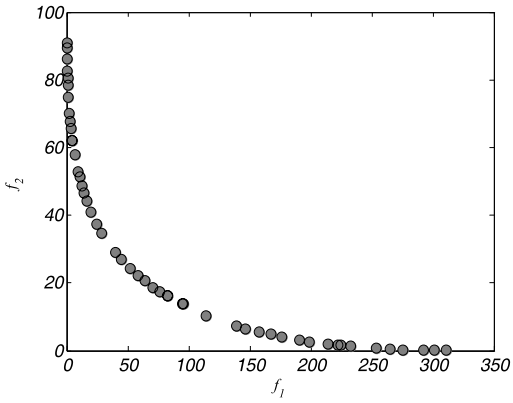
Figures (1b) and (1c) show the Pareto set and the corresponding Pareto front for the MOOP involving the performance functions $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ obtained by the NSGA II. Although the points in the Pareto front seem evenly distributed, the corresponding points in the Pareto set configure a rough approximation of the solution. To get this set, a 50 chromosomes population was used with the alternative stop of evolution criteria, Eq. (16), $tol_f = 0.001$. With these parameters the Pareto front stabilizes in 177 generations. The same parameter was used in the other NSGA II examples shown in Table 1.



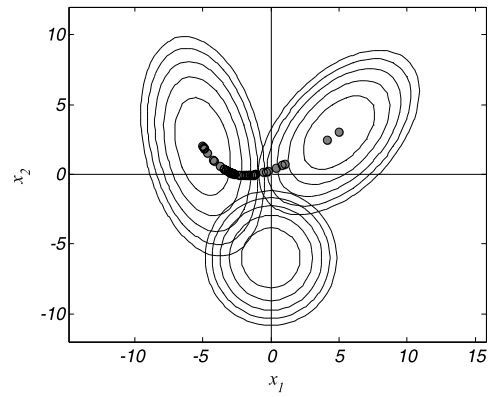
(a) Solution by the proposed methodology when f_3 is the control function.



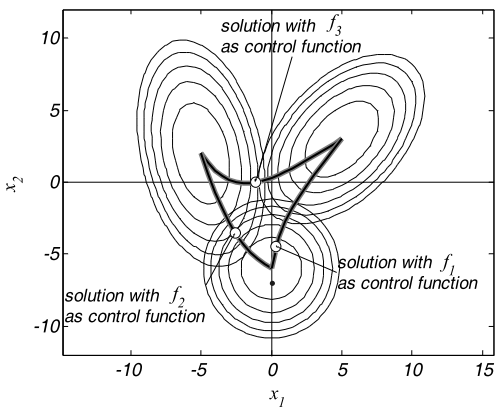
(b) Pareto set for the performance functions f_1 and f_2 obtained by the NSGA II algorithm.



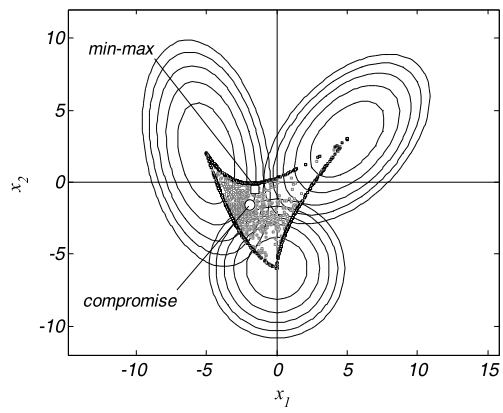
(c) Pareto front for performance functions f_1 and f_2 obtained by the NSGA II algorithm.



(d) Pareto set for the performance functions f_1 and f_2 obtained by the weighted sum approach.



(e) Solutions applying the metodology with different choices of the control function.



(f) Pareto set for the three quadratic functions MOOP obtained by the weighted sum approach.

Figure 1 – Results for the quadratic functions MOOP.

Figure (1d) shows the Pareto set for the same problem obtained by using the weighted sum approach, with 50 weight vectors chosen at random. This approach returns a well-matched Pareto set. Although the NSGA II is a powerful algorithm when solving MOOPs, it focuses on the criterion space and consequently can generate rough solutions in the design space as show the results in Figures (1b) and (1d).

Figure (1e) shows the results when another function is chosen to play the role of the control function. The resultant points are distinct but they are over the corresponding Pareto set resulting from the MOOP of the corresponding performance function group. Moreover, they belong to the Pareto set that results from the MOOP composed of the three quadratic functions, shown in Figure (1f). This Pareto set was obtained by the weighted sum approach with 1,200 sets of weights chosen at random. Observing Figure (1f), where the Pareto set is a region with infinite non-dominated points, the question that normally arises in the designer's mind is how to choose only one as a solution. In general, as more functions are aggregated to a MOOP, the more extended the Pareto set on the design space. In the example, with two functions, the Pareto set is a line. With three functions, the Pareto set is a plane surface.

In addition, comparisons for the computer performance between the algorithms are shown in Table 1. As there are no other methodologies similar to the one proposed in this paper, the results were compared to two other algorithms: the weighted sum approach and the NSGA II. For both, the process involved two phases: a) the search of the non-dominated designs for the MOOP involving the *performance functions* group and b) the manual search in the resultant set for the alternative that minimizes the *control function*. The number of function calls resulted from the first part of the process. The computational performance measure adopted for all algorithms' comparison is the number of objective functions calls. Although this measure is not perfect since both algorithms – the genetic algorithm and weight sum approach – depend on the number of points to be used in the MOOP solver, it has at least a qualitative value. For sake of comparison the computational times expended to achieve the final solution in a 2 GHz dual processor computer with 3 Gb RAM are also shown.

Two other methods that return a single solution are shown in Table 1 and figure (1f), the *compromise* and the *min-max* solutions. They are very fast, but they may not reflect the best compromise preferred by the DM.

Although the example is very simple, it shows how the proposed methodology can help decision making processes. The problem solution falls within the performance functions non-dominated solutions and it is the one that optimizes the control function.

In the quadratic functions example, the control function was chosen among the problem objective functions. In the next application, the MOOP is defined as minimizing the mass and tip deflection of a cantilever beam, two antagonist functions. With the proposed methodology these functions will compose the performance group. Afterward, it will be included a third function, the manufacturing cost, that will play the role of the control function.

5.2 Design of a cantilever beam

As a second illustrative example, consider the design problem of a cantilever beam, adapted from Deb (2001). The beam, prismatic and with a circular section, must support the weight P at its free end. The design variables are d and L, the cross section diameter and beam length, respectively. Also consider two attributes to be minimized, the beam mass and the beam tip deflection when subjected to the weight P. Moreover, the maximum bending stress must be below of the material yield stress, σ_y, and the maximum tip displacement limited to $\delta_{\max}$. A schematic representation of the problem is shown in Figure 2.

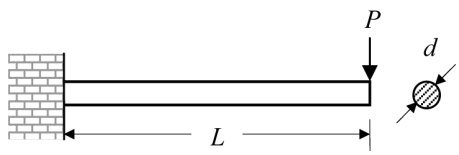


Figure 2 – Cantilever beam.

It can be demonstrated that the two selected objectives conflict since the minimization of the mass will lead to lower values of the pair (d, L) and minimization of the tip deflection will result in an increase in the cross sectional diameter with a reduction in the beam length.

