Computational Methods in Applied Sciences

António Gaspar-Cunha · Jacques Periaux ·
Kyriakos C. Giannakoglou ·
Nicolas R. Gauger · Domenico Quagliarella ·
David Greiner *Editors*

Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences

ECCOMAS
European Community
on Computational Methods
in Applied Sciences

Springer

Computational Methods in Applied Sciences

Volume 55

Series Editor

Eugenio Oñate, Universitat Politècnica de Catalunya, Barcelona, Spain

This series publishes monographs and carefully edited books inspired by the thematic conferences of ECCOMAS, the European Committee on Computational Methods in Applied Sciences. As a consequence, these volumes cover the fields of Mathematical and Computational Methods and Modelling and their applications to major areas such as Fluid Dynamics, Structural Mechanics, Semiconductor Modelling, Electromagnetics and CAD/CAM. Multidisciplinary applications of these fields to critical societal and technological problems encountered in sectors like Aerospace, Car and Ship Industry, Electronics, Energy, Finance, Chemistry, Medicine, Biosciences, Environmental sciences are of particular interest. The intent is to exchange information and to promote the transfer between the research community and industry consistent with the development and applications of computational methods in science and technology.

Book proposals are welcome at
Eugenio Oñate
International Center for Numerical Methods in Engineering (CIMNE)
Technical University of Catalunya (UPC)
Edificio C-1, Campus Norte UPC Gran Capitán
s/n08034 Barcelona, Spain
onate@cimne.upc.edu
www.cimne.com
or contact the publisher, Dr. Mayra Castro, mayra.castro@springer.com

Indexed in SCOPUS, Google Scholar and SpringerLink.

More information about this series at http://www.springer.com/series/6899

António Gaspar-Cunha · Jacques Periaux ·
Kyriakos C. Giannakoglou · Nicolas R. Gauger ·
Domenico Quagliarella · David Greiner
Editors

Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences

Springer

Editors

António Gaspar-Cunha
Department of Polymer Engineering
University of Minho
Guimaraes, Portugal

Kyriakos C. Giannakoglou
School of Mechanical Engineering
National Technical University Athens
Athens, Greece

Domenico Quagliarella
CIRA – Centro Italiano Ricerche
Aerospaziali
Capua, Italy

Jacques Periaux
International Center for Numerical Method
Barcelona, Spain

Nicolas R. Gauger
Technische Universität Kaiserslautern
Kaiserslautern, Rheinland-Pfalz, Germany

David Greiner
University of Las Palmas de Gran Canaria
Las Palmas, Spain

ISSN 1871-3033
Computational Methods in Applied Sciences
ISBN 978-3-030-57421-5 ISBN 978-3-030-57422-2 (eBook)
https://doi.org/10.1007/978-3-030-57422-2

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book contains book chapters from selected and extended contributions presented at the EUROGEN 2019 International Conference that took place at the University of Minho at the historical town of Guimarães in the north of Portugal on September 12–14, 2019.

EUROGEN 2019 was the 13th of a series of International Conferences previously held in Las Palmas de Gran Canaria (1995), Trieste (1997), Jyväskylä (1999), Athens (2001), Barcelona (2003), Munich (2005), Jyväskylä (2007), Kracow (2009), Capua (2011), Las Palmas de Gran Canaria (2013), Glasgow (2015) and Madrid (2017) mainly focused on Evolutionary and Deterministic Computing for Industrial Applications. This conference aims at bringing together specialists from universities, research institutions and industries developing or applying evolutionary and deterministic methods in design optimization and control and emphasizing industrial and societal applications.

This series of conferences was originally launched by the EC–Strategic/Prospective Network INGENET in 1998. EUROGEN 2019 is, also, an ECCOMAS Thematic Conference also co-ordinated by the Special Interest Group (SIG) with ERCOFTAC (European Research Community on Flow, Turbulence and Combustion).

The conference included the following topics: metaheuristics and evolutionary algorithms, multi-objective evolutionary algorithms and constraint handling techniques, adjoint-based including one-shot methods, hybrid optimisation methods, high-performance computing, goal-oriented optimization for mesh and meshless methods, game strategies, surrogate models for optimisation, parallel and distributed evolutionary algorithms, multi-disciplinary optimization methods, design optimization under uncertainties, multi-criteria decision making and topology optimization.

EUROGEN 2019 beneficiated from the presentation of five keynote invited speakers: Kyriakos Giannakoglou from the National Technical University of Athens (NTUA), Greece; Maria João Alves from the University of Coimbra, Portugal; José Covas from the University of Minho, Portugal; Massimiliano Vasile from the University of Strathclyde, United Kingdom and Carlos Fonseca from the University of Coimbra, Portugal.

With a peer review selection procedure, within the 91 papers submitted from 17 countries, 29 were selected to be published in this volume, which has been organized in taking into account the following eleven (11) mini-symposia:

- Multi-fidelity, surrogate modelling and design exploration of real-world problems
- Recent Advances in Numerical Optimization and Optimal Control and its Applications
- Single and Multi-objective Bilevel Optimization
- Adjoint methods for Multi-physics, including Applications
- Design support tools in industrial and scientific applications
- EMO—Evolutionary Multi-Objective Optimization
- Optimization under Uncertainty
- Numerical simulation as a tool in product development for the industry
- Design of polymer processing equipment: numerical simulation and optimization
- Particle-based simulation
- Game Theory and Optimization: From Theory to Applications.

These above Mini-Symposia (MS) has been classified into two major components:

- Theoretical and Numerical Methods and Tools for Optimization and Control;
- Engineering Design and Societal Applications.

We express our gratitude to the keynote speakers for accepting our invitation, to all authors who submitted their research and industrial contributions, to the International Corresponding Members and to the members of the European Technical Committee and of the Scientific Programme Committee.

We acknowledge also the support of the School of Engineering of the University of Minho and the financial support of the NEWEX European Project under grant 734205, a Marie Sklodowska Curie Action, Research and Innovation Staff Exchange (RISE) project.

Finally, the editors are indebted to Nathalie Jacobs and Eugenio Onate, ECCOMAS-Springer Manager and Director of this series continuously involved in the publication of books linked to EUROGEN and at last thank the Springer Manager line for assistance and patience in compiling this volume.

Guimaraes, Portugal	António Gaspar-Cunha
Barcelona, Spain	Jacques Periaux
Kaiserslautern, Germany	Nicolas R. Gauger
Athens, Greece	Kyriakos C. Giannakoglou
Capua, Italy	Domenico Quagliarella
Las Palmas, Spain	David Greiner
March 2020	

Contents

Chapter 1
A Bi-Level Optimization Approach to Define Dynamic Tariffs with Variable Prices and Periods in the Electricity Retail Market

Inês Soares, Maria João Alves, and Carlos Henggeler Antunes

Abstract Dynamic time-differentiated pricing structures are expected to become a common practice in smart grids, bringing benefits for all stakeholders involved: grid operators, retailers and consumers. The optimization of dynamic time-of-use (ToU) tariffs by a retailer considering the consumers' response can be modeled through bi-level (BL) programming. The retailer first defines the prices for each period and the consumer then reacts by rescheduling the operation of appliances, in face of prices and comfort requirements. In this paper, we present two BL models to define dynamic tariffs, in which the goal is to determine both the price values and the periods in which they hold by considering: (i) variable periods with a maximum number of different prices and (ii) total freedom to define the periods and the corresponding prices. Both models are highly difficult to solve, mainly due to the size of the search space of the upper level (UL) problem, in which the combinations of prices and periods are determined. We describe the development of hybrid approaches considering a population metaheuristic for the UL problem and a mixed-integer linear programming (MILP) solver to address the lower level (LL) problem, in which the optimal appliance scheduling for the pricing structure is computed. The exploration of the UL search space is crucial to obtain good solutions within an acceptable computational effort.

I. Soares (✉)
INESC Coimbra, Rua Sílvio Lima, Pólo II, 3030-290 Coimbra, Portugal
e-mail: inesgsoares@gmail.com

M. J. Alves
CeBER and Faculty of Economics, University of Coimbra / INESC Coimbra, Av. Dr. Dias da Silva, 165, 3004-512 Coimbra, Portugal
e-mail: mjalves@fe.uc.pt

C. H. Antunes
Department of Electrical and Computer Engineering, INESC Coimbra, University of Coimbra (DEEC-UC), Rua Sílvio Lima, Pólo II, 3030-290 Coimbra, Portugal
e-mail: ch@deec.uc.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_1

Keywords Bi-level optimization · Hybrid algorithms · Metaheuristics · Mixed-integer linear programming · Dynamic tariffs · Pricing problems · Electricity retail market

1.1 Introduction

The problem of optimizing dynamic ToU tariffs involves two decision makers: the retailer is the leader defining a pricing strategy that maximizes his profit and the consumers are the follower seeking to satisfy their energy needs at minimal cost. The retailer decides first, but the consumers' reaction will affect the retailer's profit.

BL programming encompasses a leader–follower hierarchical structure, in which the leader takes into account in his decision process the reaction of the follower. In the framework of a pricing problem, the optimal decision for the leader's problem depends on the follower's reaction to the possible price structures set by the leader, i.e. the optimal solution to the consumers' problem for each instantiation of the leader's decision variables (the prices).

BL problems are difficult to solve and even the linear BL problem is NP-hard. The existence of discrete variables, particularly in the LL problem, further aggravates the difficulty of solving the BL problem. This is the case of the models we deal with in this work, which include continuous variables at the UL (prices) and continuous and binary variables at the LL (to model the operation of the consumer's appliances). Thus, it is of utmost importance to consider the structure and features of the BL model to develop computationally efficient solution approaches.

In the study by Alves et al. [1], the interaction between an electricity retailer and residential consumers is modeled as a BL programming problem. A metaheuristic to address the retailer's problem (maximizing profits) is combined with an exact solver to deal with the consumer's problem (minimizing costs). The UL problem is tackled by a genetic algorithm (GA), in which the decision variables are the prices to be established in pre-determined periods (ToU tariff). The LL problem is solved by an exact MILP solver, in which the decision variables are associated with appliance operation. In that work only shiftable loads were considered: loads for which the start time of the operation cycle, which cannot be interrupted, should be determined. The model has been extended by considering also interruptible loads: loads whose operation can be interrupted and the periods of energy supply should be determined for a given amount of energy [2].

In the current work, we are particularly interested in addressing the pricing problem of defining dynamic ToU tariffs when not only the prices are decision variables of the UL problem, but also the periods in which those prices are applied. As far as we know, a pricing problem with these features was never studied in the literature.

In the context of electricity retail markets, several models to study the interaction between retailers and consumers have been developed recently. Nevertheless, in all the models found in the literature, the tariff periods are pre-determined and just the prices are subject to the decision of the retailer. The studies [3–7] are examples of pricing problems with these features. They are mainly focused on determining just the price for each tariff period, which usually coincides with the time discretization of the planning period (e.g., one day) set as one hour. These studies consider loads supplied with a given amount of energy for service completion, not accounting for actual appliance operation cycles. This relevant issue has been duly taken into account in recent works [1, 2, 9], where physical information related with load operation/control is considered [8], enabling a more accurate representation of the consumer's energy management problem and thus transmitting to the retailer a more realistic reaction of the consumer. However, in these works pre-determined pricing periods are also considered.

Pricing problem optimization approaches have also been widely developed in other research areas out of the electricity context (for instance, see the work [10] for a review). However, similarly to what happens with studies published in the context of electricity retail markets, those works do not consider the definition of pricing periods as variables of the problem, but only the prices.

In this work, we formulate the interaction between an electricity retailer and a cluster of consumers with similar consumption patterns as a BL optimization model. The maximization of the retailer's profit is the UL objective function and the minimization of the consumer' costs is the LL objective function. At the UL, the retailer should determine both tariff periods and the corresponding prices: (i) when a maximum number of different prices for the entire planning period is imposed (e.g., due to regulatory requirements) and ii) when the retailer has total freedom to define the tariff periods and the corresponding prices. Both BL problems are highly challenging due to the size of the UL search space resulting from the combination of periods and prices. Two hybrid population-based approaches are proposed, each one combining a metaheuristic—GA or Particle Swarm Optimization (PSO) algorithms—for the UL problem and an exact MILP solver to solve the LL problem for each instantiation of the UL decision variables. Enhancing the exploration capability of the UL search space is of great importance to improve the retailer's profit within an acceptable computational time. Therefore, two schemes of an adaptive mutation operator were also designed in both metaheuristics.

The main contributions of this paper are novel models to optimize periods and prices and two hybrid BL approaches. These approaches make the most of the structure of the problem to determine dynamic ToU tariffs where both periods and prices are the decisions in the UL problem.

The manuscript is organized as follows. In Sect. 1.2, the BL formulation to model the interaction between the retailer and the consumers in the retail electricity market is presented. Section 1.3 describes two hybrid population-based algorithmic approaches, combining either a GA or a PSO with a MILP exact solver. The results are presented and discussed in Sect. 1.4. The conclusions are drawn in Sect. 1.5.

1.2 Bi-Level Optimization Model

In a BL optimization model, two problems are hierarchically involved. The UL refers to the leader's problem and the LL arises as a constraint of the UL problem, referring to the follower's problem. In the BL model presented in the current work, the retailer is the leader and the consumer is the follower. The retailer first sets the tariff periods and the corresponding prices and then the consumer optimizes the operation of the appliances in face of that tariff structure. The goal of the leader is to maximize profits, while the follower aims to minimize the electricity bill considering comfort requirements.

The consumer's problem encompasses two types of appliances with different physical features and type of control, in addition to a base (uncontrollable) load: shiftable appliances (whose operation cycle cannot be interrupted once initiated) and interruptible loads (for which the energy supply can be interrupted provided that the required amount of energy is supplied). In the following, the BL model is described in detail.

The planning period T is discretized into a time unit of length h (hour, minute, quarter of hour, etc.), such that $\mathrm{T} = \{1, \ldots, T\}$ with T being the number of time units. The UL decision variables are the electricity prices x_i (in €/kWh) to be charged by the retailer to the consumers in each periodP_i.The$P_i \subset \mathrm{T}$, $i \in \{1, \ldots, I\}$, are pricing periods in which the planning period T is divided into, such that $\bigcup_{i=1}^{I} P_i = \mathrm{T}$. The periods P_i may be constant or variable for the optimization problem. Three cases may occur: (1) the retailer defines a priori I disjoint periods of prices, each with dimension $\overline{P_i}$ (as was considered in our previous studies [1, 2, 9]); (2) the pricing periods can coincide with the time units $t \in \mathrm{T}$ (thus, $I = T$), thus allowing for total freedom in establishing the combinations periods-prices; (3) a maximum number I of different prices for the whole planning period T can be imposed; the I periods P_i in which each price x_i holds also result from the optimization model. In this work, the second and the third cases are studied (designated as *Free BL T* and *Free BL I*, respectively).

The electricity prices x_i set by the retailer are limited to minimum and maximum values, respectively $\underline{x}$ and $\overline{x}$, (constraints (2) and (3) in the BL model presented below). An average electricity price x^{AVG} for the whole planning period T is also imposed (constraint (4)) to account for competition in the electricity retail market, otherwise the retailer would establish the prices at the maximum value allowed.