The multi-objective optimization problem can be formulated as:

$$\text{minimize tip deflection:} \quad \delta(d, L) = 64PL^3/(3\pi Ed^4) \tag{19a}$$

$$\text{minimize beam mass:} \quad m(d, L) = 0.25\pi\rho d^2 L \tag{19b}$$

$$\text{subjected to:} \quad 200 \leq L(mm) \leq 1000 \tag{19c}$$

$$10 \leq d(mm) \leq 50 \tag{19d}$$

$$\delta(d, L) \leq d_{\max} \tag{19e}$$

$$\sigma(d, L) = 32PL/(\pi d^3) \leq \sigma_y \tag{19f}$$

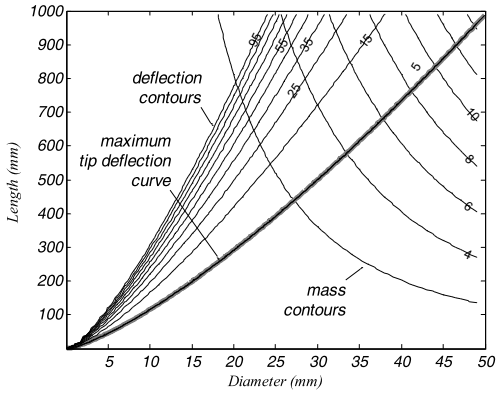
Adopting the following parameters:

$$P = 1kN; \quad \delta_{\max} = 5mm; \quad \sigma_y = 300MPa; \quad E = 210GPa; \quad \rho = 7850kg/m^3$$

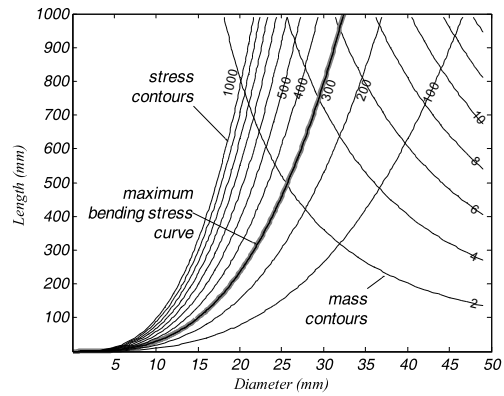
the shape of mass, tip deflection, and bending stress surfaces are shown in Figures (3a) and (3b).

A clearer view of the antagonism of the objective functions can be obtained using a method to find the Pareto front of multi-objective optimization problems. Approximations of the Pareto set and the Pareto front and are shown in Figures (4a) and (4b), respectively. These results were found by generating 10,000 feasible designs at random and then selecting the non-dominated ones.

Figure (4a) shows that all alternatives for the beam dimensions have length of 200 mm and cross section diameter ranging from 19 mm to 50 mm, approximately. These results are expected because, for a fixed diameter, to reduce the mass and the tip deflection the length decreases and the lower bound for this design variable is 200 mm, as shown in Figure (3a), but an inverse



(a) Contours of the tip deflection and the beam mass.



(b) Contours of the bending stress and the beam mass.

Figure 3 – Cantilever beam attributes.

relationship is observed considering the cross section diameter. Fixing the beam length, the lower the diameter the higher the tip deflection and the lower the beam mass.

Similar results were obtained by the NSGA II algorithm, using a 50 chromosomes population which has evolved 58 generations, conditioned by the stop criteria, Eq. (16), with $tol_f = 0.001$.

For comparison, Figures (4e) and (4f) show the results found by the weighted sum approach. Fifty random sets of weights were used to reduce the normalized mass and deflection beam functions to a single scale.

Now suppose that the problem must include the beam manufacturing cost and that it should be minimized. Additionally, suppose the cost function being defined by the function:

$$\$(d, L) = 100\left[\left(\frac{d-30}{50}\right)^2 + \left(\frac{d-30}{50}\right)\left(\frac{L-400}{1000}\right) + \left(\frac{L-400}{1000}\right)^2\right] \tag{20}$$

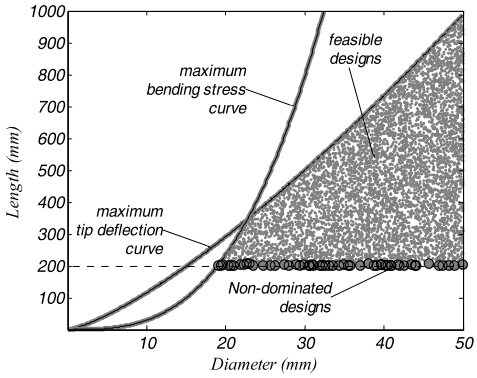
The contours of the cost function are shown in Figure (5a).

If the cost function is included as the third function in the MOOP, there is a growth and dispersion of Pareto optimal solutions in the design space, as shown in Figure (5b). This spread of the non-dominated alternatives in the design space makes decision making process even more difficult.

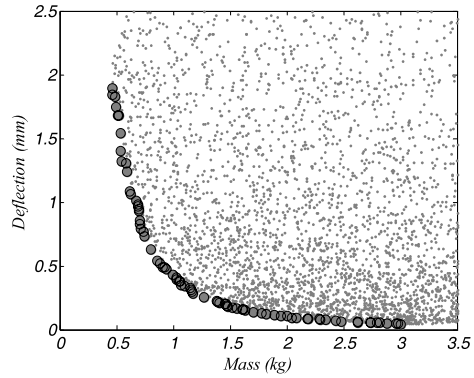
The proposed methodology can help the DM elect one alternative among all non-dominated, while preserving the concept of multi-objective optimization. The MOOP of the design of a cantilever beam can be re-formulated as:

> *Among all non-dominated design alternatives for the cantilever beam, where it is searched to minimize the beam tip deflection, $\delta(d, L)$, and minimize the beam mass, $m(d, L)$, find the one that has the minimum manufacturing cost.*

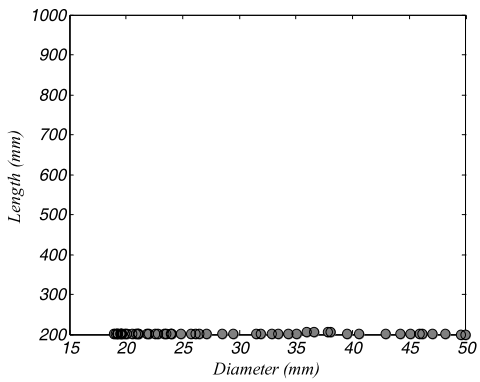
Another interpretation for problem formulation would be: among the solutions that satisfy a technical or engineering criterion, choose the one that has the lowest manufacturing cost.



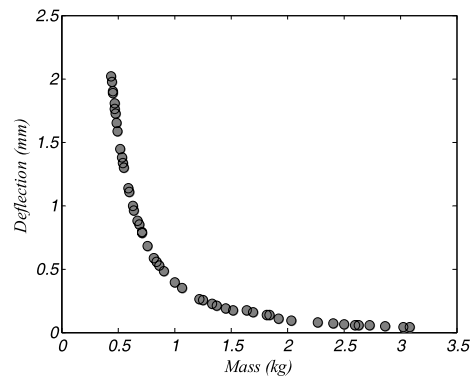
(a) Feasible and non-dominated designs for the cantilever beam in the design space.



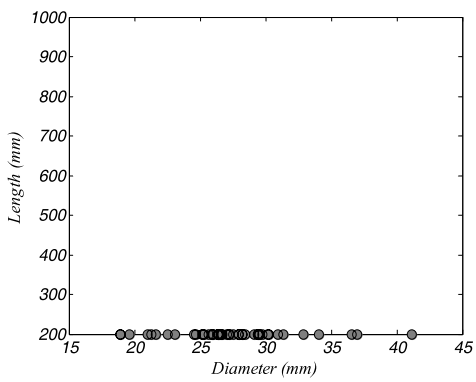
(b) Feasible and non-dominated designs for the cantilever beam in the criterion space.



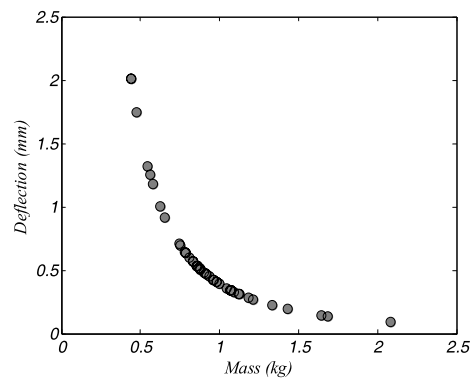
(c) Pareto set of the MOOP involving the beam mass and tip deflection obtained by the NSGA II algorithm.



(d) Pareto front of the MOOP involving the beam mass and tip deflection obtained by the NSGA II algorithm.

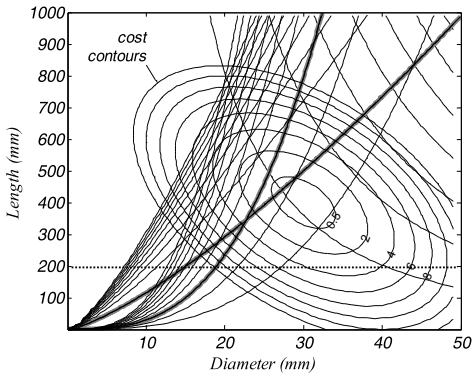


(e) Pareto set of the MOOP involving the beam mass and tip deflection obtained by the weighted sum approach.

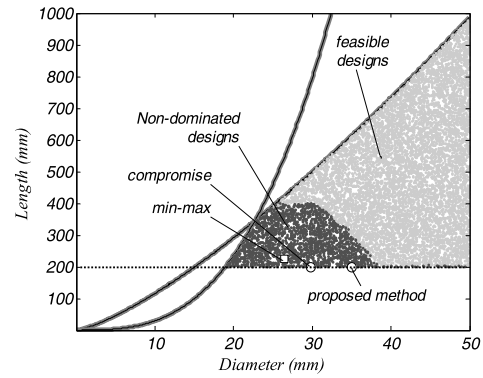


(f) Pareto front of the MOOP involving the beam mass and tip deflection obtained by the weighted sum approach.

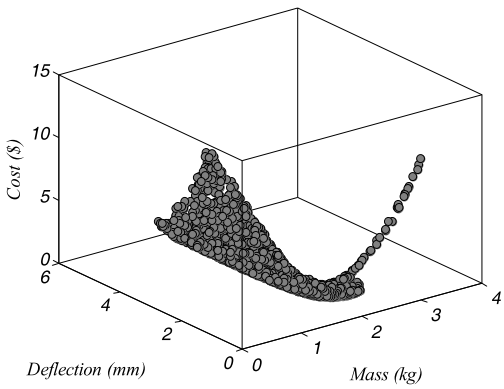
Figure 4 – Cantilever beam Pareto set and Pareto front found by distinct algorithms.



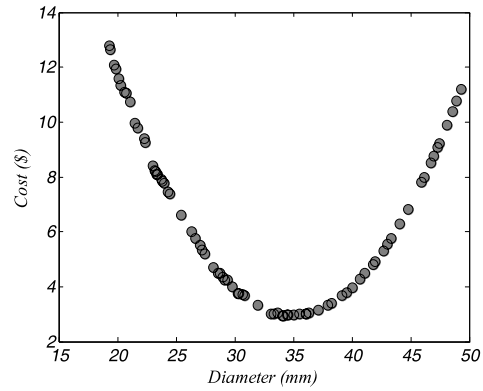
(a) Feasible and non-dominated designs for the cantilever beam in the design space.



(b) Non-dominated designs, in the design space, considering cost as the third objective function.



(c) Non-dominated designs, in criterion space, considering cost as the third objective function.



(d) Cost of the non-dominated designs obtained by the random walk and ilustrated in Figure (4a).

Figure 5 – Adding the cost function to the MOOP involving the design of a cantilever beam.

Applying the proposed methodology, the MOOP can be rewritten as:

$$\text{find } \mathbf{X}_{extended} = \left[x_1, x_2, \alpha_1, \alpha_2, \lambda_1 \cdots \lambda_6\right]^T$$

that

$$\text{minimizes:} \quad \$(d, L) = 100\left[\left(\frac{d-30}{50}\right)^2 + \left(\frac{d-30}{50}\right)\left(\frac{L-400}{1000}\right) + \left(\frac{L-400}{1000}\right)^2\right] \tag{21a}$$

$$\text{subject to:} \quad \alpha_1 \nabla\delta + \alpha_2 \nabla m + \sum_{j=1}^{6} \lambda_j \nabla g_j = 0 \tag{21b}$$

$$g_j \leq 0, \quad j = 1, 2, \ldots, 6 \tag{21c}$$

$$\lambda_j g_j = 0, \quad j = 1, 2, \ldots, 6 \tag{21d}$$

$$\lambda_j \geq 0 \tag{21e}$$

$$\alpha_i \geq 0; \quad \alpha_1 + \alpha_2 = 1 \tag{21f}$$

$$g_1 = \sigma(d, L) - \sigma_y \tag{21g}$$

$$g_2 = \delta(d, L) - \delta_{\max} \tag{21h}$$

$$g_3 = 10 - d \tag{21i}$$

$$g_4 = d - 50 \tag{21j}$$

$$g_5 = 200 - L \tag{21k}$$

$$g_6 = L - 1000 \tag{21l}$$

The *fmincon* was used to solve this single-objective optimization problem, which returns the following values

$$\begin{aligned} (d, L) &= \big(35.0077, 200.0000\big) \\ \boldsymbol{\alpha} &= \big[0.8144 \;\; 0.1856\big]^T \\ \boldsymbol{\lambda} &= \big[0.0000 \;\; 0.0000 \;\; 0.0000 \;\; 0.0000 \;\; 0.0035 \;\; 0.0000\big]^T \end{aligned}$$

Note that the design variables (d, L) and the weighting factors (only λ_5 is not zero, revealing that only the constraint equation g_5, Eq. (21k), is active) converged to a point on the Pareto optimal set for the MOOP involving the performance functions only, as shown in Figure (4a). For the sake of illustration, the values of the manufacturing cost calculated over the Pareto set obtained for the bi-objective optimization problem are shown in Figure (5d). The minimum cost is located in the diameter axis at the neighborhood of 35 mm.

Table 2 shows a comparison of the methodologies used to find the single solution for the beam design. Although the proposed methodology does not return the Pareto front it is very efficient to find the final solution with number of function calls significantly smaller than the other two methods used for the same purpose.

Table 2 – Results for the design optimization of a cantilever beam.

control function	method	d	L	α_1	α_2	function calls	computational time (s)
manufacturing cost	proposed	35.0	200.0	0.8142	0.1858	297	1.014
	weighted	34.1	200.0	0.7883	0.2117	2774	1.778
	NSGA II	34.3	200.0	na	na	2791*	4.075
compromise solution		29.9	200.0	na	na	75	0.0552
min-max solution		26.5	223.6	na	na	277	0.1176

* 58 generations completed.

To conclude, if the cost function is treated globally as the third function in MOOP, the set of Pareto optimal solutions will no longer be a line, as shown in Figure (4a); instead it is spread

over a surface, as shown in Figure (5b). In fact, the greater the number of conflicting functions in a multi-objective optimization problem, the greater the region in the space of the design variables that defines the Pareto optimal solutions. Consequently, the decision making process to elect a single solution is more difficult. How do we choose a non-dominated alternative among those present in Figure (5b)? The answer is impossible unless a fourth criterion is used to drive the choice. Rather, in the present proposal, this additional criterion is added to the MOOP as a control function to guide a convenient solution that satisfies all the constraints in a very efficient way, a solution that is non-dominated inside the group of the performance functions, and is an appropriate optimal solution obtained by optimizing the control function.

5.3 Conceptual design of a bulk carrier

The third application of the developed methodology was in the conceptual design of a bulk carrier. The conceptual design of a cargo vessel is not a trivial task. For decades, this problem has been handled in two ways, either by the adapting a known design aiming to the new requirements, or by the aid of simplified mathematical models driven by an optimization algorithm that enables obtaining the optimal solution based on technical or economic criteria previously established.

This work explores the second alternative with the help of the mathematical model for conceptual design of bulk carriers developed by Pratyush & Yang (1998) and presented in Appendix A. The model is made up of functions that define the vessel attributes from which are drawn those that constrain the design space, those to be optimized and those that characterize the technical and economic performance of the vessel and allow the evaluation of each design alternative. Among them are the *annual transportation cost*, the *annual transported cargo*, the *ship lightweight*, the *ship initial cost* and other functions of the vessel design variables such as length, width, depth, draft, block coefficient and speed, respectively (L, B, D, h, C_B, V_k). Pratyush & Yang (1998) proposed a general problem with explicit bounds on the block coefficient that should range between 0.63 and 0.75, and the ship speed that varies between 14 and 18 knots, and nine constraints in attributes, presented in Table A1 of Appendix A.

Parsons & Scott (2004) used the same model for comparing optimization methods, but they have altered the explicit bounds on the design variables. They limited the ship length at 274.32 m (900 ft) and the minimum deadweight at 25,000 ton, as shown in Table A2 of Appendix A. They also evaluated the problem by constraining the decision space in order to design ships suitable to cross the Panama Canal, with beam and draft limited to 32.31 m and 11.71 m, respectively, as shown in Table A3 of Appendix A.