BL model (with pre-defined periods P_i)

$$\max_{x} F = \sum_{i=1}^{I} \sum_{t \in P_i} x_i \left(b_t + \sum_{j=1}^{J} p_{jt} + \sum_{k=1}^{K} q_{kt} \right) + \sum_{l=1}^{L} e_l u_l - \sum_{t=1}^{T} \pi_t \left(b_t + \sum_{j=1}^{J} p_{jt} + \sum_{k=1}^{K} q_{kt} \right) \tag{1}$$

s.t.

$$x_i \leq \overline{x}\,, \quad i = 1, \ldots, I \tag{2}$$

$$x_i \geq \underline{x}\,, \quad i = 1, \ldots, I \tag{3}$$

$$\frac{1}{T} \sum_{i=1}^{I} \overline{P_i} x_i = x^{AVG} \tag{4}$$

$$\min_{p,w,q,v,u} \ f = \sum_{i=1}^{I} \sum_{t \in P_i} x_i \left(b_t + \sum_{j=1}^{J} p_{jt} + \sum_{k=1}^{K} q_{kt} \right) + \sum_{l=1}^{L} e_l u_l \tag{5}$$

s. t.

$$p_{jt} = \sum_{r=1}^{d_j} f_{jr} w_{jrt}\,, \quad j = 1, \ldots, J, \ \ t = T_{1_j}, \ldots, T_{2_j} \tag{6}$$

$$p_{jt} = 0\,, \quad j = 1, \ldots, J, \ \ t < T_{1_j} \vee t > T_{2_j} \tag{7}$$

$$\sum_{r=1}^{d_j} w_{jrt} \leq 1\,, \quad j = 1, \ldots, J, \ \ t = T_{1_j}, \ldots, T_{2_j} \tag{8}$$

$$w_{j(r+1)(t+1)} \geq w_{jrt}\,, \quad j = 1, \ldots, J, \ \ r = 1, \ldots, (d_j - 1), \ \ t = T_{1_j}, \ldots, \left(T_{2_j} - 1 \right) \tag{9}$$

$$\sum_{t=T_{1_j}}^{T_{2_j}} w_{jrt} = 1\,, \quad j = 1, \ldots, J, \ \ r = 1, \ldots, d_j \tag{10}$$

$$\sum_{t=T_{1_j}}^{T_{2_j} - d_j + 1} w_{j1t} \geq 1\,, \quad j = 1, \ldots, J \tag{11}$$

$$w_{jrt} \in \{0,1\}\,, \quad j = 1, \ldots, J, \ \ r = 1, \ldots, d_j, \ \ t = T_{1_j}, \ldots, T_{2_j} \tag{12}$$

$$p_{jt} \geq 0\,, \quad j = 1, \ldots, J, \ \ t = 1, \ldots, T \tag{13}$$

$$q_{kt} = v_{kt} Q_k\,, \quad k = 1, \ldots, K, \ \ t = T_{1_k}, \ldots, T_{2_k} \tag{14}$$

$$q_{kt} = 0\,, \quad k = 1, \ldots, K, \ \ t < T_{1_k} \vee t > T_{2_k} \tag{15}$$

$$\sum_{t=T_{1_k}}^{T_{2_k}} q_{kt} = E_k\,, \quad k = 1, \ldots, K \tag{16}$$

$$v_{kt} \in \{0,1\}\,, \quad k = 1, \ldots, K, t = T_{1_k}, \ldots, T_{2_k} \tag{17}$$

$$q_{kt} \geq 0\,, \quad k = 1, \ldots, K\,, \ \ t = 1, \ldots, T \tag{18}$$

$$\sum_{l=1}^{L} u_l = 1 \tag{19}$$

$$b_t + \sum_{j=1}^{J} p_{jt} + \sum_{k=1}^{K} q_{kt} \leq \sum_{l=1}^{L} P_l^{Cont} u_l\,, t = 1, \ldots, T \tag{20}$$

$$u_l \in \{0,1\}\,, \quad l = 1, \ldots, L \tag{21}$$

The UL objective function (Eq. 1) in this BL model with pre-defined periods) relates to the maximization of the retailer’s profit, being defined as the difference between the revenue with the sale of energy to consumers and the cost of buying energy in the wholesale market. Coefficients π_t are the prices of energy incurred by the retailer at time $t \in$ T.

The LL objective function (Eq. 5) corresponds to the minimization of the electricity bill (which matches the retailer’s revenue), being defined as the sum of the cost of the energy consumed by controllable and uncontrollable loads and the contracted power for the whole planning period T. The retailer defines L levels of power demand, P_l^{Cont} (in kW), $l \in \{1, \ldots, L\}$, and the consumer pays e_l (in €) for the power level *l* corresponding to the peak.

For each time t of the planning period T, the BL model considers a (non-controllable) base load b_t (in kW), J shiftable appliances, such that each load $j \in \{1, \ldots, J\}$ requires from the grid a power p_{jt} (in kW), and K interruptible appliances, each requiring from the grid a power q_{kt} (in kW), $k \in \{1, \ldots, K\}$.

For controllable loads, the consumer should specify the comfort time slots in which each load should operate, according to his preferences and routines, namely $T_j = \left[T_{1_j}, T_{2_j} \right] \subseteq$ T for each shiftable load j, $j \in \{1, \ldots, J\}$, and $T_k = \left[T_{1_k}, T_{2_k} \right] \subseteq$ T for each interruptible load k, $k \in \{1, \ldots, K\}$. The power requested by each shiftable load j at stage $r \in \left\{ 1, \ldots, d_j \right\}$ of its operation cycle is f_{jr} (in kW), being d_j the duration of the operation cycle. For each interruptible load k, the power requested at each time is Q_k (in kW) and E_k is the total energy required.

The LL decision variables for each shiftable load j are w_{jrt}, which are binary variables specifying whether appliance j is "on" or "off" at each time $t \in T_j$ and at each stage r of its operation cycle. These binary variables are used to define the auxiliary continuous variables p_{jt} that appear in the UL and LL objective functions. The LL decision variables for each interruptible load k are the binary variables v_{kt}, which specify whether load k is "on" or "off" at each time $t \in T_k$. The variables v_{kt} define the auxiliary variables q_{kt} (equal to 0 or Q_k), which also appear in objective functions (1) and (5). Constraints (6)–(13) model the operation of shiftable loads and the set of constraints (14)–(18) models the operation of interruptible appliances.

Constraints (6) define p_{jt} when shiftable load $j \in \{1, \ldots, J\}$ is allowed to operate (i.e. for $t \in T_j$) and (7) impose $p_{jt} = 0$ outside the comfort time slot of load j. Constraints (8) guarantee that, at time t of the planning period, each shiftable load j is either "off" or "on" at only one stage r of its operation cycle. Constraints (9) guarantee that if load j is "on" at time t and at stage $r < d_j$ of its operation cycle, then it must also be "on" at time $t + 1$ and stage $r + 1$. Constraints (10) ensure that each load j operates exactly once at stage r and this should occur for $t \in T_j$ (i.e. when load j is allowed to operate). Constraints (11) ensure that load j starts its working cycle within its allowed comfort time slot, i.e. at most at time $T_{2_j} - d_j + 1$, thus assuring that it never finishes later than T_{2_j}. Therefore, constraints (9–11) ensure that each shiftable appliance j operates precisely d_j consecutive time units, thus forcing the LL decision variables w_{jrt} to be zero whenever appliance j is "off".

Constraints (14) define variables q_{kt} when interruptible load $k \in \{1, \ldots, K\}$ is allowed to operate (i.e. for $t \in T_k$), and constraints (15) set $q_{kt} = 0$ outside the comfort time slot of load k. Constraints (16) ensure that the total amount of energy required from the grid by interruptible load k within the comfort time slot is E_k.

The LL binary decision variables for the power component are u_l, which specify the peak power level $l \in \{1, \ldots, L\}$, where $u_l = 1$ means that the consumer should pay for the power price corresponding to the l power level. Constraint (19) ensures that a single power price should be charged to the consumer in the whole planning period. Constraints (20) guarantee that the total power required from the grid at each time $t \in$ T should satisfy the operation of all loads.

The BL model just described is for the case 1, in which the I periods P_i are constant and pre-defined.

Let us denote by $R_t = b_t + \sum_{j=1}^{J} p_{jt} + \sum_{k=1}^{K} q_{kt}$ the total power required by all loads at time t and $S = \sum_{l=1}^{L} e_l u_l$ the power cost. Therefore, the UL and LL objective functions (1) and (5) may be written for short as $F = \sum_{i=1}^{I} \sum_{t \in P_i} (x_i R_t) + S - \sum_{t=1}^{T} (\pi_t R_t)$ and $f = \sum_{i=1}^{I} \sum_{t \in P_i} (x_i R_t) + S$, respectively.

In the case 2 (*Free BL T*), in which a different price can be defined for each time unit, the UL problem (1)–(4) is replaced by (22)–(24), while keeping the LL problem as before with the LL objective function being $f = \sum_{t=1}^{T} (x_t R_t) + S$.

$$\max_{x} F = \sum_{t=1}^{T}(x_t R_t) + S - \sum_{t=1}^{T}(\pi_t R_t)$$

$$\text{s.t.} \tag{22}$$

$$\underline{x} \leq x_t \leq \overline{x}, t = 1, \ldots, T \tag{23}$$

$$\frac{1}{T}\sum_{t=1}^{T} x_t = x^{AVG} \tag{24}$$

For the case 3 (*Free BL I*), the optimization problem sets I prices x_i, as in case 1, but it must also specify each period P_i, which can result as the union of discontinuous intervals. To formulate this model as a mathematical program, additional binary variables controlled in the UL problem are required: $z_{it} \in \{0, 1\}, i = 1, \ldots, I$, $t = 1, \ldots, T$, which specify whether price x_i holds or not at time t. The UL problem of the *Free BL I* model can be formulated as (25)–(29). The LL problem is the same as before with the LL objective function being $f = \sum_{i=1}^{I}\sum_{t=1}^{T}(z_{it} x_i R_t) + S$.

$$\max_{x} F = \sum_{i=1}^{I}\sum_{t=1}^{T}(z_{it} x_i R_t) + S - \sum_{t=1}^{T}(\pi_t R_t)$$

$$\text{s.t.} \tag{25}$$

$$\underline{x} \leq x_i \leq \overline{x}, i = 1, \ldots, I \tag{26}$$

$$\frac{1}{T}\sum\nolimits_{i=1}^{I}\sum_{t=1}^{T} z_{it} x_i = x^{AVG} \tag{27}$$

$$\sum_{i=1}^{I} z_{it} = 1, \ t = 1, \ \ldots, T \tag{28}$$

$$z_{it} \in \{0, 1\}, i = 1, \ldots, I, t = 1, \ldots, T \tag{29}$$

The constraints (28) ensure that exactly one of the I prices holds at each time t.

Since the UL problem is dealt with a metaheuristic, the binary variables z_{it} and the constraints (28) are not explicitly considered in the solution approach developed to tackle this problem. A convenient representation of the solution takes care of these issues, ensuring that at most I different prices x_i are determined, also defining the intervals of the time associated with each price.

1.3 Hybrid Approaches

In this work, two BL hybrid approaches are proposed to address the two cases of the BL model in which P_i are free. The *Free BL I* problem considers variable tariff periods with a maximum number I of different prices. In the *Free BL T* problem, the number of different prices may be as many as the time units the planning period is discretized into. The two hybrid BL approaches combine a metaheuristic to explore the UL solution space (GA and PSO algorithms) with an exact MILP solver (*Cplex*) to solve the LL problem. These approaches are presented below for each BL model.

1.3.1 Free BL I

In this section, the problem to define dynamic ToU tariffs with a maximum number I of different prices for the whole planning period T is addressed. The global framework of the hybrid algorithm is first presented; then, the characteristics of the two population-based approaches (GA and PSO) developed for the UL search are described in detail.

The algorithm uses two structures to encode a solution: a vector of I prices $x = (x_1, \dots, x_I)$ and a vector of T labels $\ell = (\ell_1, \dots, \ell_T)$ that indicate the index of the price (from 1 to I) that holds at each time $t = 1, \dots, T$.

The algorithm starts by creating an initial population of N prices and N labels, respectively, $x^h = \left(x_1^h, \dots, x_I^h\right)$ and $\ell^h = \left(\ell_1^h, \dots, \ell_T^h\right)$, $h = 1, \dots, N$. Together, these two vectors define each individual $\wp^h = \left(x^h_{\ell_1^h}, \dots, x^h_{\ell_T^h}\right)$, which represents the electricity prices set by the retailer with at most I different price values for the whole planning period T. Thus, the UL objective function (25) becomes $\sum_{t=1}^{T}\left(x^h_{\ell_t^h} R_t\right) + S - \sum_{t=1}^{T}(\pi_t R_t)$.

For each feasible electricity price vector $\wp^h$ defined at the UL, a LL solution y^h is determined by *Cplex* (y^h denotes the values of all decision variables of the LL problem). The Pseudocode 1 summarizes the steps of the hybrid *Free BL I* approach.

In Step 2.1. of the Pseudocode 1, if the price solution $\wp^h$ resulting from decoding the prices x^h, $h \in \{1, \dots, N\}$, using the time information in the label vector ℓ^h, does not satisfy the UL constraints, then it should be repaired according to the repair routine described in Soares et al. 2019 [2].

The output of the hybrid BL approach is the solution $(\wp, y)$ that gives the highest retailer's profit $F(\wp, y)$ after performing the G iterations.

Pseudocode 1 – Hybrid Free BL I approach

```
1. Create the initial population Pop of size N, $h = 1, \dots, N$:
   N prices: $x^h = (x_1^h, \dots, x_I^h)$
   N labels: $\ell^h = (\ell_1^h, \dots, \ell_T^h)$
   N electricity price individuals set by the retailer: $\wp^h = \left(x_{\ell_1^h}^h, \dots, x_{\ell_T^h}^h\right)$
2. Repeat:
   2.1. Repair each price solution $x^h$ (respectively, $\wp^h$), $h \in \{1, \dots, N\}$, of Pop to
        guarantee that the UL constraints are satisfied;
   2.2. Solve the LL problem using the Cplex solver for each $\wp^h$, $h \in \{1, \dots, N\}$,
        in Pop to obtain the corresponding $y^h$;
   2.3. Determine $F(\wp^h, y^h)$ for all $(\wp^h, y^h)$, $h \in \{1, \dots, N\}$;
   2.4. Update $x^h$ and $\ell^h$, $h \in \{1, \dots, N\}$, of Pop according to the population-based
        algorithm (GA or PSO as described below);
   2.5. Update solution $\wp^h$, $h \in \{1, \dots, N\}$, according to the corresponding
        updated vectors $x^h$ and $\ell^h$.
   Until G iterations are performed;
Output: $(\wp, y)$ with the best $F(\wp, y)$.
```

After creating the initial population of prices and labels and evaluating their fitness $F(\wp^h, y^h)$, a new population is iteratively produced.

In the following, the evolution of the population at each iteration is described for each population-based algorithm.

1.3.1.1 GA

In each new generation, the GA starts by creating the offspring population: one parent is selected using binary tournament (decided by the F value) and the other parent is randomly selected. The parent solutions are then subject to crossover to generate two descendants, having both solutions the same probability of being the first or the second parent. If $\ell^{'h}$ and $\ell^{''h}$ are the first and the second parent solutions of labels, the one-point crossover operator produces the children $\ell^{'c^h} = \left(\ell^{'h}{}_1, \dots, \ell^{'h}{}_\varsigma, \ell^{''h}{}_{\varsigma+1}, \dots, \ell^{''h}{}_T\right)$ and $\ell^{''c^h} = \left(\ell^{''h}{}_1, \dots, \ell^{''h}{}_\varsigma, \ell^{'h}{}_{\varsigma+1}, \dots, \ell^{'h}{}_T\right)$, $\varsigma \in \{2, \dots, T-1\}$. Being $x^{'h}$ and $x^{''h}$ the first and second parent solutions of prices, the children components $x_i^{'c^h}$ and $x_i^{''c^h}$, $i \in \{1, \dots, I\}$, respectively, are determined by using the geometric crossover operator, according to the following expressions:

$$x_i^{'c^h} = \lambda_1 x_i^{'h} + (1 - \lambda_1) x_i^{''h} \text{and } x_i^{''c^h} = \lambda_2 x_i^{'h} + (1 - \lambda_2) x_i^{''h}$$

where λ_1 and λ_2 are uniform random numbers in the interval [0, 1].

Then, a mutation operator with a given probability p_m is applied to the children labels $\ell^{'c^h}$ and $\ell^{''c^h}$. Two kinds of mutation were considered for the labels: random mutation—in which, each position $\ell_t^{'c^h}$ and $\ell_t^{''c^h}$, $t \in \{1, \dots, T\}$, can be randomly changed to a different value in $\{1, \ldots, I\}$—and adjacent mutation—in which a comparison with positions immediately before and after is made and only a mutation to such values is allowed unless those values are the same (in this case, the random mutation is applied). Similarly, a mutation operator with a given probability is applied to both $x^{'c^h}$ and $x^{''c^h}$: for each position $x_i^{'c^h}$ and $x_i^{''c^h}$, $i \in \{1, \ldots, I\}$ of $x^{'c^h}$ and $x^{''c^h}$, respectively, perturbations γ_1 and γ_2 are randomly generated in the range $\left[-\delta\left(\overline{x} - \underline{x}\right), \delta\left(\overline{x} - \underline{x}\right)\right]$ and added to $x_i^{'c^h}$ and $x_i^{''c^h}$, i.e. $x_i^{'c^h} \leftarrow x_i^{'c^h} + \gamma_1$ and $x_i^{''c^h} \leftarrow x_i^{''c^h} + \gamma_2$ (δ is a pre-defined constant).