Using Parsons and Scott's second model, Hart & Vlahopoulou (2009) constrained the depth to the minimum and the maximum of 13 m and 25 m, respectively, and extended the acceptable range for ship speed to the values of 11 knots and 20 knots, minimum and maximum respectively, as shown in Table A4 of Appendix A.

The bounds for the constraints feasibility lead to different results in optimization and the constraint bounds imposed in Pratyush & Yang's (1998) original work may be associated with the mathematical model, based on data reductions from operating ships and whose extrapolation

may undermine not only the quality but even the validity of the results obtained by using the model. In this paper was used the constraint bounds on design variables proposed in Pratyush & Yang's (1998) original work. For practical reasons, the limits they have not mentioned were adopted wide enough to not influence the optimization results. Accordingly, the ranges for the design variables are described in Table A5 of Appendix A.

5.3.1 Conflicting objectives

Pratyush & Yang (1998) chose to minimize the *annual transportation cost*, maximize the amount of *annual transported cargo* and minimize *vessel lightweight*. In this work, we replaced the vessel lightweight by the *ship initial cost*. Although *vessel lightweight* and *ship initial cost* functions are related, the latter has a financial appeal like the former two objective functions. *Annual transported cargo* is associated with the annual income, the *annual transportation cost* with the annual expenses and the *ship cost* is related to the capital required for the ship purchase.

The design alternative that optimizes the single-objective problem can be easily obtained by taking each objective function separately. Any algorithm running with non-linear optimization problems and constraints can be used to solve this single-objective optimization problem. The *fmincon* was used to get the results shown in Table 3.

Table 3 – Results of the single-objective optimization of each objective function.

Design variables and attributes		Units	Bounds		Objective functions		
					Minimize	Minimize	Maximize
			Lower	Upper	$T_C \downarrow$	$C_S \downarrow$	$C_A \uparrow$
Design Variables	Length	(m)	60	600	221.9	83.9	470.6
	Beam	(m)	10	100	37.0	14.0	78.4
	Depth	(m)	4	40	19.8	6.7	34.4
	Draft	(m)	3	30	14.6	5.4	24.8
	Block Coefficient		0.63	0.75	0.72	0.63	0.63
	Speed	(knots)	14	18	14.0	14.0	18.0
Ship Attributes	Annual Cost (T_C)	(10£/t)			0.797	2.259	1.519
	Vessel Cost (C_S)	(10^8£)			0.186	0.029	0.795
	Annual Cargo (C_A)	(10^6t/year)			0.750	0.058	1.271
	Deadweight	(t)	3000	500000	74513	3000	500000
	Main Power	(kW)			7968	1336	53666
	Metacentric Height	(m)			4.13	1.35	10.18
	Froud				0.15	0.25	0.14
	Round Trips				10.2	21.1	2.6

The results indicate that the objective functions are conflicting as to maximize the *annual cargo* (A_C); the optimum design will be a large ship with the highest speed of 18 knots. The search reaches the upper bound for the total deadweight, 500,000 t. The opposite occurs when it minimizes the *ship cost* (C_S). In this case, all vessel dimensions decrease along with the ship speed, which reaches the minimum of 14 knots with a slender shape vessel with block coefficient of 0.63, assuring a low power main engine and consequently a lower main engine cost. The search for even smaller vessels is constrained by the lower bound for the total deadweight of 3,000 tons. To minimize the *annual transportation cost* (C_T), the solution is situated between the previous two.

Table 4 – Statistics of the non-dominated solutions obtained by the random walk.

Design variables	L (m)	B (m)	D (m)	T (m)	C_B	V_k (knots)
$\bar{x}_i$	352.5	54.1	30.3	19.6	0.70	16.0
σ_i	75.4	11.6	6.6	4.1	0.03	1.2
$x_{\min}$	88.4	13.6	6.4	5.1	0.63	14.0
$x_{\max}$	493.5	77.1	40.0	26.3	0.75	18.0

Table 5 – Statistics for the Pareto set obtained by the weighted sum approach.

Design variables	L (m)	B (m)	D (m)	T (m)	C_B	V_k (knots)
$\bar{x}_i$	267.5	44.0	23.8	17.4	0.74	14.4
σ_i	82.2	13.7	7.4	5.1	0.03	0.8
$x_{\min}$	94.2	15.7	7.6	6.0	0.63	14.0
$x_{\max}$	425.5	68.9	36.6	26.3	0.75	18.0

5.3.2 Solutions for the MOOP

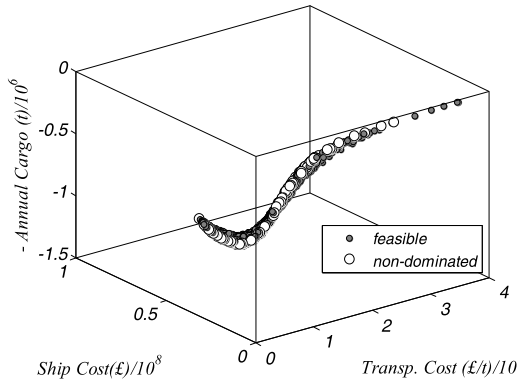
A clearer view of the antagonism of the objective functions in this problem can be obtained using an algorithm to find the Pareto front of multi-objective optimization problems. Figure 6 shows 10,000 feasible ship designs and highlights the non-dominated ones. To get these points, more than 12 million random design alternatives were generated and analyzed, taking 1097 seconds of computational time. Table 4 shows the design variables statistics for the feasible points. Although the problem shows a narrow surface in the criterion space, the design variables are well distributed over the design space.

With the aid of Figure 6, is possible to presume how the Pareto front of this problem should be, but, on the other hand, the random walk is not adequate to drive the search for the Pareto front, while the number of function calls becomes prohibitive.

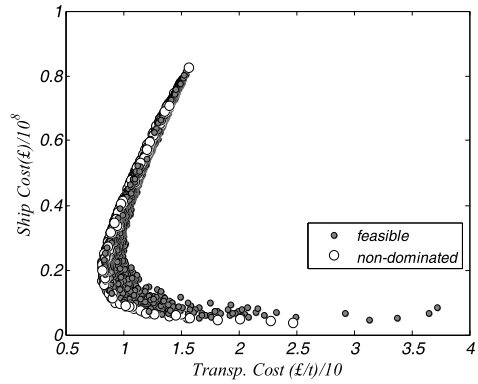
In order to compare the results of the proposed methodology with other methods, the weighted sum approach and the genetic algorithm were used to do the bulk carrier design optimization task.

Figure 7 and Table 5 show the results of the weighted sum approach algorithm with 100 sets of random weights. As *fmincon* fails in find the solution for several weights sets, the process were restarted from 5 different starting points for each set single run, being chosen the best result as the resultant point. Even though points were discarded, because they have shown to be dominated solutions.

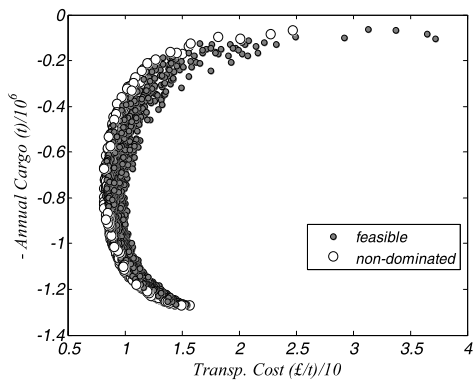
Better results are obtained from the NSGA II algorithm according to the points shown in Figure 8. These results were achieved for a population of 100 chromosomes, which has evolved 2000 generations. Table 6 shows the statistics for the non-dominated designs.



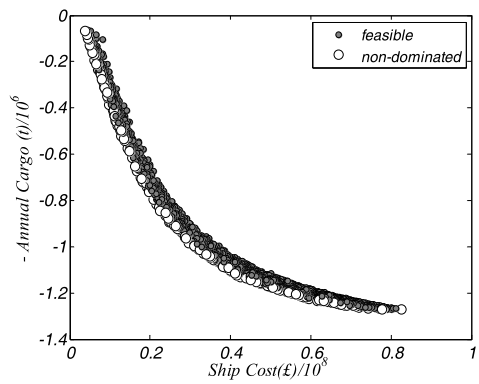
(a) Feasible and non-dominated designs for the bulk carrier obtained by a random walk in the design space.