Each run of the algorithm is initialized with a mutation probability p_m^0. If the value of the highest retailer's profit (F^{best}) does not improve over a predefined number $G^{'}$ of consecutive iterations, then the exploration capability is enhanced by increasing the mutation probability to $p_m^1 > p_m^0$. The value of the mutation probability decreases back to the previous value, p_m^0, when F^{best} changes above a given threshold value τ. For that purpose, it has been considered that the change in F^{best} does not lead to a change in the mutation probability if the inequality $\frac{F^{best^q} - F^{best^{q-1}}}{F^{best^q}} < \tau$ is satisfied, $q \in \{2, \ldots, G\}$.

If the child price solutions do not satisfy the UL constraints, they must be repaired using the repair routine.

This process is repeated until N children are generated, which constitute the offspring population. For each child prices x^{c^h} with a corresponding child labels ℓ^{c^h}, i.e. for each $\wp^{c^h}$, the LL problem is solved to compute the corresponding optimal LL solution —y^{c^h}. Then, the UL objective function is evaluated for each solution$\left(\wp^{c^h}, y^{c^h}\right)$:$F\left(\wp^{c^h}, y^{c^h}\right)$.

To generate the new population for the next iteration, first the individual with the best F value, either from the current or the offspring population, is preserved from one generation to the next. The remaining $N - 1$ individuals are selected by binary tournament without replacement between two individuals randomly selected from the merged current and offspring populations.

1.3.1.2 PSO

After randomly generating the initial population of labels and prices (the swarm of particles in PSO), the algorithm iteratively moves each particle toward: the best position visited by each particle—*personal best,* x^{best} and ℓ^{best} for prices and labels, respectively; and the best known position of the entire swarm—*global best*, g^{xbest} and $g^{\ell best}$ for prices and labels, respectively.

In each iteration $q \in \{1, \ldots, G\}$, the position of each particle is updated according to its own velocity vector. For that purpose, for each coordinate $i \in \{1, \ldots, I\}$ of each x^h and each coordinate $t \in \{1, \ldots, T\}$ of each ℓ^h, the corresponding velocity components ${v_i}^h$ and ${v_t}^h$ for the next iteration q are given by the following equations:

$${v_i}^{h^q} = \eta {v_i}^{h^{q-1}} + r_1 C_1\left(x_i^{best} - {x_i}^{h^{q-1}}\right) + r_2 C_2\left(g_i^{xbest} - {x_i}^{h^{q-1}}\right)$$

$$v_t{}^{hq} = \eta v_t{}^{hq-1} + r_1 C_1 \left(\ell_t^{best} - \ell_t{}^{hq-1}\right) + r_2 C_2 \left(g_t^{\ell best} - \ell_t{}^{hq-1}\right)$$

where η is the inertia weight, C_1 and C_2 are the cognitive and social parameters, r_1 and r_2 are random numbers in the interval [0,1]. The new positions of x^h and ℓ^h are then given by:

$$x^{hq} = x^{hq-1} + v^{hq} \text{and } \ell^{hq} = \ell^{hq-1} + v^{hq}$$

Each label vector is then repaired to guarantee that any component $\ell_t{}^h, t \in \{1, \ldots, T\}$, falls into the set $\{1, \ldots, I\}$. Firstly, if $\ell_t{}^h$ is out of the bounds, then it is pushed to the closest bound (minimum 1 or maximum I). Otherwise, the value of $\ell_t{}^h$ is rounded to the closest integer.

Similarly to the GA, in the PSO a mutation operator with a given probability is also applied to each pair of vectors x^h and ℓ^h (as described in the GA—Sect. 3.1.1). Additionally, an adaptive mutation to induce some turbulence in the swarm to enhance the exploration capability is considered (in the same way as in the GA algorithm), if no improvement of F^{best} is verified over a predefined number G' of consecutive iterations.

If the price solutions do not satisfy the UL constraints, then they are repaired according to the repair routine. Each pair x^h and ℓ^h is then combined to form the solution $\wp^h$ and the LL problem is solved, computing the corresponding optimal y^h; then, the UL objective function is evaluated—$F\left(\wp^h, y^h\right)$. Whenever better solutions are found in each iteration according to F values, the individual best, x^{best} and ℓ^{best}, and the global best, g^{xbest} and $g^{\ell best}$, are updated.

1.3.2 *Free BL T*

The model designated as *Free BL T* involves the definition of completely free prices, i.e. the determination of electricity prices for the whole planning period T without restrictions for the number of tariff periods and prices. Thus, at most, we can get T tariff periods with T different prices. The GA and PSO metaheuristics developed for this model are described below.

The proposed population-based approaches begin by creating an initial population of N individuals of electricity prices set by the retailer for the entire planning period T: $x^h = \left(x_1^h, \ldots, x_T^h\right)$, $h = 1, \ldots, N$. A LL solution y^h is computed for each x^h, then evaluated by the UL objective function, $F\left(x^h, y^h\right)$. After that, a new population is iteratively produced until reaching a given number of G iterations. At the end, the output of each hybrid population-based approach is the final solution (x, y) with the highest $F(x, y)$ value.

Note that the labels used in the *Free BL I* model to enforce a maximum number of different prices are not necessary in the *Free BL T* model.

The Pseudocode 2 summarizes the steps of the hybrid *Free BL T* approach.

Pseudocode 2 - Hybrid Free BL T approach

```
1. Create the initial population Pop of size N, $h = 1, \dots, N$: $x^h = (x_1^h, \dots, x_T^h)$
2. Repeat:
    2.1. Repair each individual price solution $x^h$, $h \in \{1, \dots, N\}$, of Pop to ensure
         that the UL constraints are satisfied;
    2.2. Solve the LL problem for each $x^h$, $h \in \{1, \dots, N\}$, in Pop, using the Cplex
         solver to obtain the corresponding $y^h$;
    2.3. Determine $F(x^h, y^h)$ for all $(x^h, y^h)$, $h \in \{1, \dots, N\}$;
    2.4. Update each solution $x^h$, $h \in \{1, \dots, N\}$, of Pop according to the population-
         based algorithm;
    Until G iterations are performed;
Output: $(x, y)$ with the best $F(x, y)$.
```

According to the step 2.1. of the Pseudocode 2, if a solution x^h, $h \in \{1, \dots, N\}$, does not satisfy the UL constraints of the BL model described in Sect. 1.2, then it should be repaired using the repair routine.

1.3.2.1 Upper Level Population-Based Search

The creation of a new population at each iteration $q \in \{1, \dots, G\}$ depends on the population-based algorithm that is applied, GA or PSO. The steps followed in this approach are quite similar to those previously described for the *Free BL I*.

In the *Free BL T*, only the electricity prices vectors $x^h, h \in \{1, \dots, N\}$, are updated in each iteration of the algorithm. The dimension of each solution x^h coincides with the number of time units, $x^h = \left(x_1^h, \dots, x_T^h\right)$.

The creation of each new solution x^h in each iteration q follows the steps of each metaheuristic approach described in Sects. 3.1.1. and 3.1.2, respectively for the GA and the PSO.

1.4 Experimental Results and Discussion

In this section, the problem data and the parameters of the two population-based algorithms developed are described. Also, the results obtained are presented and analyzed.

In the experiments performed, a planning period of 24 h split in time units of 15 min was considered. This generates a planning period with $T = 96$ time units, $\mathrm{T} = \{1, \dots, 96\}$. Each time unit $t \in \mathrm{T}$ represents a quarter-hour, i.e. $h = \frac{1}{4}$h.

The electricity prices ($\pi_t, t \in \{1, \dots, 96\}$, in €/kWh) that the retailers pay in the spot market can be seen in the Supplementary Material of Soares et al. 2019 [2]—Table SM-1. Regarding the electricity prices charged to the consumers by the retailer, the following minimum and maximum electricity prices were considered in the current study: $\underline{x} = 0.08$ €/kWh and $\overline{x} = 0.35$ €/kWh, respectively. The average electricity price was set to $x^{AVG} = 0.18$ €/kWh. Detailed data regarding the power cost component is displayed in Table SM-3 of Soares et al. 2019 [2]; $L = 9$ power levels were considered in this work.

Regarding the consumers' problem, a total of five controllable appliances were considered: $J = 3$ shiftable loads—dishwasher (DW), laundry machine (LM) and clothes dryer (CD); $K = 2$ interruptible loads—electric vehicle (EV) and electric water heater (EWH). The information associated with the controllable loads, the comfort time slots allowed for the operation of each load and the (non-controllable) base load is displayed in Tables SM-4 to SM-6 (Supplementary Material of Soares et al. 2019 [2]). For the dataset we have used, the dimension of the LL problem, which must be solved for each UL solution, is: 559 binary variables, 141 continuous variables, 578 inequality constraints and 161 equality constraints.

In both hybrid population-based algorithms, $G^{'} = 5$ was considered as the number of consecutive iterations without improvement of F^{best} that leads to change the mutation probability, p_m. The probability values adopted for the adaptive mutation were $p_m^0 = 0.05$ and $p_m^1 = 0.1$. The parameter $\tau = 0.001$ was considered as the threshold value to induce some turbulence in the UL search space and enhance the exploration capability. In the mutation process, the parameter $\delta = 0.2$ was considered. In the PSO algorithm, the learning parameters in both cognitive and social components were kept equal, $C_1 = C_2 = 2$, and the inertia parameter was set to $\eta = 0.2$.

In the *Free BL I* model, a limit of $I = 6$ different electricity prices for the entire planning period was imposed. Furthermore, it was also imposed that the same electricity price prevails over two adjacent time units (i.e., for half hour), that is $\wp_1 = \wp_2, \wp_3 = \wp_4, \ldots \wp_{95} = \wp_{96}$. This assumption was considered in the implementation of the algorithm of the *Free BL I* model.

In the computational experiments, the hybrid BL algorithms were run 10 independent times, each involving $G = 200$ iterations and $N = 40$ individual. Thus, a total of 8000 MILP problems were solved. Each instantiation of the LL problem is solved to optimality in less than 0.13 s. Each complete run took approximately 17 min on average.

All parameters used in the current study were set after experimentation, reflecting the quality of results obtained vs. the computational effort.

The algorithm was written in R language and all runs were carried out in a computer with an Intel Xeon Gold 6138 CPU@3.7 GHz and 320 GB RAM.

1.4.1 Results

The two hybrid approaches based on GA and PSO algorithms were tested with the two BL models: *Free BL I* (considering two types of mutation—random and adjacent—for the price label vectors) and *Free BL T*. The information about the best solutions in the 10 independent runs, i.e. the statistics of the solutions with the highest value of F, is displayed in Tables 1.1 and 1.2 for the *Free BL I* and *Free BL T*, respectively.

Table 1.1 Statistics of F^{best} in the 10 runs for the *Free BL I* model

	GA		PSO	
	Random Mutation	Adjacent Mutation	Random Mutation	Adjacent Mutation
Maximum	5.8770	*6.0911*	5.9855	*6.1177*
Minimum	5.4291	*5.8060*	4.6647	*5.1509*
Average	5.6038	*6.0176*	5.5925	*5.9441*
Standard Deviation	0.1397	*0.0989*	0.3630	*0.2943*
Median	5.5852	*6.0586*	5.6837	*6.0705*

Table 1.2 Statistics of F^{best} in the 10 runs for the *Free BL T* model

	GA	PSO
Maximum	5.2880	5.8554
Minimum	5.0696	5.7044
Average	5.1589	5.7962
Standard Deviation	0.0610	0.0503
Median	5.1547	5.8051

The maximum, minimum, average, standard deviation and median of F^{best} obtained with the GA and PSO algorithms over the 10 runs are presented.

The information displayed in Table 1.1 for the *Free BL I* model shows that the best results are always obtained when the adjacent mutation is applied to the price labels vectors (values in italics). Furthermore, the PSO approach attained better (underlined values) maximum and median values of F^{best} than the GA. The GA approach overcomes the PSO in the minimum, average and standard deviation values of F^{best} for the 10 runs.

Regarding the *Free BL T* model, Table 1.2 shows that the PSO algorithm always reached the best solutions (underlined values). In addition to obtaining a higher maximum value of F^{best}, the PSO approach further obtained minimum, average and median values higher than the maximum value obtained with the GA algorithm.

In general, the results obtained for both free BL models reveal that the PSO algorithm reaches better solutions than those reached by the GA.

Figure 1.1 shows the electricity prices and the power requested by the consumer during the planning period T in the best solution obtained with the PSO algorithm for the *Free BL I* model with the adjacent mutation operator applied to the price label vectors (solution with profit 6.1177€).

Figure 1.1 shows that four different electricity prices were defined for the entire planning period, with higher prices for the first half of the planning period T (in which the amount of electricity required by the consumer is higher, mainly due to the charging of the electric vehicle). The values of the four electricity prices in this

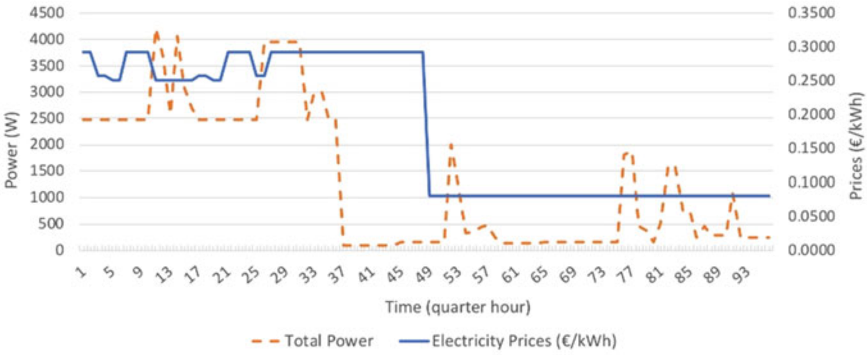


Fig. 1.1 Best solution obtained by PSO for the *Free BL I* model with adjacent mutation in the price labels vectors

solution are: $x_1 = 0.2928$ €/kWh, $x_2 = 0.2568$ €/kWh, $x_3 = 0.2504$ €/kWh and $x_4 = 0.0800$ €/kWh.

In general, the results show that the current version of the algorithmic approaches are not able to provide good quality solutions for the model with higher degrees of freedom (*Free BL T*), mainly due to the difficulties inherent to the exploration of the UL solution search space resulting from the combinations of periods and prices. The free BL *T* model gives the retailer a total freedom of choosing the tariff periods and the corresponding prices values. Therefore, the current approaches should be further developed to find good quality solutions in an acceptable computation time. The feasible solution set of the *Free BL I* model is included in the feasible solution set of the *Free BL T* model. Nevertheless, the solutions obtained for this latter model are, in general, worse than the ones obtained for the former.

1.5 Conclusions

In this work, two hybrid BL population-based approaches to model the interaction between a retailer and consumers in the electricity retail market were developed. The consumer owns shiftable and interruptible loads. One approach is based on a GA algorithm and the other on PSO to address the retailer's problem (UL), both calling an exact MILP solver to solve the consumer's problem (LL). The goal of the retailer is to determine optimal dynamic ToU tariffs to be charged to the consumers in order to maximize his profit. For each instantiation of electricity prices (i.e. the retailer's decision variables), the consumer reacts by scheduling his loads to minimize his electricity bill considering comfort requirements.

Two different models to define dynamic ToU tariffs (defining periods and prices) were developed. One model considers variable pricing periods with a maximum number of different prices for the whole planning period while the other model considers that a different price can be defined for each time unit. The results obtained revealed strong difficulties for the algorithmic approaches to achieve high quality solutions for these models, which seem to be related with the inefficient exploration of the UL search space. The results also show that such difficulties are even higher for the *Free BL T* model, which presents more degrees of freedom and thus the number of feasible combinations of prices. The results obtained for each model reveal that the PSO algorithm outperforms the GA.

The capability of exploration of the UL search space is a key feature in solving the retailer's problem to obtain good solutions within an acceptable computation effort. For this purpose, future work will develop new approaches based on global optimization algorithms for mixed-integer BL programming, namely using optimal-value-function reformulations to obtain increasingly tighter bounds.

Acknowledgements This work was supported by projects UIDB/00308/2020, UIDB/05037/2020, ESGRIDS (POCI-01-0145-FEDER-016434) and MAnAGER (POCI-01-0145-FEDER-028040).

References

1. Alves MJ, Antunes CH, Carrasqueira P (2016) A hybrid genetic algorithm for the interaction of electricity retailers with demand response. In: Squillero G, Burelli P (eds) Applications of evolutionary computation. Lecture Notes in Computer Science, 9597: 459–474. Springer.
2. Soares I, Alves MJ, Antunes CH (2020) Designing time-of-use tariffs in electricity retail markets using a bi-level model—Estimating bounds when the lower level problem cannot be exactly solved. Omega 93. 102027
3. Zugno M, Morales JM, Pinson P et al (2013) A bilevel model for electricity retailers' participation in a demand response market environment. Energy Econ 36:182–197
4. Meng FL, Zeng XJ (2016) A bilevel optimization approach to demand response management for the smart grid. In: 2016 IEEE Congress on Evolutionary Computation, pp 287–294
5. Meng FL, Zeng XJ (2013) An optimal real-time pricing algorithm for the smart grid: a bi-level programming approach. In: 2013 imperial college computing student workshop (ICCSW'13) - OpenAccess Series in Informatics, pp 81–88
6. Sekizaki S, Nishizaki I, Hayashida T (2016) Electricity retail market model with flexible price settings and elastic price-based demand responses by consumers in distribution network. Int J Electr Power Energy Syst 81:371–386
7. Aussel D, Brotcorne L, Lepaul S et al (2020) A trilevel model for best response in energy demand-side management. Eur J Oper Res 281(2):299–315
8. Soares A, Gomes Á, Antunes CH (2014) Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions. Renew Sustain Energy Rev 30:490–503
9. Carrasqueira P, Alves MJ, Antunes CH (2017) Bi-level particle swarm optimization and evolutionary algorithm approaches for residential demand response with different user profiles. Inf Sci 418–419:405–420
10. Labbé M, Violin A (2016) Bilevel programming and price setting problems. Ann Oper Res 240(1):141–169

Chapter 2
An Evolutionary Algorithm for a Bilevel Biobjective Location-Routing-Allocation Problem

Herminia I. Calvete, Carmen Galé, and José A. Iranzo

Abstract In the distribution of goods to final customers, interrelated decisions have to be made, such as the location of the collection points for the goods, the routes served from the central warehouse and the allocation of customers to the collection points. The problem becomes even more complex when several decision makers are involved and multiple objectives should be taken into consideration. This paper addresses a vehicle routing problem in which customers are allowed to select the location in which they want to receive their goods among those made available by the distribution company. The aim of this company is to minimize the total cost of serving the routes as well as to satisfy customers. A bilevel biobjective problem with multiple followers is proposed to model this hierarchical supply chain. The upper level decision maker is the distribution company which decides on the locations made available and the routes which are used to serve these locations. Each customer plays the role of a follower and decides where to collect his/her goods. An evolutionary algorithm involving the solution of several optimization problems is developed for approaching the Pareto front, whose performance is assessed in a computational experiment.

Keywords Vehicle Routing Problem · Customer quality service · Biobjective · Bilevel · Evolutionary algorithm · Pareto front

H. I. Calvete
Departamento de Métodos Estadísticos, IUMA, Universidad de Zaragoza, Pedro Cerbuna 12,
50009 Zaragoza, Spain
e-mail: herminia@unizar.es

C. Galé (✉)
Departamento de Métodos Estadísticos, IUMA, Universidad de Zaragoza, María de Luna 3,
50018 Zaragoza, Spain
e-mail: cgale@unizar.es

J. A. Iranzo
Centro Universitario de la Defensa de Zaragoza, IUMA, Carretera de Huesca s/n, 50090
Zaragoza, Spain
e-mail: joseani@unizar.es

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_2

2.1 Introduction

Companies face a very complex problem when aiming to determine how to distribute commodities to final customers at low cost, with high quality service. This paper addresses a two echelon supply chain with a distribution company owning a central warehouse which serves a set of customers. The company, at the upper level of the hierarchy, designs the distribution network, i.e. it decides on the locations which are visited and the routes which are used to serve these locations using a set of homogeneous vehicles available at the central warehouse. Those locations can be either a depot to which a set of customers needs to travel to pick up their goods, or a customer who is served directly and receives his/her own goods as well as those of other customers who come to collect their goods there. In the process of designing the distribution network the company needs to take into account that customers, who are at the lower level of the hierarchy, are allowed to select their most convenient available location. Therefore, this problem involves three interconnected problems which have to be solved simultaneously: to select the locations to be visited, to identify which customers go to each location according to their preferences, and to determine the routes bearing in mind the capacity of the vehicle. The goal of the distribution company is twofold. On the one hand, as is common in Vehicle Routing Problems (VRP), it aims to minimize the total cost of serving the routes. On the other hand, bearing in mind the quality of service, it seeks customer satisfaction. Both objectives will be treated individually, thus giving rise to a biobjective problem. Figure 2.1 displays a scheme of the distribution network considered. The large red square represents the central warehouse, the small green squares refer to the depots, and the small blue circles represent the customers. There are three routes and the customers who are not visited by a route are allocated to a visited location (another customer or a depot) using the black arrows.

Bilevel programming problems have been proposed in the literature to deal with decision processes involving two decision-makers with a hierarchical structure. They are formulated as optimization problems which involve another optimization problem in the constraint set. Bilevel programs are nonconvex and difficult to solve. Even the bilevel optimization problem in which all the functions involved are linear is NP-hard. Dempe [10] provides an updated review on this topic.

Concerning the application of bilevel optimization models reported in the literature to decentralized supply chain management, Huang and Liu [12] approach a location allocation problem in which the upper level decides on the distribution center locations whereas the lower level decides on the allocation of customers, aiming to balance the workload. The algorithm proposed combines enumeration with a genetic algorithm. Cao and Chen [6] address a capacitated plant selection problem. The principal firm at the upper level selects the opening of new plants and the closing of existing plants based on its overall business considerations. At the lower level, the open plants operate independently to minimize their production and operation costs. The problem is transformed into a single level model and solved using commercial optimization software. Marinakis and Marinaki [13] reformulate a location

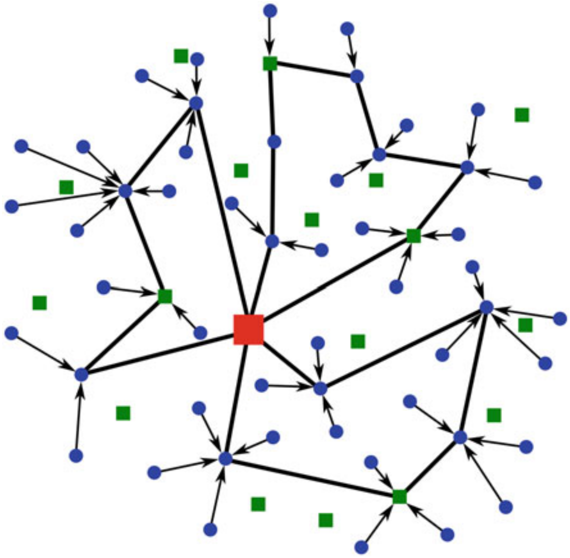


Fig. 2.1 A scheme of the distribution network

routing problem as a bilevel problem. The upper level decides on the location of the facilities, whereas the lower level decides which routes serve the customers. The algorithm proposed to solve the problem combines a genetic algorithm to solve a capacitated facility location problem with an expanding neighborhood search method to solve a VRP. Calvete et al. [5] address a production-distribution planning problem in which the distribution company at the upper level designs the routes which serve the customers, whereas the production company at the lower level controls the manufacturing process. An ant colony optimization based approach is developed to solve this problem which uses ants to construct the routes and exact optimization to solve the production problem. Calvete et al. [4] model a production-location-distribution network as a bilevel problem. The purpose is to decide which depots should be used and how the product should be distributed from manufacturing plants to depots and from these to customers, aiming to minimize fixed costs plus delivery costs, and taking into account that the manufacturing plants operate with relative independence of the distribution network, aiming to minimize their own operational costs. A hybrid evolutionary algorithm is developed to solve the problem whose key idea is to control by an evolutionary algorithm the opening of the depots together with their product availability, whereas the delivery problem from depots to customers and the manufacturing problem are exactly solved.

In this paper, a bilevel optimization problem is proposed for modeling the complex hierarchical location-routing-allocation system described above. From now on this problem will be denoted by BB-LRA. The first contribution of the paper is to extend classical location-routing models to allow for a more realistic system in which the preferences of the customers are taken into account. The resulting model is a bilevel biobjective integer optimization problem. The second contribution of the paper is to propose an evolutionary algorithm to approach the Pareto front. The paper is organized as follows. Section 2.2 describes the distribution network problem and formulates the bilevel model. In Sect. 2.3, a metaheuristic approach is developed based on evolutionary algorithms. In this approach the chromosomes control the locations which are visited. The allocation of customers is made according to their preferences. Then, the routes are obtained by solving a VRP. Section 2.4 analyzes the computational performance of the algorithm and Sect. 2.5 concludes the paper with some final remarks.

2.2 BB-LRA problem Formulation

Let $G = (\widetilde{V}, A_1 \cup A_2)$ be a graph, where $\widetilde{V}$ is the set of nodes and $A_1 \cup A_2$ is the set of arcs. The set of nodes is defined as $\widetilde{V} = \{0\} \cup V$. Node 0 represents the central warehouse, where a homogeneous fleet of vehicles is available each with capacity Q. Set $V = U \cup W$ is the set of locations, where U refers to the set of customers and W to the set of depots. Let q_u represent the demand of the customer $u \in U$.

The arcs in set $A_1 = \{(i, j) : i, j \in \widetilde{V}, i \neq j\}$ are used to link the nodes to construct the routes. We assume that there is a nonnegative cost c_{ij} associated with each arc $(i, j) \in A_1$, representing the cost of connecting nodes i and j. The arcs in set $A_2 = \{(u, i) : u \in U, i \in V\}$ are used to connect the customers to nodes in the routes. There is a nonnegative allocation cost d_{ui} associated with each arc (u, i), referring to the traveling cost due to the customer u going to the node i. We assume that $d_{uu} = 0$ for all $u \in U$.

The distribution company, acting as the upper level decision maker, determines the nodes of V which are visited and the set of routes which visit them. These routes are node-disjoint except for the central warehouse. A route is a simple cycle visiting a subset of nodes including the central warehouse. A node which is visited by a route is called a route node. Each customer $u \in U$, playing the role of a lower level decision maker, selects his/her preferred route node according to his/her preferences. In this paper, we associate preferences with allocation costs. Thus, he/she selects the route node i_u that minimizes the allocation cost over the route nodes. Since $d_{uu} = 0$, if a customer u is visited by a route, then he/she is allocated to himself/herself.

In order to formulate the model, we define the following binary variables:

$$z_i = \begin{cases} 1, & \text{if } i \in V \text{ is a route node} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i, j) \in A_1 \text{ is traversed by a vehicle} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ui} = \begin{cases} 1, & \text{if customer } u \in U \text{ is allocated to node } i, (u, i) \in A_2 \\ 0, & \text{otherwise} \end{cases}$$

To simplify the notation, we denote $\{z_i,\ i \in V;\, x_{ij},\ (i,j) \in A_1;\, y_{ui},\ (u,i) \in A_2\}$ by (z, x, y). Then, the BB-LRA problem can be formulated as the following bilevel biobjective integer multi-follower optimization problem with as many followers as customers:

$$\min_{z,x,y} \left(\sum_{(i,j)\in A_1} c_{ij}x_{ij}, \quad \sum_{(u,i)\in A_2} d_{ui}y_{ui} \right) \tag{2.1a}$$

s.t.

$$\sum_{(i,j)\in A_1} x_{ij} = z_j, \quad j \in V \tag{2.1b}$$

$$\sum_{(i,j)\in A_1} x_{ij} = z_i, \quad i \in V \tag{2.1c}$$

$$\sum_{i\in V} x_{0i} = \sum_{i\in V} x_{i0} \tag{2.1d}$$

$$Q\sum_{i\in S}\sum_{j\in \widetilde{V}\setminus S} x_{ij} \geqslant \sum_{i\in S,(u,i)\in A_2} q_u y_{ui}, \quad \forall S \subseteq V \tag{2.1e}$$

$$z_i \in \{0, 1\}, \quad i \in V; \qquad x_{ij} \in \{0, 1\}, \quad (i, j) \in A_1 \tag{2.1f}$$

where, for each customer $u \in U$, the variables y_{ui}, $(u,i) \in A_2$, solve:

$$\min_{y} \sum_{i\in V} d_{ui}y_{ui} \tag{2.1g}$$

s.t.

$$\sum_{i\in V} y_{ui} = 1 \tag{2.1h}$$

$$y_{ui} \leqslant z_i, \quad i \in V \tag{2.1i}$$

$$y_{ui} \in \{0,\ 1\}, \quad i \in V \tag{2.1j}$$

The objective function (2.1a) is a two dimensional vector. The first component represents the total cost of serving the routes. The second component measures the total allocation cost incurred by customers, and so reflects global customer satisfaction. Constraints (2.1b) and (2.1c) ensure that exactly one arc enters and leaves each route node and no arcs go through non-route nodes. Constraint (2.1d) guarantees that the same number of arcs enters and leaves the central warehouse. Constraints (2.1e) ensure connectivity of routes as well as capacity requirements. Constraints (2.1f) state that the variables z and x are binary. The lower level problem associated with the customer $u \in U$ is defined by (2.1g)–(2.1j). The objective function (2.1g) refers to the allocation cost. Constraints (2.1h) and (2.1i) ensure that the customer u is allocated to a single route node. Constraints (2.1j) state that the variables y are binary.

Because of the coupling constraints (2.1e), those points (z, x) for which the optimal solution of the lower level problem allocates to a route node a number of customers whose total demand exceeds Q should be rejected. This is one of the key ideas used in the evolutionary algorithm developed in Sect. 2.3.

Moreover, since each follower problem involves only its own variables and the upper level variables, the followers are independent [3]. Therefore, the $|U|$ lower level problems can be transformed into a single lower level problem by taking the sum of the $|U|$ objective functions as the new objective function, and mixing all the constraints.

Let us denote by $[F_1(z, x, y), F_2(z, x, y)]$ the objective function (2.1a). According to the theory of multiobjective optimization [11], a feasible solution (z, x, y) of problem (2.1) is efficient if and only if there is no other feasible solution $(\tilde{z}, \tilde{x}, \tilde{y})$ so that $F_\alpha(\tilde{z}, \tilde{x}, \tilde{y}) \leqslant F_\alpha(z, x, y), \alpha = 1, 2$ with at least one strict inequality. Let $\mathcal{F}$ be the set of feasible points in the objective space. A point $F \in \mathcal{F}$ is a nondominated outcome vector if there exists at least one efficient solution (z, x, y) so that $F = [F_1(z, x, y), F_2(z, x, y)]$. The set of all nondominated outcome vectors is the Pareto front and, in general, to construct the entire set of Pareto optimal solutions is impossible due to the complexity of problem (2.1). Therefore, below we propose an evolutionary algorithm to find a good approximation of the Pareto front.

2.3 EBA: An Evolutionary Biobjective Algorithm for Solving the BB-LRA Problem

Evolutionary algorithms are stochastic search techniques inspired by natural biological evolution. Since their introduction, they have been increasingly applied to find good solutions to complex optimization problems in acceptable computational times. Affenzeller et al. [1] and Chion et al. [7] are good texts on this topic. Coello [8] provides a recent review on multiobjective evolutionary algorithms.