(b) x-y view of the feasible and non-dominated designs.



(c) x-z view of the feasible and non-dominated designs.



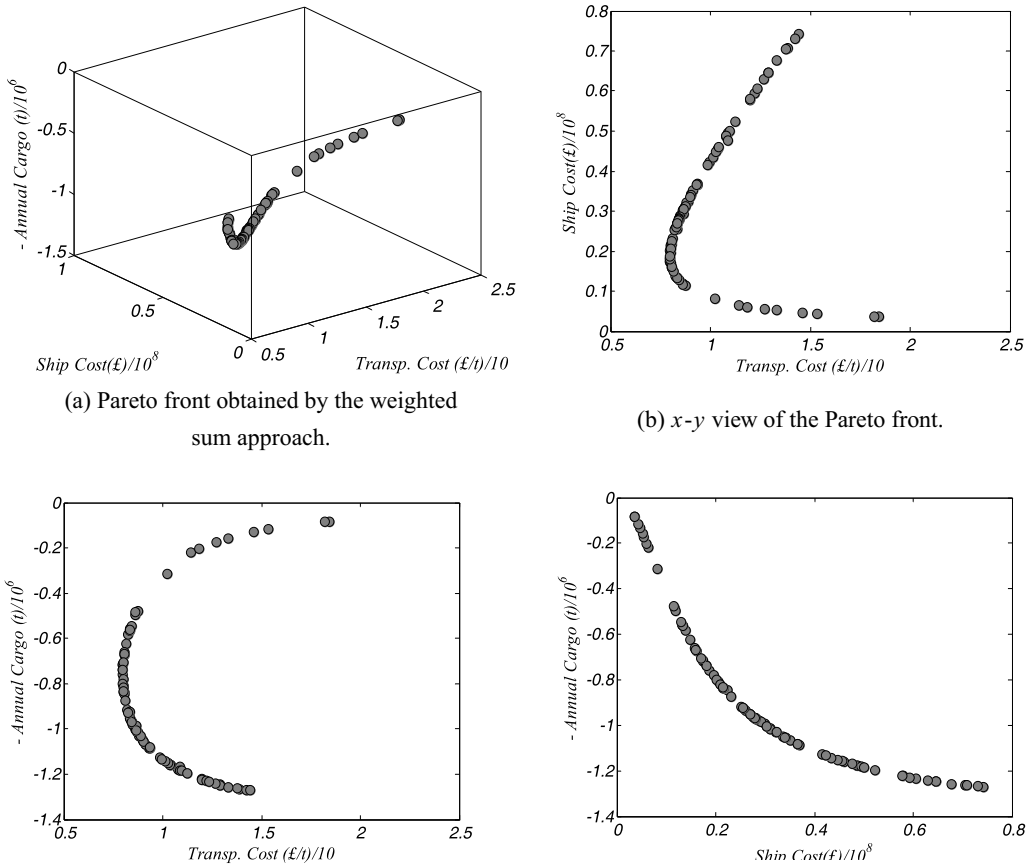
(d) y-z view of the feasible and non-dominated designs.

Figure 6 – Feasible and non-dominated designs for the MOOP with the *annual transportation cost* (C_T) minimization, the *ship cost* (C_S) minimization and the *annual cargo* (C_A) maximization, obtained by a random walk over the design variables.

The statistics for the design variables indicate a reasonable dispersion of the values. With the set of non-dominated solutions, shown in Figure 8, the question that often arises for the DM is: *which design alternative should be implemented*?

5.3.3 Applying the proposed methodology

Applying the proposed methodology one attribute function can be chosen as the control function and optimized in the Pareto front generated by the optimization of the performance functions group. Consider the previous functions, the *annual transportation cost*, the *annual cargo* and the *ship cost*, incorporating the performance group and the *voyage cost* as the control function. Accordingly, the multi-objective optimization problem can be defined as:



(a) Pareto front obtained by the weighted sum approach.

(b) x-y view of the Pareto front.

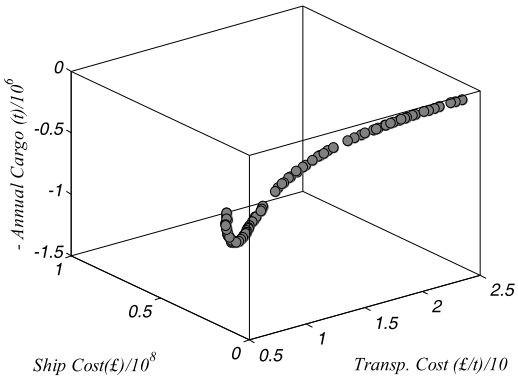
(c) x-z view of the Pareto front.

(d) y-z view of the Pareto front.

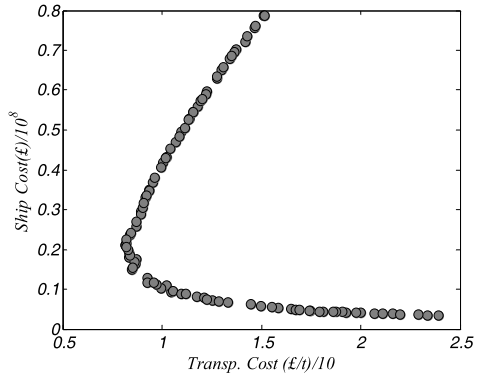
Figure 7 – Pareto front for the MOOP with the *annual transportation cost* (C_T) minimization, the *ship cost* (C_S) minimization and the *annual cargo* (C_A) maximization obtained by the weighted sum approach with 100 random weight sets.

> *among all the non-dominated solutions that would be obtained by resolving the MOOP for the bulk carrier design, namely, minimizing the annual transportation cost* (C_T)*, minimizing the ship cost* (C_S) *and maximizing the annual cargo* (C_A)*, find the solution that minimizes the voyage cost* (C_V).

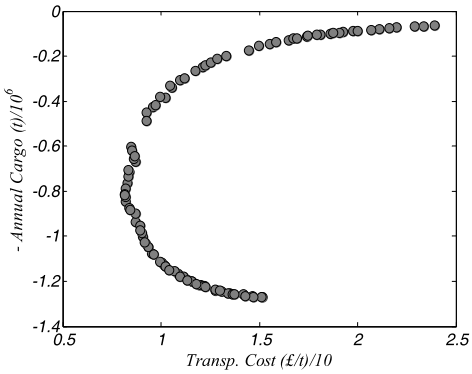
The mathematical formulation of this optimization problem is shown in detail in Appendix B. Note that the problem still is multi-objective, with four objective functions and no value function involved. However, with the use of the proposed approach, it can be formulated and solved in a single passage of any algorithm dedicated to single-objective optimization with inequality and equality constraints and able to work with non-linear functions. In this manner formulated, the problem will have 30 unknowns (six design variables of the original problem, plus three



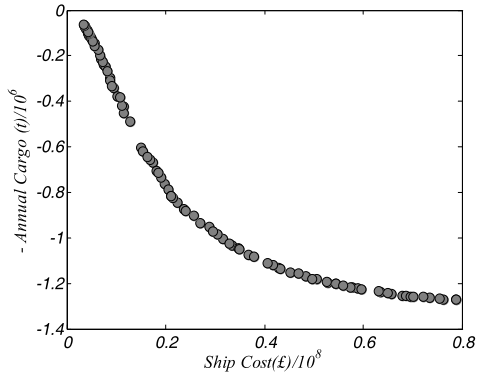
(a) Pareto front for the bulk carrier design obtained by the NSGA II algorithm.



(b) x-y view of the Pareto front.



(c) x-z view of the Pareto front.



(d) y-z view of the Pareto front.

Figure 8 – Pareto front for the MOOP with the *annual transportation cost* (C_T) minimization, the *ship cost* (C_S) minimization and the *annual cargo* (C_A) maximization obtained by the NSGA II algorithm.

weighting factors α_k related to the gradients of the performance functions and more 21 weighting factors λ_j related to the explicit and implicit constraints).

Table 7 shows the design alternative with some ship attributes obtained by using the *fmincon* algorithm. Additionally, the solution is highlighted in Figure 9 over the Pareto front of performance functions obtained thanks to the NSGA II.