The evolutionary biobjective algorithm EBA is a hybrid algorithm which embeds the optimal allocation of customers and the construction of routes inside an evolutionary algorithm which controls the locations which are visited. As mentioned above, one of the main ideas underlying the algorithm is that only those selections of route nodes whose allocated customers allow construction of the routes (due to the capacity constraint) are of interest. Moreover, having selected the route nodes, solving the lower level problem is easy since each customer chooses the route node which causes the least allocation cost. Knowing the route nodes (variables z) and their allocated customers (variables y) is enough to check if the current set of route nodes will be able to provide a permissible solution (z, x). If so, the value of the variables x which provide the best value of the routing cost can be obtained by solving a VRP. Next we describe in detail the characteristics of the algorithm.

2.3.1 Chromosome Encoding and Fitness Evaluation

Unlike most of the evolutionary algorithms that encode the feasible solutions of the incumbent problem, we propose to encode only the variables z which control the locations which can be visited, i.e. the route nodes. Hence, the chromosomes are encoded as binary $|V|$-dimensional vectors. Let $\Delta = (\delta_1, \ldots, \delta_{|V|})$ be a chromosome. Then, for each $i \in V$

$$\delta_i = \begin{cases} 1, & \text{if } i \text{ is a route node} \\ 0, & \text{otherwise} \end{cases}$$

As a result, $z_i = \delta_i$, $i \in V$. Note that a chromosome provides the nodes to which customers can be allocated, but it does not give any information about either the routes in which they are included or how the customers are allocated to them.

If $\sum_{i \in V} \delta_i < H$, where H is defined as:

$$H = \left\lceil \frac{\sum\limits_{u \in U} q_u}{Q} \right\rceil$$

i.e. H is a lower bound on the number of routes (hence, route nodes) which are needed, then the chromosome is repaired as indicated in Sect. 2.3.2.

Otherwise, in order to associate a bilevel feasible solution of the BB-LRA problem with the chromosome Δ we propose to solve several optimization problems. First, the value of the variables y is obtained by solving the lower level problems. Bearing in mind that each customer prefers the route node with the least allocation cost, the customer $u \in U$ is allocated to the route node i_u:

$$i_u = \begin{cases} u & \text{if } \delta_u = 1 \\ i & \text{if } \delta_u = 0, \text{ where } i = \arg\min\{d_{ui} : \delta_i = 1, (u, i) \in A_2\} \end{cases} \tag{2.2}$$

Thus, $y_{ui_u} = 1$ and $y_{ui} = 0$ for all $i \in V, i \neq i_u$.

After knowing this allocation, let Q_i be the total demand of the customers allocated to route node i, $Q_i = \sum_{u \in U} q_u y_{ui}$. If $Q_i > Q$, it is not possible to construct a route visiting this route node due to the vehicle capacity constraint. Hence, this chromosome should be rejected because it is not able to provide a permissible solution (z, x). In Sect. 2.3.2 we will explain two procedures to repair these chromosomes that will result in two variants of the algorithm. Assume for the time being that $Q_i \leqslant Q$ for every route node i. In order to compute the value of the variables x, we solve a VRP in which each route node i has a demand Q_i. Route nodes without allocated customers are not taken into consideration in this step.

At the end of this process, the chromosome Δ has associated a bilevel feasible solution (z, x, y) of the BB-LRA problem. Note that, for the fixed set of route nodes provided by the chromosome and their allocated customers, the algorithm solves a VRP to compute the value of the variables x. Hence, each chromosome has associated a bilevel feasible solution which implicitly discards any other bilevel feasible solutions with the same set of route nodes since they cannot be efficient. In fact, in the implementation of the algorithm we do not solve the VRP to optimality but apply a heuristic algorithm. Therefore, only the solutions with a larger routing cost than the routing cost provided by this algorithm are discarded.

We define the fitness of Δ as $[F_1(z, x, y), F_2(z, x, y)]$.

2.3.2 Repairing a Chromosome

Let Δ be a chromosome for which the number of route nodes is fewer than H. Then, it is repaired by switching the allele of as many nodes as needed, randomly selected among the nodes which currently are not route nodes.

Now, let Δ be a chromosome for which at least one route node has a set of allocated customers whose total demand exceeds the capacity of the vehicle. Let i be one of these nodes randomly selected. In order not to reject the chromosome, we propose two methods for repairing it. The first method, called RM$_1$, selects the nearest node to i, in terms of the routing cost, which currently is not a route node and switches its allele from 0 to 1. Since the chromosome has changed, all the customers are reallocated in accordance with expression (2.2). This process is repeated until $Q_i \leqslant Q$ for every route node i in the updated chromosome. The second repairing method, called RM$_2$, selects the customer allocated to the route node i with the largest demand. Let u be this customer. Then, the allele of the chromosome corresponding to this customer is changed to 1, i.e. it becomes a route node. As above, all the customers are reallocated according to expression (2.2) and the process is repeated until $Q_i \leqslant Q$ for every route node i in the updated chromosome.

After this repairing process has been carried out using one method or the other, we are in a position to compute the value of the variables x as explained in Sect. 2.3.1.

After repairing the chromosome (if necessary) and removing the route nodes without allocated customers (if any), we are also interested in determining if it is better either to maintain the original chromosome, or to update the chromosome to leave only as route nodes the nodes used when solving the VRP. This gives rise to two more variants of the algorithm. Whether or not to update a chromosome affects the offspring generated when the crossover and mutation operators are applied to a population of chromosomes.

2.3.3 Initial Population

Let P_{size} be the population size. The initial population is formed by two special chromosomes Δ_1 and Δ_2 and $P_{size} - 2$ randomly generated chromosomes. The chromosome Δ_1 is obtained by setting the H locations closest to the central warehouse, in terms of the routing cost, as route nodes. The chromosome Δ_2 is obtained by setting all the locations as route nodes. The bilevel feasible solution associated with Δ_1 (or the updated chromosome after repairing it, if necessary) gives an idea of the minimum routing cost, while that associated with Δ_2 provides the least allocation cost.

Regarding the remaining $P_{size} - 2$ chromosomes, for each chromosome a random number $p \in [0, 1]$ is generated. Then, each node is selected to be a route node with probability p. Unlike fixing a value of the probability p a priori for the chromosome as a whole, this way of selecting the route nodes encourages the existence of chromosomes having a variety in the number of route nodes.

2.3.4 Population Handling: Crossover, Mutation and Selection

The crossover operator combines parents of the incumbent population to form offspring which are potential members of a successor population. We apply the uniform crossover that randomly selects P_{size} pairs of parents and generates one offspring from each pair. Each gene of the offspring is selected from one of the parents with equal probability. Next, the mutation operator is applied to the offspring. After a chromosome has been selected, a gene is randomly selected and its allele value is switched. At the end of this process, if needed, the offspring chromosome is repaired as indicated in Sect. 2.3.2.

To handle the populations we propose to use the well-known Nondominated Sorting Genetic Algorithm II (NSGA-II) developed by Deb et al. [9] and the Indicator-Based Evolutionary Algorithm (IBEA) introduced by Zitzler and Kunzli [14], which results in two more variants of the algorithm.

When applying NSGA-II, the chromosomes of the current population plus offspring are ranked into several nondominated fronts in accordance with their fitness value and assigned a nondomination rank. For each chromosome, a second value called crowding distance is computed which gives an estimation of the density of solutions surrounding the solution associated with the chromosome in the population. The next population is obtained by selecting the best P_{size} individuals in accordance with the nondomination rank or, in case of a tie, according to the crowding distance.

Unlike NSGA-II which is based on ranking solutions, other procedure to handling the population when there are more than one objective is an indicator-based selection. This general framework allows the use of any performance indicator into the selection mechanism of a multiobjective evolutionary algorithm. In this paper, we consider IBEA which is based on a pairwise comparison of solutions by using the binary additive ϵ-indicator.

2.4 Computational Experiment

In order to analyze the performance of the EBA, a computational experiment has been carried out to analyze the influence of three factors: the survivor selection method, the chromosome updating, and the repairing process.

Since problem (2.1) has not been previously studied in the literature, no benchmark instances are available. Therefore, we have decided to adapt the set of 45 Class A instances used as benchmark instances for the Capacitated m-Ring Star Problem [2]. These instances were generated from the three TSPLIB instances called $eil51$, $eil76$ and $eil101$, as well as a fourth set which consists of the first 26 nodes of $eil51$. The location of the central warehouse and customers have been maintained, and the depots correspond to the Steiner points in those instances. The demand of the k-th customer is generated as $((k-1) \mod 5) + 1$, in accordance with the order established in the original file. The capacity of the vehicles is three times the capacity of the rings. The characteristics of the instances are shown in Table 2.1. There are nine instances with 26 nodes and twelve instances with 51, 76 and 101 nodes.

The numerical experiments have been performed on a PC Intel Core i7-6700 with 3.4 gigahertz, 32.0 gigabyte of RAM and Windows 10 64-bit as the Operating System. The code has been written in C++, TDM-GCC 4.9.2. In the computational experiment we have selected the algorithm VRP_RTR developed by C. Goer which is an implementation of the RTR metaheuristic to generate good solutions to a VRP instance. This algorithm is available at the VRPH library: https://sites.google.com/site/vrphlibrary/home.

Each combination of the three factors mentioned above provides a configuration of the algorithm. They are shown in Table 2.2. In a preliminary study, we studied the influence of the population size, but it was not significant in the performance of the algorithm. Therefore we set $P_{size} = 20$. Each test instance has been solved once under each configuration. The termination condition was established in terms of computing time, 5 min for the instances with 26 nodes, 10 min for the ones with 51 nodes, 15 min for the instances with 76 nodes and 20 min for the ones with 101 nodes.

Table 2.1 Characteristics of the instances

Instance	# of customers	# of depots	Q	Instance	# of customers	# of depots	Q
P1	12	13	15	P22	18	57	21
P2			12	P23			15
P3			9	P24			12
P4	18	7	21	P25	37	38	42
P5			15	P26			33
P6			12	P27			27
P7	25	0	30	P28	56	19	63
P8			21	P29			48
P9			18	P30			39
P10	12	38	15	P31	75	0	84
P11			12	P32			63
P12			9	P33			51
P13	25	25	30	P34	25	75	30
P14			21	P35			21
P15			18	P36			18
P16	37	13	42	P37	50	50	57
P17			33	P38			42
P18			27	P39			36
P19	50	0	57	P40	75	25	84
P20			42	P41			63
P21			36	P42			51
				P43	100	0	114
				P44			84
				P45			69

Table 2.2 Characteristics of the configurations

Configuration	Handling population	Chromosome update	Chromosome repair
1	NSGA-II	No	RM_1
2	NSGA-II	No	RM_2
3	NSGA-II	Yes	RM_1
4	NSGA-II	Yes	RM_2
5	IBEA	No	RM_1
6	IBEA	No	RM_2
7	IBEA	Yes	RM_1
8	IBEA	Yes	RM_2

Table 2.3 Results of the benchmark instances for I_H^-

Instance	Configuration							
	1	2	3	4	5	6	7	8
P1	0.005	**0.000**	0.005	0.017	0.017	**0.000**	0.005	0.017
P2	0.001	0.011	0.003	0.002	0.005	0.004	0.002	**0.000**
P3	**0.000**	**0.000**	0.001	0.016	0.001	0.001	0.001	0.016
P4	0.002	0.003	0.003	0.003	0.002	0.003	**0.001**	**0.001**
P5	0.001	0.001	0.001	0.011	0.001	**0.000**	0.001	0.002
P6	0.003	0.006	0.028	0.021	0.036	**0.000**	0.001	0.001
P7	0.004	0.004	0.004	0.005	**0.002**	**0.002**	0.011	0.003
P8	0.007	0.021	0.008	0.024	**0.006**	**0.006**	0.013	0.039
P9	0.021	0.018	0.029	0.019	0.019	**0.009**	0.024	0.016
P10	0.013	0.007	0.006	0.004	0.007	0.008	**0.002**	0.003
P11	0.012	0.010	0.017	0.019	0.011	0.010	0.012	**0.009**
P12	0.010	0.046	0.034	**0.005**	0.011	0.084	0.026	0.056
P13	**0.009**	0.013	0.014	0.034	0.010	0.014	0.019	0.048
P14	**0.011**	0.065	0.022	0.060	0.035	0.037	0.020	0.114
P15	0.024	0.018	0.017	0.106	0.016	0.009	**0.006**	0.053
P16	0.015	0.020	0.012	0.033	0.010	0.015	**0.007**	0.040
P17	0.030	0.037	0.026	0.023	0.016	**0.010**	0.037	0.036
P18	**0.025**	0.026	0.038	0.061	0.026	0.049	0.042	0.037
P19	0.021	0.020	0.023	0.025	**0.011**	0.029	0.016	0.030
P20	0.026	0.039	0.035	0.040	**0.013**	0.027	0.022	0.020
P21	0.030	0.051	0.027	0.104	0.012	0.057	0.023	**0.010**
P22	0.017	0.028	0.042	0.045	**0.007**	0.068	0.028	0.056
P23	**0.011**	0.037	0.024	0.077	0.016	0.054	0.023	0.200
P24	**0.003**	0.029	0.074	0.082	0.006	0.030	0.071	0.087
P25	0.027	0.044	0.049	0.021	**0.017**	**0.017**	0.070	0.065
P26	0.023	0.034	0.032	0.078	**0.011**	0.013	0.032	0.075
P27	0.021	0.041	0.045	0.053	**0.016**	0.067	0.036	0.071
P28	0.028	0.076	0.064	0.097	**0.009**	0.025	0.055	0.059
P29	0.048	0.046	0.062	0.102	**0.009**	0.038	0.050	0.098
P30	**0.024**	0.059	0.058	0.106	0.060	0.062	0.047	0.071
P31	0.028	0.040	0.026	0.039	**0.006**	0.048	0.019	0.044
P32	0.029	0.034	0.027	0.054	**0.013**	0.034	0.020	0.032
P33	**0.025**	0.038	0.039	0.068	0.029	0.042	0.034	0.030
P34	0.025	0.025	0.013	0.045	**0.004**	0.014	0.036	0.058
P35	0.026	0.022	0.023	0.091	0.028	0.028	**0.011**	0.088
P36	**0.020**	0.042	0.046	0.097	0.038	0.032	0.051	0.153
P37	0.033	0.028	0.030	0.053	**0.013**	0.018	0.017	0.069
P38	0.038	0.049	0.038	0.051	**0.005**	0.042	0.025	0.038
P39	0.029	0.041	0.031	0.106	**0.012**	0.080	0.015	0.119
P40	0.027	0.032	0.039	0.056	**0.008**	0.016	0.029	0.095
P41	0.031	0.025	0.025	0.105	0.014	**0.012**	0.015	0.101
P42	0.038	**0.022**	0.048	0.112	0.025	0.054	0.035	0.128
P43	0.023	0.028	0.035	0.058	**0.008**	0.025	0.026	0.056
P44	0.031	0.045	0.027	0.077	**0.012**	0.034	0.015	0.064
P45	0.029	0.045	0.030	0.090	**0.012**	0.024	**0.012**	0.071