Table 8 compares the results obtained by the proposed methodology with those obtained by the weighted sum approach and the NSGA II. The proposed methodology is very efficient in finding the optimized design as the number the function calls is significantly smaller.

It can be observed that the use of the proposed methodology leads to a solution that satisfies all the constraints and provides a good technical performance and its result is optimal according to the control function.

Table 6 – Statistics for the Pareto set obtained by the NSGA II.

Design variables	L (m)	B (m)	D (m)	T (m)	C_B	V_k (knots)
$\bar{x}_i$	238.0	38.3	20.3	14.6	0.71	14.8
σ_i	113.5	18.7	10.1	7.0	0.05	1.1
$x_{\min}$	84.2	14.0	6.4	5.0	0.63	14.0
$x_{\max}$	443.1	70.2	38.3	26.3	0.75	18.0

Table 7 – Results of the design optimization with the proposed methodology.

Design Variables			
	Length	98.8	(m)
	Beam	12.8	(m)
	Depth	6.6	(m)
	Draft	5.2	(m)
	Block Coefficient	0.63	
	Speed	14.0	(knots)

Ship Attributes			
	Annual Cost (T_C)	2.276	(10£/t)
	Vessel Cost (C_S)	0.0311	(10^8£)
	Annual Cargo (C_A)	0.059	(10^6t/year)
	Voyage Cost (C_V)	1.2855	(10^4£)
	Deadweight	3000	(t)
	Main Power	1225	(kW)
	Metacentric Height	0.89	(m)
	Froud	0.23	
	Round Trips per Year	21.1	

Table 8 – Results for the design optimization of a bulk carrier.

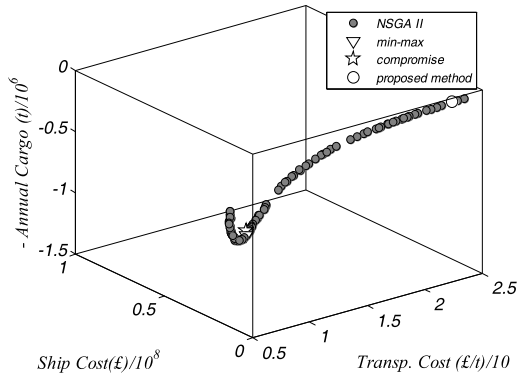
method	L, B, D, h (m)	C_B	V (knots)	control function (£)	function calls	computational time (s)
proposed	98.8, 12.8, 6.6, 5.2	0.63	14.0	12855	1031	7.872
weighted	94.2, 15.7, 7.6, 6.0	0.63	14.0	16466	69666	73.236
NSGA II	86.3, 13.5, 6.7, 5.4	0.64	14.0	13660	75783*	272.125
compromise	270.3, 45.1, 22.4, 16.4	0.64	14.0	na	239	0.206
min-max	258.8, 43.1, 23.5, 17.2	0.75	14.0	na	519	0.474

* 797 generations completed.

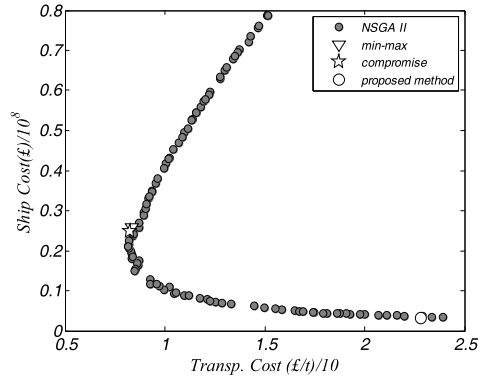
6 CONCLUSION

Most engineering design problems are multi-objective and the cases where the objective functions do not conflict are rare. To solve these kinds of problems, many researchers developed methods to search for the solution of multi-objective optimization without simplifying the problem to single-objective and having to decide *a priori* how to group the objective functions into a single scale. The evolutionary methods are frequently used to locate the set of solutions of multi-objective problems. These algorithms provide a discrete picture of the Pareto front in the criterion space. It was observed that the greater the number of objective functions, the more scattered the set of non-dominated solutions is in the decision space and the harder it is for the DM to choose an alternative to be deployed.

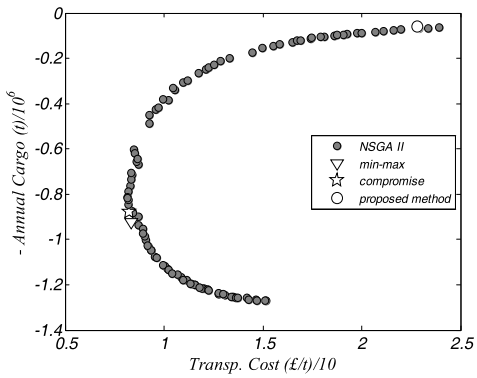
This paper proposes a new methodology to solve multi-objective optimization problems in which one objective function is proposed or isolated and is treated as a *control function* that will drive the decision-making process. The other objective functions will form the *performance functions*



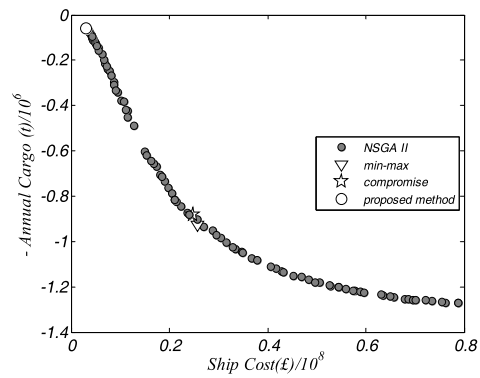
(a) Solution for the bulk carrier design obtained by the proposed methodology.



(b) x-y view of Pareto front of the bulk carrier design obtained by the NSGA II.



(c) Pareto front of the bulk carrier design obtained by the NSGA II.



(d) Pareto front of the bulk carrier design obtained by the NSGA II.

Figure 9 – Pareto front obtained by the NSGA II for the MOOP involving the performance group {*annual transportation cost* (C_T), *ship cost* (C_S), and *annual cargo* (C_A)} and solution obtained by the proposed methodology with *voyage cost* (C_V) as the control function.

group. Through this strategy, a single-objective optimization problem is formulated, in which the control function will be optimized over the Pareto set that would result from the optimization problem established by the performance functions, if this problem was previously solved.

Then the resultant single-objective optimization problem can be solved by any classical standard optimization engines, the limitations will be those present in the method used, such as the nature of the functions, whether they are linear or nonlinear, or continuous, regarding the presence or not of inequality and equality constraints.

With the proposed method a minimum knowledge is needed *a priori*. There is no need to know a value function relation to the objectives before starting to solve the problem as other *a priori* articulation of preferences methods.

As drawbacks, the method doesn't provide a set of non-dominated solutions and as in many other single-objective optimization algorithms, the solution converges to a local optimum since the conditions of Karush-Kuhn-Tucker, chief pillar of the proposed methodology, are the necessary conditions but not sufficient to ensure that the solution is in the "global" Pareto front. If the problem is convex and the functions involved are regular, the local solution also will be a global solution, ensuring a unique result for the problem. Another disadvantage, the functions involved in the problem shall be continuously differentiable.

Although these conditions are limiting, the proposed methodology is very efficient in solving engineering design problems, as demonstrated by the examples solved with its use.

REFERENCES

[1] BOUYER E, CARO S, CHABLAT D & ANGELES, J. 2007. *The Multiobjective Optimization of a Prismatic Drive*, in Proceedings of ASME Design Engineering Technical Conferences, September 4-7, Las Vegas, Nevada, U.S.

[2] CHARNES A & COOPER WW. 1977. Goal programming and multiple objective optimisation – Part 1. *European Journal of Operational Research*, **1**: 39–54.

[3] CHINCHULUUN A & PARDALOS PM. 2007. A survey of recent developments in multiobjective optimization. *Ann. Oper. Res.*, **154**: 29–50.

[4] COELLO CAC. 2000. An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys* (CSUR), **32**(2).

[5] COELLO CAC. 2010. List of References on Evolutionary Multiobjective Optimization, in URL <http://www.lania.mx/~ccoello/EMOO/EMOObib.html>, April 24, 2010.