Table 2.4 Results of the benchmark instances for $I^1_{\epsilon+}$

Instance	Configuration							
	1	2	3	4	5	6	7	8
P1	0.048	0.010	0.048	0.083	0.083	**0.005**	0.048	0.083
P2	0.019	0.061	0.020	0.020	0.027	0.027	0.020	**0.014**
P3	**0.008**	**0.008**	0.016	0.089	**0.008**	**0.008**	0.014	0.089
P4	0.020	0.020	0.020	0.020	0.020	0.020	0.020	**0.010**
P5	**0.009**	0.010	**0.009**	0.051	**0.009**	**0.009**	0.015	0.018
P6	0.026	0.031	0.120	0.068	0.099	**0.010**	0.016	**0.010**
P7	0.012	0.020	0.013	0.018	0.012	**0.009**	0.054	0.020
P8	**0.020**	0.073	0.023	0.073	0.026	**0.020**	0.034	0.108
P9	0.071	0.056	0.089	0.056	0.058	**0.022**	0.054	0.056
P10	0.064	0.030	0.021	0.020	0.021	0.021	**0.011**	0.021
P11	0.027	0.027	0.064	0.071	**0.022**	0.032	0.043	0.033
P12	0.056	0.183	0.142	**0.030**	0.056	0.244	0.107	0.203
P13	0.026	**0.022**	0.048	0.118	0.033	0.051	0.051	0.165
P14	**0.048**	0.231	0.103	0.214	0.155	0.172	0.069	0.317
P15	0.049	0.035	0.035	0.282	0.049	0.023	**0.018**	0.176
P16	0.025	0.035	0.024	0.095	0.020	0.030	**0.013**	0.101
P17	0.042	0.090	0.072	0.042	0.042	**0.018**	0.120	0.102
P18	**0.038**	0.041	0.100	0.125	0.056	0.134	0.103	0.100
P19	0.028	**0.023**	0.033	0.028	0.031	0.057	0.036	0.044
P20	**0.035**	0.091	0.076	0.103	0.071	0.108	0.076	0.063
P21	0.045	0.097	0.031	0.213	0.023	0.121	0.036	**0.020**
P22	0.035	0.115	0.124	0.128	**0.031**	0.230	0.071	0.173
P23	0.048	0.075	0.040	0.167	**0.023**	0.106	0.040	0.335
P24	**0.014**	0.064	0.236	0.241	0.027	0.155	0.241	0.277
P25	**0.034**	0.114	0.157	0.066	0.036	0.057	0.214	0.202
P26	**0.029**	0.057	0.085	0.176	0.039	0.034	0.091	0.179
P27	0.040	0.057	0.110	0.138	**0.033**	0.082	0.103	0.156
P28	0.036	0.179	0.176	0.240	**0.016**	0.096	0.165	0.179
P29	0.083	0.076	0.121	0.194	**0.023**	0.056	0.121	0.210
P30	**0.036**	0.174	0.174	0.255	0.174	0.197	0.166	0.213
P31	0.030	0.101	0.063	0.081	**0.016**	0.145	0.063	0.135
P32	0.039	0.037	0.041	0.103	**0.023**	0.068	0.050	0.089
P33	**0.037**	0.069	0.108	0.097	0.093	0.097	0.114	0.063
P34	0.051	0.063	**0.016**	0.152	**0.016**	0.066	0.142	0.190
P35	0.031	0.073	0.036	0.234	0.086	0.040	**0.022**	0.231
P36	**0.033**	0.070	0.127	0.240	0.117	0.073	0.143	0.333
P37	0.045	0.039	0.061	0.131	**0.025**	**0.025**	0.038	0.194
P38	0.061	0.130	0.117	0.135	**0.015**	0.135	0.103	0.128
P39	0.085	0.103	0.085	0.252	**0.029**	0.219	0.085	0.283
P40	0.031	0.041	0.052	0.133	**0.019**	0.035	0.085	0.217
P41	0.036	0.030	0.048	0.225	**0.027**	0.033	0.052	0.236
P42	0.052	**0.031**	0.120	0.241	0.063	0.122	0.120	0.279
P43	0.033	0.041	0.059	0.126	**0.023**	0.067	0.072	0.136
P44	**0.035**	0.075	0.045	0.153	0.045	0.099	0.047	0.155
P45	0.034	0.087	0.034	0.191	**0.023**	0.081	0.034	0.188

In order to evaluate the quality of the Pareto front approximations yielded by the algorithm configurations analyzed, we used the performance assessment tool suite provided in PISA, http://www.tik.ee.ethz.ch/pisa. For each test instance, we computed the reference set Z_N^* which is formed by all the nondominated points available, i.e. the union of the outputs obtained throughout the whole experiment. We used two indicators which measure the quality of an output set A in comparison to Z_N^*. The unary hypervolume metric I_H^- computes the area of the objective space that is weakly dominated by Z_N^* and not by A. The binary additive ϵ-indicator $I_{\epsilon+}^1$ computes the minimum factor by which A has to be translated in the objective space to weakly dominate Z_N^*. The closer the indexes to zero, the better the approximation.

Tables 2.3 and 2.4 display the results of the experiment. Both tables are similar, except for the indicator shown. The first column gives the name of the problem; the second to ninth columns the corresponding indicator values. For each instance, the best value is written in bold. Figures 2.2 and 2.3 summarize the information given by the indicator values by using individual value plots. The x-axis shows the configurations. For each configuration, the figure displays data corresponding to the nine or twelve instances included in the instance set defined by the number of nodes. The blue point is the average. A small variability means that the indicator is less sensitive to changes in the number of customers and the vehicle capacity. Looking at the whole figures, we can infer that the configuration influences the value of the indicators. Moreover, both indicators lead us to the same conclusions. Except for 26 nodes instances, the configuration 5 provides the best values, while the configurations

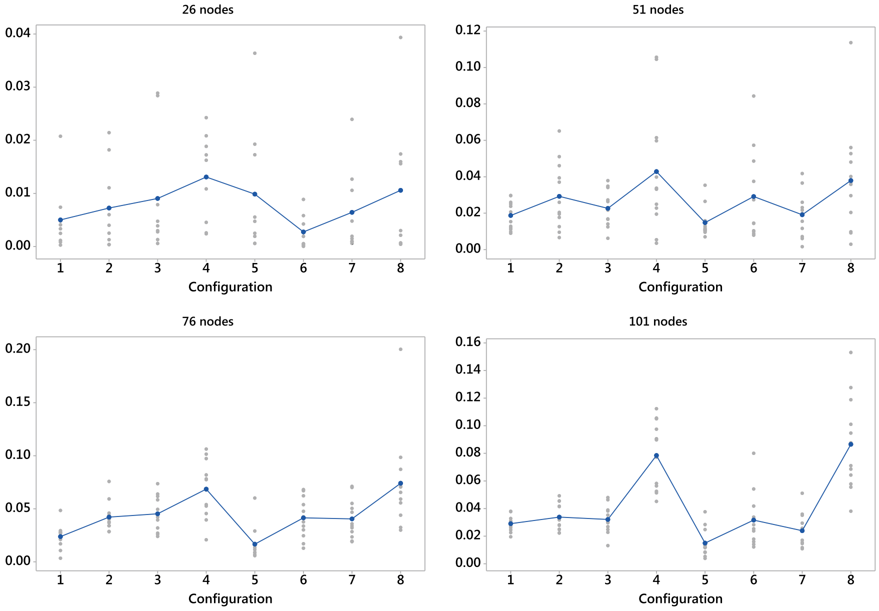


Fig. 2.2 Value plots for I_H^-

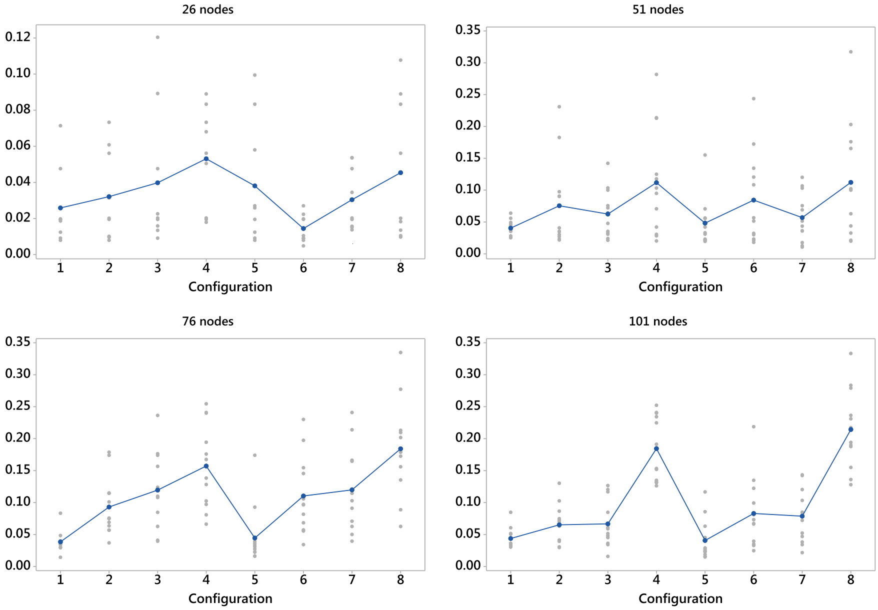


Fig. 2.3 Individual value plots for $I^1_{\epsilon+}$

4 and 8 are the worst configurations in all cases. The configuration 5 provides the smallest average and the spread of data is also smaller.

To confirm that the influence of the configuration on the indicators is statistically significant, a Friedman test has been applied. This is a nonparametric analysis of a randomized block experiment which provides an alternative to the two-way analysis of variance when the assumption of normality is not justified. A p-value equal to zero has been obtained, thus confirming that the configuration significantly influences the results, the best one being the configuration 5. As an illustration, Fig. 2.4 shows the Pareto front provided by this configuration in the first instance of each group (instances P1, P10, P22, P34).

2.5 Conclusions

This paper addresses a bilevel biobjective multi-follower optimization model to deal with a hierarchical supply chain in which the distribution company needs to take into account that customers are free to select the most convenient location for receiving their goods. This problem can be transformed into a single level biobjective problem, but its combinatorial nature make it complex enough to require metaheuristic techniques to be solved.

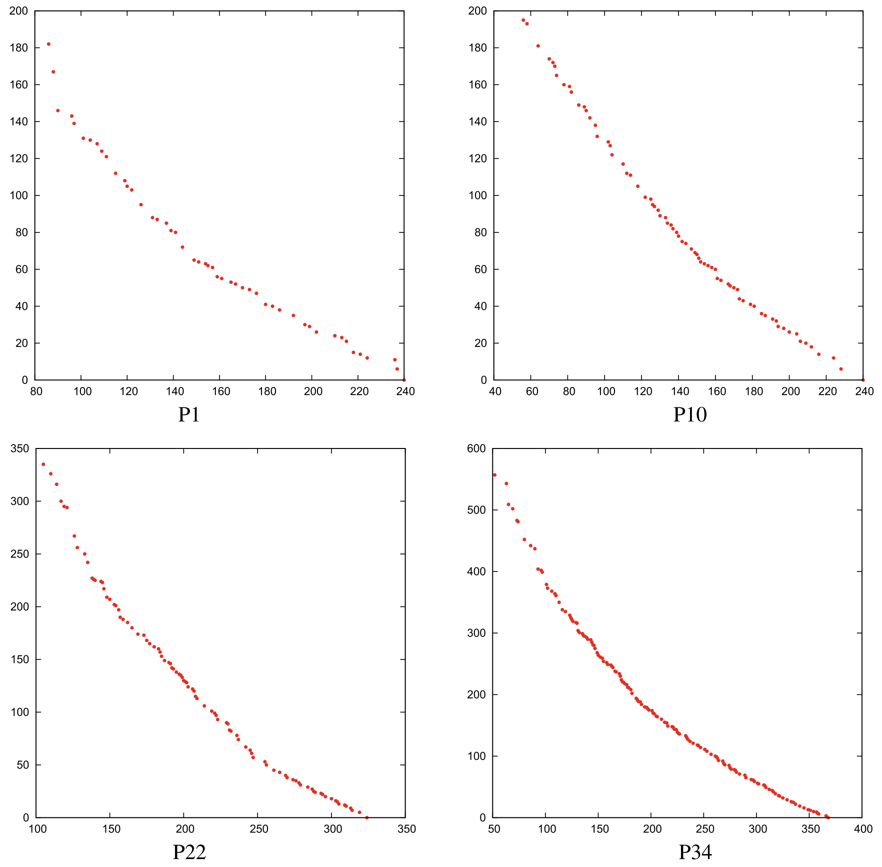


Fig. 2.4 Pareto front provided by the configuration 5 for the benchmark instances P1, P10, P22 and P34

An evolutionary algorithm involving the solution of several optimization problems has been developed for approaching the Pareto front. Each chromosome indicates which locations can be used for serving the customers. Then, each customer decides on the one where he/she prefers to collect the goods. Based on this information, a VRP is solved to provide a bilevel feasible solution. Eight variants of the algorithm have been tested which explore two methods for repairing the chromosomes in case they are not able to provide a bilevel feasible solution due to the vehicle capacity constraint, two ways of handling the chromosomes which are repaired, as well as two methods for selecting the survivors from the current population plus the offspring, NSGA-II and IBEA. All these variants are analyzed in a computational experiment, obtaining that the configuration in which IBEA is used for selecting the next population, the chromosomes are repaired according to RM_1, and no updating is carried out is the best one.

Acknowledgments This research work has been funded by the Spanish Ministry of Economy, Industry and Competitiveness under grant ECO2016-76567-C4-3-R and by the Gobierno de Aragón under grant E41_17R (FEDER 2014–2020 "Construyendo Europa desde Aragón").

References

1. Affenzeller M, Wagner S, Winkler S, Beham A (2009) Genetic algorithms and genetic programming: modern concepts and practical applications. Chapman and Hall/CRC
2. Baldacci R, Dell'Amico M, Salazar González JJ (2007) The capacitated m-ring-star problem. Oper Res 55(6):1147–1162
3. Calvete HI, Galé C (2007) Linear bilevel multi-follower programming with independent followers. J Global Optim 39(3):409–417
4. Calvete HI, Galé C, Iranzo JA (2011) Planning of a decentralized distribution network using bilevel optimization. Comput Oper Res 38(1):320–327
5. Calvete HI, Galé C, Oliveros MJ (2011) Bilevel model for production-distribution planning solved by using ant colony optimization. Comput Oper Res 38(1):320–327
6. Cao D, Chen M (2006) Capacitated plant selection in a decentralized manufacturing environment: a bilevel optimization approach. Eur J Oper Res 169(1):97–110
7. Chion R, Weise T, Michalewicz Z (eds) (2012) Variants of evolutionary algorithms for real-world applications. Springer, Berlin
8. Coello CA (2017) Recent results and open problems in evolutionary multiobjective optimization. In: Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). LNCS:3–21, p 10687
9. Deb K, Pratap A, Agrawal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
10. Dempe S (2018) Bilevel optimization: theory, algorithms and applications. Preprint 2018-11, Fakultät für Mathematik und Informatik, TU Bergakademie Freiberg
11. Ehrgott M (2005) Multicriteria optimization, 2nd edn. Springer, Heildeberg
12. Huang B, Liu N (1894) Bilevel programming approach to optimizing a logistic distribution network with balancing requirements. Transp Res Record J Transp Res Board 188–197:2004
13. Marinakis Y, Marinaki M (2008) A bilevel genetic algorithm for a real life location routing problem. Int J Logistics Res Appl 11(1):49–65
14. Zitzler E, Kunzli S (2004) Indicator-based selection in multiobjective search. In: Yao X, Burke E, Lozano JA, Smith J, Merelo-Guervós JJ, Bullinaria JA, Rowe J, Tiño P, Kabán A, Schwefel HP (eds) Eight international conference on parallel Problem Solving from Nature (PPSN VIII). Lecture notes in computer science, vol 3242. Springer, Berlin, pp 832–842

Chapter 3
Incorporation of Region of Interest in a Decomposition-Based Multi-objective Evolutionary Algorithm

Ivan Reinaldo Meneghini, Frederico Gadelha Guimarães, António Gaspar-Cunha, and Miri Weiss Cohen

Abstract Preference-based Multi-Objective Evolutionary Algorithm (MOEA) restrict the search to a given region of the Pareto front preferred by the Decision Maker (DM), called the Region of Interest (ROI). In this paper, a new preference-guided MOEA is proposed. In this method, we define the ROI as a preference cone in the objective space. The preferential direction and the aperture of the cone are parameters that the DM has to provide to define the ROI. Given the preference cone, we employ a weight vector generation method that is based on a steady-state evolutionary algorithm. The main idea of our method is to evolve a population of weight vectors towards the characteristics that are desirable for a set of weight vectors in a decomposition-based MOEA framework. The main advantage is that the DM can define the number of weight vectors and thus can control the population size. Once the ROI is defined and the set of weight vectors are generated within the preference cone, we start a decomposition-based MOEA using the provided set of weights

I. Reinaldo Meneghini (✉)
Instituto Federal de Educação Ciência e Tecnologia de Minas Gerais (IFMG) Campus Ibirité, 2
Mato Grosso St., Minas Gerais, Ibirité 32407-190, Brazil
e-mail: ivan.reinaldo@ifmg.edu.br

F. Gadelha Guimarães
Department of Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), 6627
Pres. Antônio Carlos Ave., Belo Horizonte, Minas Gerais 31270-901, Brazil
e-mail: fredericoguimaraes@ufmg.br

A. Gaspar-Cunha
Institute of Polymers and Composites, University of Minho, Campus de Azurém, 4800-058
Guimarães, Portugal
e-mail: agc@dep.uminho.pt

M. Weiss Cohen
Department of Software Engineering, Braude College of Engineering, Karmiel, Snunit St 51,
2161002 Karmiel, Israel
e-mail: miri@braude.ac.il

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_3

in its initialization. Therefore, this enforces the algorithm to converge to the ROI. The results show the benefit and adequacy of the preference cone MOEA/D for preference-guided many-objective optimization.