[6] DEB K, AGRAWAL S, PRATAB A & MEYARIVAN T. 2000. *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization, NSGA*, KanGAL report 200001.

[7] DEB K. 2001. *Multi-objective Objective Optimization using Evolutionary Algorithms*, Wiley and Sons Ltd.

[8] EDGEWORTH FY. 1881. *Mathematical Psychic: an essay on the application of mathematics to the moral sciences*, Editor P. Kegan, London, England.

[9] GIASSI A, BENNIS F & MAISONNEUVE JJ. 2004. Multi-Objective Optimization Using Asynchronous Distributed Applications. *J. Mech. Des.*, **26**: 767.

[10] GOLDBERG D. 1989. *Genetic Algorithms in Search and Machine Learning*, Reading, Addison Wesley.

[11] HAIMES YY, LASDON LS & WISMER DA. 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Trans. Syst. Man Cybern. SMC-1*, 296–297.

[12] HART CG & VLAHOPOULOS N. 2009. *An Integrated Multidisciplinary Particle Swarm Optimization Approach to Conceptual Ship Design*, Struct. Multidisc. Optim. DOI 10.1007/s00158-009-0414-0. Springer-Verlag.

[13] HOLLAND J. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

[14] HWANG CLM, MASUD AS, in collaboration with PAIDY SR & YOON K. 1979. Multiple objective decision making, methods and applications: a state-of-the-art survey. In: BECKMANN M AND KUNZI HP (eds.). *Lecture Notes in Economics and Mathematical Systems*, **164**. Berlin: Springer-Verlag.

[15] KENNEDY J & EBERHART RC. 1995. *Particle swarm optimization*. In: Proceedings of the IEEE International Conference on Neural Networks, Piscataway, New Jersey, 1942–1948.

[16] KIRKPATRICK S, GELATT JR CD & VECCHI MP. 1983. Optimization by simulated annealing. *Science*, **220**: 671–680.

[17] KONAK A, COIT DW & SMITH AE. 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, **91**: 992–1007.

[18] LIGHTNER MR & DIRECTOR SW. 1981. Multiple criterion optimisation for the design of electronic circuits. *IEEE Transactions on Circuits and Systems*, **28**(3): 169–179.

[19] MARLER RT & ARORA JS. 2004. Survey of Multi-objective Optimization Methods for Engineering. *Structural Multidisciplinary Optimization*, **26**: 369–395.

[20] MIETTINEN KM. 1998. *Nonlinear Multiobjective Optimization*, Springer-Verlag.

[21] OSYCZKA A. 1984. *Multicriterion Optimiration in Engineering with FORTRAN Programs*, Ellis Horwood Ltd, Chichester, UK.

[22] PARETO V. 1971. *Manual of Political Economy*, Translated by Schiwier, A.S. from French Edition of 1927, Augustus M. Kelley Publishers, New York.

[23] PARSONS MG & SCOTT RL. 2004. Formulation of Multicriterion Design Optimization Problems for Solution with Scalar Numerical Optimization Methods. *J. Ship Res.*, **48**(1): 61–76.

[24] PARSONS MG. 2004. *Parametric Ship Design, in Ship Design and Construction*. Thomas Lamb Editor, Vol. 1, SNAME, 11, 11.1–11.50.

[25] PRATYUSH S & YANG JB. 1998. *Multiple Criteria Decision Support in Engineering Design*, Springer-Verlag.

[26] RAO AR, SCANLAN JP & KEANE AJ. 2007. Applying Multiobjective Cost and Weight Optimization to the Initial Design of Turbine Disks. *J. Mech. Des.*, **129**: 1303.

[27] SCHAFFER JD. 1985. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, in Proceedings of the first International Conference on Genetic Algorithms and their Applications, 93-100, Lawrence Erlgaunn, Pittsburgh, PA.

[28] SERAFINI P. 1994. *Simulated annealing for multiple objective optimization problems*. In: TZENG GH, WANG HF, WEN VP AND YU PL (eds.). Multiple Criteria Decision Making. Expand and Enrich the Domains of Thinking and Application. Springer- Verlag, Berlin, 283–292.

[29] SUMAN B & KUMAR P. 2006. A survey of simulated annealing as a tool for singleand multiobjective optimization. *Journal of the Operational Research Society*, **57**: 1143–1160.

[30] STADLER W. 1988. *Applications of Multicriteria Optimization in Engineering and the Sciences*, **1**: 1–24, Editor STADLER W., Editor Plenum Press, New York.

[31] XUEBIN L. 2009. Multiobjective Optimization and Multiattribute Decision Making Study of Ship's Principal Parameters in Conceptual Design. *J. Ship Res.*, **53**(2): 83–92.

[32] ZADEH LA. 1963. Optimality and non-scalar-valued performance criteria. *IEEE Trans. Autom. Control AC-8*, 59–60.

APPENDIX A: CONCEPTUAL MODEL FOR BULK CARRIERS

Design Variables (6)

Length (L)		(m)
Beam (B)		(m)
Draft (T)		(m)
Depth (D)		(m)
Block coefficient (C_B)		(m)
Velocity (V_k)		(knots)

Objectives (3)

minimize	Ship costs (C_S)	(£)
minimize	Transportation cost (C_T)	(£/T)
maximize	Annual cargo (A_C)	(t)

Ship attributes

Steel weight (w_s)	$0.034L^{1.7}B^{0.7}D^{0.4}C_B^{0.5}$	(t)
Outfit weight (w_o)	$1.0^{0.8}B^{0.6}D^{0.3}C_B^{0.1}$	(t)
a	$4977.06C_B^2 - 8105.61C_B + 4456.51$	
b	$-10847.2C_B^2 - 12817C_B - 6960.32$	
Salt water specific weight (γ)	1.025	(t/m^3)
Gravity (g)	9.81	(m/s^2)
Displacement (Δ)	$\gamma LBTC_B$	(t)
Froud number (F_n)	$V/\sqrt{gL},\ V = 0.5144V_k$	
Power (P)	$\sqrt[3]{\Delta^2}V_k^3/(a + bF_n)$	(kW)
Machinery weight (w_m)	$0.17P^{0.9}$	(t)
Ship lightweight (w_{ls})	$w_s + w_o + w_m$	(t)
Deadweight (dwt)	$\Delta - w_{ls}$	(t)
Fuel rate (f_R)	190	(g/(kWh))
Daily fuel consumption (D_{FC})	$24f_R10^{-6}P + 0.2$	(t/day)
Round trip miles (R_{TM})	5000	(M)
Sea days (D_S)	$R_{TM}/(24V_k)$	
Fuel price (F_P)	100	(£/t)
Fuel cost (C_F)	$1.05D_{FC}D_SF_P$	(£)
Port cost (C_P)	$6.3dwt^{0.8}$	(£)
Fuel carried (F_C)	$D_{FC}(D_S + 5)$	(t)
Misc. deadweight (dwt_m)	$2.0dwt^{0.5}$	(t)
Cargo deadweight (dwt_c)	$dwt - F_C - dwt_m$	(t)

Handling rate (H_R)	8000	(t/day)
Port days (D_P)	$2(dwt_C/H_R + 0.5)$	
Round trips per year (R_{TPY})	$350/(D_S + D_P)$	
• Voyage costs (C_V)	$(C_F + C_P)R_{TPY}$	(£)
• Ship cost (C_S)	$1.3(2000w_s^{0.85} + 3500w_o + 2400P^{0.8})$	(£)
Capital cost (C_C)	$0.2C_S$	(£)
Running cost (C_R)	$40000dwt^{0.3}$	(£)
Annual cost (C_A)	$C_C + C_R + C_V$	(£)
• Annual cargo (A_C)	$dwt_c R_{TPY}$	(t/year)
• Transportation cost (C_T)	C_A/A_C	(£/t)
Vertical center of buoyance (KB)	$0.53T$	(m)
Metacentric radius (BM_T)	$(0.085C_B - 0.002)B^2/(TC_B)$	(m)
Vertical center of gravity (KG)	$1.0 + 0.52D$	(m)

Constraints

Table A1 – Pratyush S., Yang J.B., 1998, (13 constraints).

Explicit (4)	Implicit (9)
$0.63 \leq C_B \leq 0.75$	$L/B \geq 6$
$14 \leq V_k \leq 18$	$L/D \leq 15$
	$L/T \leq 19$
	$T \leq 0.45dwt^{0.31}$
	$T \leq 0.7D + 0.7$
	$KB + BM_T - KG \geq 0.07B$
	$3000 \leq dwt \leq 500000$
	$F_n \leq 0.32$

Table A2 – Parsons M.G., Scott R.L., 2004 (model a), (14 constraints).

Explicit (5)	Implicit (9)
$L \leq 274.32$	$L/B \geq 6$
$0.63 \leq C_B \leq 0.75$	$L/D \leq 15$
$14 \leq V_k \leq 18$	$L/T \leq 19$
	$T \leq 0.45dwt^{0.31}$
	$T \leq 0.7D + 0.7$
	$KB + BM_T - KG \geq 0.07B$
	$25000 \leq dwt \leq 500000$
	$F_n \leq 0.32$

Table A3 – Parsons M.G., Scott R.L., 2004 (model b), (16 constraints).

Explicit (7)	Implicit (9)
$L \leq 274.32$	$L/B \geq 6$
$T \leq 11.71$	$L/D \leq 15$
$B \leq 32.31$	$L/T \leq 19$
$0.63 \leq C_B \leq 0.75$	$T \leq 0.45dwt^{0.31}$
$14 \leq V_k \leq 18$	$T \leq 0.7D + 0.7$
	$KB + BM_T - KG \geq 0.07B$
	$25000 \leq dwt \leq 500000$
	$F_n \leq 0.32$

Table A4 – Hart C.G., Vlahopoulos N., 2009, (21 constraints).

Explicit (12)	Implicit (9)
$150 \leq L \leq 274.32$	$L/B \geq 6$
$20 \leq B \leq 32.31$	$L/D \leq 15$
$13 \leq D \leq 25$	$L/T \leq 19$
$10 \leq T \leq 11.71$	$T \leq 0.45dwt^{0.31}$
$0.63 \leq C_B \leq 0.75$	$T \leq 0.7D + 0.7$
$14 \leq V_k \leq 20$	$KB + BM_T - KG \geq 0.07B$
	$25000 \leq dwt \leq 500000$
	$F_n \leq 0.32$

Table A5 – Adopted in the present work (21 constraints).

Explicit (12)	Implicit (9)
$60 \leq L \leq 600$	$L/B \geq 6$
$10 \leq B \leq 100$	$L/D \leq 15$
$4 \leq D \leq 40$	$L/T \leq 19$
$3 \leq T \leq 30$	$T \leq 0.45dwt^{0.31}$
$0.63 \leq C_B \leq 0.75$	$T \leq 0.7D + 0.7$
$14 \leq V_k \leq 18$	$KB + BM_T - KG \geq 0.07B$
	$3000 \leq dwt \leq 500000$
	$F_n \leq 0.32$

APPENDIX B:
BULK CARRIERS DESIGN MODEL WITH THE PROPOSED METHODOLOGY

To apply the proposed methodology is convenient to define the following entities:

a) The control function:

voyage cost: C_v

b) The performance functions: $\mathbf{f}_p = \left[C_S, C_T, A_C\right]^T$

ship cost: C_S

transportation cost: C_T

annual cargo: A_C

c) The implicit constraints: $g_c = \left[g_1, g_2, \ldots, g_9\right]^T$

$g_1 = 6 - L/B$ $\quad$ $g_6 = 0.07B - KB - BM_T - KG$

$g_2 = L/D - 15$ $\quad$ $g_7 = dwt - 500000$

$g_3 = L/T - 19$ $\quad$ $g_8 = 3000 - dwt$

$g_4 = T - 0.45dwt^{0.31}$ $\quad$ $g_9 = F_n - 0.32$

$g_5 = T - 0.7D + 0.7$

d) The explicit lower bound constraints: $g_{\text{inf}} = \mathbf{X}_{\text{inf}} - \mathbf{X} = \left[g_{10}, g_{11}, \ldots, g_{15}\right]^T$

$g_{10} = 60 - L$ $\quad$ $g_{13} = 3 - T$

$g_{11} = 10 - B$ $\quad$ $g_{14} = 0.63 - C_B$

$g_{12} = 4 - D$ $\quad$ $g_{15} = 14 - V_k$

e) The explicit upper bound constraints: $g_{\text{sup}} = \mathbf{X} - \mathbf{X}_{\text{sup}} = \left[g_{16}, g_{17}, \ldots, g_{21}\right]^T$

$g_{16} = L - 600$ $\quad$ $g_{19} = T - 30$

$g_{17} = B - 100$ $\quad$ $g_{20} = C_B - 0.75$

$g_{18} = D - 40$ $\quad$ $g_{21} = V_k - 18$

f) The (unknown) design variables:

$\mathbf{X} = \left[L, B, D, T, C_B, V_k\right]^T$

g) The (unknown) performance functions gradient multipliers:

$\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \alpha_3\right]^T$

h) The (unknown) implicit constraints gradient multipliers:

$\boldsymbol{\lambda}_c = \left[\lambda_1, \lambda_2, \ldots, \lambda_9\right]^T$

i) The (unknown) explicit lower bound constraints gradient multipliers:

$\boldsymbol{\lambda}_{\text{inf}} = \left[\lambda_{10}, \lambda_{11}, \ldots, \lambda_{15}\right]^T$

j) The (unknown) explicit upper bound constraints gradient multipliers:
$\boldsymbol{\lambda}_{\text{sup}} = \left[\lambda_{16}, \lambda_{17}, \ldots, \lambda_{21}\right]^T$

k) The gradient vector:
$\boldsymbol{\nabla} = \left[\frac{\partial(\cdot)}{\partial L}, \frac{\partial(\cdot)}{\partial B}, \frac{\partial(\cdot)}{\partial D}, \frac{\partial(\cdot)}{\partial T}, \frac{\partial(\cdot)}{\partial C_B}, \frac{\partial(\cdot)}{\partial V_k}\right]^T$

Let the Karush-Kuhn-Tucker condition be defined by:

$$\mathbf{F}(\mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{inf}}, \boldsymbol{\lambda}_{\text{sup}}) = \boldsymbol{\nabla}\mathbf{f}_k^T\boldsymbol{\alpha} + \boldsymbol{\nabla}\mathbf{g}_c^T\boldsymbol{\lambda}_c + \boldsymbol{\nabla}\mathbf{g}_{\text{inf}}^T\boldsymbol{\lambda}_{\text{inf}} + \boldsymbol{\nabla}\mathbf{g}_{\text{sup}}^T\boldsymbol{\lambda}_{\text{sup}}$$

Then, with the proposed methodology, the multi-objective optimization problem involving the bulk carrier design can be rewritten as the following single-objective optimization problem:

find

$\mathbf{X}_{\text{extended}} = (\mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{\text{inf}}, \boldsymbol{\lambda}_{\text{sup}})$ that

minimizes the voyage cost:

$$C_v \tag{B1}$$

subject to the following constraints:

$$\mathbf{F} = \mathbf{0} \tag{B2}$$

$$\lambda_j g_j = 0, j = 1, 2, \ldots, 21 \tag{B3}$$

$$\sum_{i=1}^{3} \alpha_i - 1 = 0 \tag{B4}$$

$$g_j \leq 0, \quad j = 1, 2, \ldots, 21 \tag{B5}$$

$$\lambda_j \geq 0 \tag{B6}$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, 3 \tag{B7}$$

This single-objective optimization problem was solved by fmincon which returns the following values:

$$\mathbf{X} = \left[98.78,\ 12.76,\ 6.59,\ 5.22,\ 0.63,\ 14.00\right]^T$$

$$\boldsymbol{\alpha} = \left[0.00,\ 0.00,\ 1.00\right]^T$$

$$\boldsymbol{\lambda}_c = \left[0.40,\ 0.02,\ 0.03,\ 0.13,\ 0.81,\ 0.01,\ 0.00,\ 0.01,\ 0.97\right]^T$$

$$\boldsymbol{\lambda}_{\text{inf}} = \left[0.00,\ 0.00,\ 0.00,\ 0.02,\ 1.00,\ 0.46\right]^T$$

$$\boldsymbol{\lambda}_{\text{sup}} = \left[0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.01\right]^T$$