Keywords MOEA/D, ROI · Multi-Objective Optimization · Weight Vectors

3.1 Introduction

Multi-objective evolutionary algorithms (MOEA) are recognized as suitable methods to find high quality approximations to the set of solutions to multi-objective optimization problems [19]. These optimal solutions, known as Pareto optimal solutions, are characterized by the trade-off relation between the conflicting objectives, such that some improvement in one objective function must lead to deterioration in at least one of the other objectives. However, as the number of objectives grows, we reach the field of many-objective optimization problems (MaOPs) [11]. This boundary is usually defined when the number of objective functions is greater than three or four, given empirical studies about the downgrading performance of most multi-objective algorithms when the number of objectives increase, see for instance [2].

Without any prior preference provided by the decision-maker (DM), MOEA are designed to find an unbiased, well-distributed approximation of the entire Pareto Front (PF), a task that becomes increasingly harder in MaOPs. This brings a number of challenges related to converging to such a large set of solutions, visualizing solutions found, performing decision-making with a large number of alternatives [11]. Moreover, a high computational cost of properly sampling of the high-dimensional Pareto front. For this reason, many preference-based MOEA have been proposed in the literature [3], designed to converge to a subset of Pareto-optimal solutions located at a given region of the PF preferred by the DM, usually called Region of Interest (ROI). These preference-based MOEA are an intermediate approach for incorporating preferences in multi-objective optimization: a priori information is needed to define the ROI and, after some desirable solutions are found, the DM can select the most satisfying one *a posteriori* or restart the process by adjusting the ROI, hence following an *interactive* approach. With this novel approach, one can avoid the main disadvantages of the a priori methods. Defining the ROI might be easier for the DM than modelling the preferences into specific parameters of a parameterized single-objective optimization problem. Furthermore, it can alleviate the high computational cost and time consumption of full *a posteriori* methods.

The proposed methodology in this paper is therefore an intermediate approach for incorporating preferences into many-objective optimization problems. It is a methodology aligned with the trend of interactive approaches, and follows the framework of any MOEA based on decomposition.

In the last decade, MOEA based on decomposition/aggregation methods have attracted the attention of the evolutionary multi-objective optimization community, with several studies to show their potential and limitations, and to improve their performance in constrained multi- and many-objective optimization problems [17]. The decomposition-based MOEA rely on aggregation functions that are based on different weight vectors. Those weight vectors might represent a weighted aggregation of objectives or a preference direction in objective space depending on the interpretation and the context of the decomposition method adopted within the algorithm. The key point is that the weight vector generation method is the primary step in decomposition-based MOEA, affecting the diversity of the Pareto approximation and overall performance of the algorithm.

In this paper, we define the ROI as a preference cone in the objective space. The preference cone could be defined by a preferential direction vector $\mathbf{v}$, which corresponds to the axis of the cone, having the origin or utopian point as the apex, and the angle τ between the axis and the generating lines (generatrix). The preferential direction and the aperture of the cone are the parameters that the DM has to provide to define the ROI. Given the preference cone, we employ a weight vector generation method that is based on a steady-state evolutionary algorithm. The main idea of our method is to evolve a population of weight vectors towards the characteristics that are desirable for a set of weight vectors in a decomposition-based MOEA framework. Once the ROI is defined and the set of weight vectors are generated within the preference cone, we start a decomposition-based MOEA using the provided set of weights in its initialization. Therefore, this enforces the algorithm to converge to the preference cone, which in turn represents the ROI to the DM.

3.2 Background

3.2.1 Many-Objective Optimization

A multi-objective optimization problem (MOP) [19] is defined by:

$$\mathbf{x}^{\star} = \arg\min F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))$$
$$\text{subject to: } \begin{cases} G(\mathbf{x}) \leq 0, \\ H(\mathbf{x}) = 0, \\ \mathbf{x} \in \Omega \end{cases} \tag{3.1}$$

where $\mathbf{x} \in \Omega$ are the decision variables in the decision space Ω. Their image, $\mathbf{y} = F(\mathbf{x})$ given by the function F, is the objective space. The functions $G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_P(\mathbf{x}))$ and $H(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_Q(\mathbf{x}))$ define the inequality and equality constraints respectively. The constraint functions define the feasible set $\Omega \subseteq X$ and the feasible region in the objective space $F(\Omega) \subseteq Y$. This paper will consider only the case where $X \subseteq \mathbb{R}^N$ and $Y \subseteq \mathbb{R}^M$. In the application $F : X \to Y$, each coordinate $f_i(\mathbf{x})$ of $F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))$ is an objective function of the MOP defined in (3.1).

The solution of problem (3.1) uses the following relationship between $\mathbb{R}^M$ vectors: Let $\mathbf{u} = (u_1, \ldots, u_M)$ and $\mathbf{v} = (v_1, \ldots, v_M)$ be vectors in $\mathbb{R}^M$. Then $\mathbf{u} \prec \mathbf{v}$ if and only if $u_i \leq v_i \; \forall i \in \{1, \ldots, M\}$ and $\exists \; i \in \{1, \ldots, M\}$ such that $u_i \neq v_i$.

If $\mathbf{u} \prec \mathbf{v}$ then $\mathbf{u}$ (Pareto) dominates $\mathbf{v}$ and if neither $\mathbf{u} \not\prec \mathbf{v}$ nor $\mathbf{v} \not\prec \mathbf{u}$ then $\mathbf{u}$ and $\mathbf{v}$ are said to be non-dominated. In this case, the solution of the MOP (3.1) is defined by the set $\mathcal{P} = \{\mathbf{x} \in \Omega; \nexists \; \mathbf{s} \in \Omega \text{ such that } F(\mathbf{s}) \prec F(\mathbf{x})\}$. This set is called the Pareto-optimal Set of the MOP (3.1) and its elements are minimal solutions by the partial order $\prec$. The image $F(\mathcal{P})$ of the points in $\mathcal{P}$ in the objective space Y is (in general) an $(M-1)$-dimensional manifold (for continuous problems), called the Pareto Front (PF).

Obtaining the exact solution of a MOP is a very difficult task. Since the objectives in a MOP are conflicting and the order relation that establishes the best solution is a partial order. There is no single solution for a given MOP, but a set of non-dominated solutions of large or even infinite cardinality. The general desirable solution of a MOP consists of an approximation of a subset of $\mathcal{P}$ (or $F(\mathcal{P})$) with the following important characteristics:

- the approximation set is sufficiently close to a subset of the Pareto Front;
- the approximation set presents maximum coverage of the Pareto Front.

This second item guarantees the existence of feasible solutions in any part of the Pareto Front. In a hypothetical situation of no preference on the part of the decision maker, any of these solutions can be chosen arbitrarily.

An efficient way of determining an approximation of the solution of these problems is through Multi-Objective Evolutionary Algorithms (MOEA) [19]. In this methodology, in each iteration a set of new candidate solutions is produced from the current population in order to determine, a set of non-dominated points (from the objective space sets). The main difference between the types of MOEA is according to the selection of a new population. These algorithms are categorized as follows [19]:

- Decomposition-based MOEA: In this approach, a set of weight vectors (or direction vectors) are created along the objective space and associated to the population. Then, the MOP is decomposed into a number of Single Objective Problems, each one representing a parameterized scalarizing function. MOEA/D [18] and NSGA-III [6] are examples of algorithms that use this method.
- Dominance-based MOEA: In this approach, all the objectives are optimized simultaneously and the new individuals in the populations are selected using the dominance relation. NSGA-II [5] and SPEA2 [21] are examples of algorithms that use this method.
- Indicator-based MOEA: In this approach, all the objectives are optimized simultaneously and the new individuals in the populations are selected using a quality indicator, as the hypervolume. IBEA [20] and HypE [1] are examples of algorithms which are based on this method.

Usually, if $M > 3$, the problem (3.1) becomes a MaOP [3]. The increase in the number of objectives is accompanied by the exponential increase in the number of non-dominated solutions, incomparable by the criteria of optimality, Hence, resulting, the convergence of the population becomes an extremely difficult task. Another problem is the generation and the selection of new individuals in the population, since the high number of non-dominated points in the population causes the selection to be random due to the lack of similarity parameters or differences between the points. In addition to the loss of selective pressure, the set of points required to represent or approximate the Pareto Front becomes very large. This increase in the number of points capable of representing the real Pareto Front implies in the increase of the population size used in the evolutionary algorithms, which becomes unnecessarily large. Finally, visualization of the final solution in the objective space is very limited in MaOPs.

3.2.2 Introducing Preferences in MOEA

The incorporation of preferences by the DM in MOEA can be determined in a threefold manner: before the search (a priori approach), during the search (interactive approach) or after the search (*a posteriori* approach) [3].

Without any prior preference provided by the DM, MOEA methods are designed to obtain an unbiased, well-distributed approximation of the entire Pareto front, a task that becomes definitely harder in MaOPs. Preference-based MOEA are designed to converge to a subset of Pareto-optimal solutions located at a specific region of the Pareto Front preferred by the DM, usually called Region of Interest (ROI), which can be defined in several ways.

For dominance-based MOEA, a popular method are MOEA based on an Achievement Scalarizing Function (ASF) [8, 14]. Those MOEA use a reference (or goal) point **Z** in the Objective Space, representing the DM preference. Combining the information of dominance and the reference point, the MOP is transformed into a single objective problem by the minimization of the scalarizing function using some distance or norm. A drawback of this method is that the size of the ROI is affected by the type of norm used as well as the relative position between the reference point **Z** and the PF.. In a decomposition-based MOEA, the preferences of the DM can be articulated through weight vectors [4].

3.3 Methodology

3.3.1 The Preference Cone

The proposed method uses a cone of vectors to define the ROI. A cone is defined as a geometric shape formed by a set of half-lines (called generatrices) connecting to a common coordinate point (apex). The base is a defined plane which does not contain

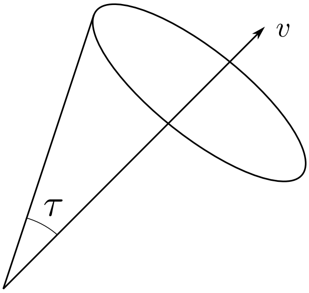


Fig. 3.1 Axis v, angle τ and a generatrix of the cone

the apex point (coordinates). A preference cone is defined by a preferential direction vector $\mathbf{v}$, which corresponds to the axis of the cone. The origin or utopian point serves as the apex, and the angle τ between $\mathbf{v}$ and the generating lines (generatrix). These elements are illustrated in Fig. 3.1.

The preferential direction vector $\mathbf{v}$ indicates the preference of the DM. The coordinates of this vector can be the desired value for each objective or present the relative importance between each of them. For example, in a problem with three objectives, if the first objective has double importance value in comparison of the remaining two, this information translates into the vector $\mathbf{v} = (2, 1, 1)$. The aperture angle τ indicates the extension of the ROI: a small value produces a small region, providing a localized solution search. Increasing the value of this angle extends the search to a larger region. Important methods to obtain these parameters are available in the literature such as the Analytical Hierarchy Process (AHP) [15] and the Stepwise Weight Assessment Ratio Analysis (SWARA) [10] methods.

Similar to the weight vectors used in Decomposition/Aggregation-based algorithms, the weight vectors of the cone are located in the hypercube $[0, 1]^M \subset \mathbb{R}^M$. The generation of weight vectors inside the preference cone is based on a steady-state evolutionary algorithm. The basic idea is to evolve an initial population W containing n vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ in the hypercube $[0, 1]^M \subset \mathbb{R}^M$ at random. Next, normalize these vectors and calculate the distance matrix $d_{i,j}$ between every pair $\mathbf{w}_i$ and $\mathbf{w}_j$. For each vector $\mathbf{w}_i \in W$, create a new vector $\mathbf{w}'_i$ from $\mathbf{w}_i$ and calculate the distance d'_j between $\mathbf{w}'_i$ and $\mathbf{w}_j \in W$. The new vector $\mathbf{w}'_i$ is created by adding a Gaussian perturbation to $\mathbf{w}_i$. After that, remove one vector from W in order to maximize the shortest distance between the new vector $\mathbf{w}'_i$ and the remaining vectors $\mathbf{w}_j \in W$, following an ES$(\mu + 1, \mu)$ selection scheme. The sum of the distances to the closest neighbors in W is the fitness function that guides the evolution of the set of weight vectors. Algorithm 1 presents the summarized structure of the proposed method. The details of the method are described in the following steps.

Initialization: In this step the initial parameters are defined.

1. Define the number of weight vectors n to be generated.
2. Define the axis $\mathbf{v}$ and the angle τ of the cone.
3. Define the p-norm to be used. Let $\mathbf{x} = (x_1, \ldots, x_M)$ be a vector in the M-dimensional vectorial space, its p-norm is given by

```
$iter \leftarrow 0$;
$W = \{\mathbf{w}_1, \ldots \mathbf{w}_n\} \leftarrow$ Initialize population;
$W \leftarrow$ Normalize population($W$);
$d_{i,j} \leftarrow$ Evaluate distance between $\mathbf{w}_i$ and $\mathbf{w}_j$;
$\phi(\mathbf{w}_i) \leftarrow$ Evaluate fitness function for $\mathbf{w}_i \in W$;
while stop criterion is not met do
    Choose $i$ from $\{1, \ldots, n\}$ at random;
    $\mathbf{w}'_i \leftarrow \mathbf{w}_i + \delta$;
    $d'_j \leftarrow$ Evaluate distance between $\mathbf{w}'_i$ and $\mathbf{w}_j \in W$;
    $\xi_i \leftarrow$ Evaluate the angle between the new element $\mathbf{w}'_i$ and the axis $\mathbf{v}$ of the cone;
    $\phi(\mathbf{w}'_i) \leftarrow$ Evaluate fitness function of the new element $\mathbf{w}'_i$;
    Replace the worst $\mathbf{w}_j$ from $W$ by $\mathbf{w}'_i$;
    $d_{i,j} \leftarrow$ Update the distance matrix;
    $iter \leftarrow iter + 1$;
end
```

Algorithm 1: Weight Vector Generation pseudo code

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{M} |x_i|^p\right)^{1/p} \tag{3.2}$$

If $p = 1$, the Manhattan norm is defined, and if $p = 2$ the Euclidean norm is described. The following equation is characterized:

$$\mathbf{y} = \frac{\mathbf{x}}{\|\mathbf{x}\|_p} \tag{3.3}$$

we define $\|\mathbf{y}\|_p = 1$. By this, $\|\mathbf{y}\|_2$ is a point on a sphere centered at $O = (0, \ldots, 0)$ and unitary radius, while $\|\mathbf{y}\|_1$ is on the plane $x_1 + \ldots + x_M = 1$.

4. Define the maximum number of iterations $itermax$ and the number of neighbors T. The value of T is used in the fitness function computation. After conducting some test, we ascertained the value $T = 2$.

After establishing these initial parameters, generate the initial population W at random and normalized according to (3.3). Finally, calculate the Euclidean distance matrix $d_{i,j}$ between pairs of $\mathbf{w}_i$ and $\mathbf{w}_j$.

Evolutionary cycle: While the stop criteria is false, repeat the following steps below:

1. Choose an arbitrary element $\mathbf{w}_i \in W$ at random.
2. Create a new weight vector

$$\mathbf{w}'_i = \mathbf{w}_i + \delta_i \tag{3.4}$$

The perturbation vector δ_i is obtained as follows:

a. Determine the smallest distance $d_{i,\min}$ between $\mathbf{w}_i$ and other vectors $\mathbf{w}_j \in W$, $i \neq j$.

b. Compute the penalty $\alpha = (1 - t)a + tb$ with $t = \frac{k}{itermax}$, where k is the current iteration and $itermax$ is the maximum number of iterations. In our tests we use $a = 1.5$ and $b = 0.1$.
c. Calculate $\delta_i = (\delta_{i1}, \ldots, \delta_{iM})$, where δ_{ij} is a random variable with normal distribution of zero mean and standard deviation $\sigma_i = \alpha \times d_{i,\min}$, that is, $\delta_{ij} \sim N(0, \sigma_i)$. This choice allows the adaptation of the mutation size according to the neighborhood of the point. Moreover, it favors exploration in the beginning and local search towards the end.
d. Normalize the new vector $\mathbf{w}'_i = (w'_{i1}, \ldots, w'_{iM})$ using Eq. (3.3).

3. Calculate the Euclidean distance between $\mathbf{w}'_i$ and the remaining vectors $\mathbf{w}_j \in W$.
4. Calculate the angle ξ_i between $\mathbf{w}'_i$ and the axis of the cone $\mathbf{v}$.
5. Calculate the fitness function $\phi(\mathbf{w}_j)$, $\mathbf{w}_j \in W$.
6. Let

$$\mathbf{w}_{min} = \arg\min_j \phi(\mathbf{w}_j), \ \ \mathbf{w}_j \in W \tag{3.5}$$

If $\phi(\mathbf{w}'_i) > \phi(\mathbf{w}_{min})$ replace $\mathbf{w}_{min}$ by $\mathbf{w}'_i$ in W and update the distance matrix $d_{i,j}$. Otherwise discard $\mathbf{w}'_i$.

7. Update the iteration counter.

The fitness function $\phi_i(\mathbf{w}_i)$ of $\mathbf{w}_i \in W$ is given by the sum $S_T(\mathbf{w}_i)$ of the distances from $\mathbf{w}_i$ to its T closest neighbors in W, penalized by the angle ξ_i between $\mathbf{w}'_i$ and the axis of the cone $\mathbf{v}$. If $\xi_i > \tau$, thus the vector lies outside the cone and therefore its fitness function value should be penalized. The fitness function is defined as follows:

$$\phi(\mathbf{w}_i) = S_T(\mathbf{w}_i) - M \times \max(\xi_i - \tau, 0) \tag{3.6}$$

3.4 Results and Discussion

This section presents some experiments using the problems (case studies) DTLZ1, DTLZ2 [7] and WFG1 [9] with 3, 5 and 10 objectives. These problems are often used by the scientific community and are suitable for this study. They are scalable to any number of objectives and have PF with leading characteristics. The aim of these experiments is to compare Dominance-based and Decomposition-Based MOEA in the exploration of a ROI in the Objective Space, using multi-objective problems. The Dominance-based algorithm selected is the g-NSGA-II [13] and the MOEA/D [18] representing the decomposition-based algorithm. For the decomposition-based algorithm, a cone of weight vectors is used instead the usual weight vector generation. As mentioned in [6], the set of weight vectors can represent the preferences of the DM for the location of the solutions in the objective space.

Table 3.1 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the $v1$ direction.

		DTLZ1	DTLZ2	WFG1
3 objectives				
IGD	gNSGAII	2.217e+00(3.570e+00)	3.759e–01(1.237e–03)	1.272e+00(3.184e–02)
	MOEA/D	1.354e–01(6.500e–05)	3.681e–01(2.238e–04)	1.300e+00(2.943e–03)
HV	gNSGAII	2.536e–02(3.127e–02)	2.775e–01(2.255e–03)	3.058e+01(1.859e+00)
	MOEA/D	7.683e–02(8.592e–05)	2.896e–01(2.485e–04)	3.942e+01(6.912e–02)
5 objectives				
IGD	gNSGAII	5.948e+02(1.444e+02)	1.626e+00(6.276e–01)	2.938e+00(8.431e–01)
	MOEA/D	1.480e–01(7.993e–05)	4.828e–01(5.274e–04)	1.621e+00(4.876e–02)
HV	gNSGAII	0(0)	2.832e–04(7.921e–04)	1.333e+03(1.229e+03)
	MOEA/D	2.800e-02(9.229e–05)	3.546e–01(1.938e–03)	4.881e+03(1.764e+01)
10 objectives				
IGD	gNSGAII	⋆	7.913e+00(2.663e+00)	6.514e+00(1.777e+00)
	MOEA/D	1.828e–01(2.365e–03)	7.574e–01(1.942e–02)	2.708e+00(1.358e–01)
HV	gNSGAII	⋆	0(0)	4.142e+08(8.492e+08)
	MOEA/D	1.313e–03(5.117e–05)	2.117e–01(3.408e–02)	7.096e+09(8.364e+08)

3.4.1 Experimental Setup

The experiments were performed using the PLATEMO platform [16]. In this work, the common method of weight vector generations was substituted by the proposed novel method of generating a cone of weight vectors in the MOEA/D algorithm. All MOEA/D methods employ the same following parameters:

- Population size: The population size is $300 + 15 \times M$ individuals, where M is the number of objectives.
- Maximum number of iterations: 500 iterations;
- Genetic operators: SBX recombination ($\mu_c = 20$) and polynomial mutation ($\mu_m = 20$);

In the Decomposition Algorithm, other than the cone of vectors that define the preferences of the DM, other auxiliary weight vector cones were created. This was done by using the vectors of the canonical basis of the decision space as axis. Each extra auxiliary cone consists of 15 vectors, restricted to the first orthant, i.e., for each weight vector in the auxiliary cones $\mathbf{w} = (w_1, \ldots, w_M)$, we define $w_i \geq 0 \;\; i = 1, \ldots, M$. These extra cones are required to guide the population to the correct location indicated by the cone of preferences. Experiments were performed with and without the extra cones and best results were obtained with the use of the auxiliary cones. Figure 3.3b illustrates a weight vector cone and a set of auxiliary cones in the first octant of space $\mathbb{R}^3$. The main cone uses the vector $\mathbf{v} = (1, 1, 1)$ as axis. Each auxiliary cone uses a vector of the canonical bases (i.e. $\mathbf{e_1} = (1, 0, 0)$, $\mathbf{e_2} = (0, 1, 0)$ and $\mathbf{e_3} = (0, 0, 1)$,) as axis. In all cones, the opening angle of $\arccos(1/\sqrt{3})/5$ radians was used. This special value will be discussed in the Sect. 3.4.2.

To analyze the performance of the algorithms, the obtained solutions are classified in three groups according to their convergence in the ROI defined by the preference

Table 3.2 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the $v2$ direction.

		DTLZ1	DTLZ2	WFG1
3 objectives				
IGD	gNSGAII	1.688e+01(3.883e+01)	4.154e−01(1.641e−03)	1.662e+00(2.155e−02)
	MOEA/D	1.457e−01(6.822e−05)	4.084e−01(1.331e−03)	1.664e+00(3.792e−03)
HV	gNSGAII	1.598e−02(2.702e−02)	2.784e−01(2.192e−03)	3.649e+01(1.076e+00)
	MOEA/D	7.676e−02(4.079e−05)	2.882e−01(1.566e−03)	3.954e+01(6.408e−02)
5 objectives				
IGD	gNSGAII	5.840e+02(1.593e+02)	1.825e+00(9.919e−01)	3.287e+00(1.025e+00)
	MOEA/D	1.524e−01(1.473e−04)	5.031e−01(3.834e−04)	1.929e+00(2.632e−02)
HV	gNSGAII	0(0)	1.922e−05(5.783e−05)	1.254e+03(1.287e+03)
	MOEA/D	2.812e−02(9.667e−05)	3.679e−01(1.667e−03)	4.855e+03(8.508e+01)
10 objectives				
IGD	gNSGAII	⋆	7.724e+00(2.670e+00)	7.002e+00(1.568e+00)
	MOEA/D	1.843e−01(2.685e−03)	6.848e−01(3.813e−03)	2.782e+00(6.948e−02)
HV	gNSGAII	⋆	0(0)	1.523e+08(5.646e+08)
	MOEA/D	1.315e−03(5.233e−05)	4.215e−01(1.012e−02)	7.435e+09(7.131e+08)

Table 3.3 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the $v3$ direction

		DTLZ1	DTLZ2	WFG1
3 objectives				
IGD	gNSGAII	4.190e–01(7.944e–01)	5.309e–01(1.846e–03)	9.045e–01(1.002e–02)
	MOEA/D	1.820e–01(9.186e–05)	5.251e–01(1.507e–03)	9.602e–01(8.225e-02)
HV	gNSGAII	5.380e–02(2.732e–02)	2.782e–01(9.709e–04)	4.093e+01(3.744e–01)
	MOEA/D	7.150e–02(6.373e–05)	2.824e–01(1.500e–03)	4.002e+01(1.875e+00)
5 objectives				
IGD	gNSGAII	5.052e+02(1.477e+02)	1.788e+00(8.907e–01)	4.124e+00(1.795e+00)
	MOEA/D	1.652e–01(1.527e–04)	5.590e–01(5.380e–04)	1.608e+00(4.128e–02)
HV	gNSGAII	0(0)	2.732e–05(1.187e–04)	1.028e+03(1.057e+03)
	MOEA/D	2.687e–02(8.523e–05)	3.689e–01(1.858e–03)	4.801e+03(1.573e+02)
10 objectives				
IGD	gNSGAII	⋆	9.493e+00(2.404e+00)	8.698e+00(8.120e–01)
	MOEA/D	1.736e–01(3.873e–04)	7.037e–01(2.822e–03)	2.706e+00(1.346e–01)
HV	gNSGAII	⋆	0(0)	2.028e+07(3.188e+07)
	MOEA/D	1.549e–03(1.956e–05)	4.315e–01(1.074e–02)	6.605e+09(1.106e+09)

cone. The first group consists of solutions located in the region defined by the preference cone, ie, the angle θ between the solution p and the axis $\mathbf{v}$ of the cone is less than or equal to the angle τ that define the cone. The second group is composed of solutions located in the neighborhood of the region defined by the cone. In the experiments performed, a obtained solution is in the group 2 if the angle θ between the obtained solution p and the axis $\mathbf{v}$ of the cone is greater than τ and smaller than 2τ, i.e., $\tau < \theta < 2\tau$. All other solutions are classified in group 3. The classification of solutions into groups aims to verify the ability of each method to obtain solutions that adequately reflect the aspirations of the decision maker. Convergence and distribution of the solutions obtained will be verified through the usual metrics, restricted to solutions obtained in the selected groups.

Table 3.4 IGD and HV metrics for the problems with 3, 5 and 10 objectives in the $v4$ direction

		DTLZ1	DTLZ2	WFG1
3 objectives				
IGD	gNSGAII	7.918e+00(1.558e+01)	4.331e–01(2.067e–03)	1.641e+00(2.516e–02)
	MOEA/D	1.502e–01(8.893e–05)	4.249e–01(1.821e–04)	1.644e+00(4.087e–03)
HV	gNSGAII	1.759e–02(2.640e–02)	2.637e–01(1.860e–03)	3.256e+01(1.655e+00)
	MOEA/D	7.637e–02(6.880e–05)	2.750e–01(3.371e–04)	3.722e+01(1.164e–01)
5 objectives				
IGD	gNSGAII	6.379e+02(1.810e+02)	2.592e+00(1.695e+00)	2.714e+00(6.954e–01)
	MOEA/D	1.576e–01(1.705e–04)	5.292e–01(8.470-e-04)	1.860e+00(4.701e–02)
HV	gNSGAII	0(0)	0(0)	1.724e+03(1.162e+03)
	MOEA/D	2.818e–02(1.272e–04)	3.626e–01(1.533e–03)	4.675e+03(3.658e+01)
10 objectives				
IGD	gNSGAII	8.886e+02(6.025e+01)	8.543e+00(2.488e+00)	4.022e+00(8.363e–01)
	MOEA/D	1.737e–01((7.081e–04)	7.022e–01(2.897e–03)	2.740e+00(9.967e–02)
HV	gNSGAII	0(0)	0(0)	1.452e+09(1.820e+09)
	MOEA/D	1.586e–03(2.086e–05)	4.298e–01(1.136e–02)	7.150e+09(7.259e+08)

For the experiments carried out, we considered only the solutions of group 1 and 2. All solutions p belonging to group 2 are penalized by the factor:

$$r = (\theta - \tau)^2 + e^{(1+\theta-\tau)^2} \tag{3.7}$$

where τ is the angle that defines the preference cone and θ is the angle between the obtained solution p and the axis $\mathbf{v}$ of the preference cone. If p is a solution of group 2, it will be evaluated as $r \cdot p$. The solutions in group 1 and the penalized solutions in group 2 are analyzed using the performance metrics *Inverted Generational Distance* (IGD) [23] and *Hypervolume Indicator* (HV) [22].

3.4.2 ROI Definition

For the decomposition algorithms defines the ROI is calculated by an axis $\mathbf{v}$ and an aperture angle τ. Table 3.5 shows four different directions used as axis of the preference cone.

The chosen directions have the following characteristics:

Table 3.5 Directions used in the experiments

Objectives	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
3	(1, 1, 1)	(1, 1, 0.5)	(0.1, 1, 1)	(2, 1, 1)
5	(1, 1, 1, 1, 1)	(1, 1, 1, 1, 0.5)	(0.1, 1, 1, 1, 1)	(2, 1, 1, 1, 1)
10	$(1, 1, \ldots, 1)$	$(1, \ldots, 1, 0.5)$	$(0.1, 1, \ldots, 1)$	$(2, 1, \ldots, 1)$

Direction $\mathbf{v}_1$: The ROI defined in the direction $\mathbf{v}_1$ uses the hyperdiagonal of the Objective Space. This direction choice seeks to maximize the balance among objectives.

Direction $\mathbf{v}_2$: In direction $\mathbf{v}_2$, the last objective is equal to 0.5 and all the remaining are equal to 1. By choosing these values, the defined ROI will be further away from the last objective, but is balanced in relation to the others.

Direction $\mathbf{v}_3$: The direction $\mathbf{v}_3$ presents a low value in the first coordinate (only 1/10 of the value of the other objectives). As a result, the ROI defined in this direction will contain few points (or no point) in this region.

Direction $\mathbf{v}_4$: In the direction $\mathbf{v}_4$, the first coordinate is equal to 2 and all the others are equal to 1. By this, the defined ROI will be closer to the first objective and farther from the remaining objectives.

ROI size is defined by angle τ. Suposing that all objectives assume nonnegative values, the objective space is in the first orthant of the space $\mathbb{R}^M$. The angle between the hyperdiagonal $\mathbf{v} = (1, \ldots, 1)$ and any vector of the canonical basis $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is equal to $\arccos(1/\sqrt{M})$. This value is used as a reference to define the aperture angle τ of the cone. For example, in a cone in $\mathbb{R}^3$ where the hyperdiagonal $\mathbf{v} = (1, 1, 1)$ and the vector of the canonical basis $\mathbf{e}_1 = (1, 0, 0)$ are opposite generatrices, the angle τ is equal to $\arccos(1/\sqrt{3})/2$. Using a direction vector $\mathbf{v}$ and a τ aperture angle is a simple, intuitive and efficient way to define ROI. Assuming a PF contained in a hypersphere, the ROI defined in this way always yields the same proportion in relation to the complete PF, regardless of the hypersphere radius. For all cones, including auxiliary cones, the angle τ is defined as $\tau = \arccos(1/\sqrt{M})/5$, where M is the number of objectives.

For the g-NSGA-II algorithm, the ROI is defined by a reference point $\mathbf{G}$. Since all problem have the true pareto front well defined, the G point is easy to define: let $\mathcal{G} = \{\mathbf{g}^1 \ldots \mathbf{g}^q\}$ be points in the PF located in the ROI defined by the preference cone and set $\mathbf{G}$ the ideal point of $\mathcal{G}$.

Thirty instances of each algorithm were performed for each ROI. Table 3.5 presents the direction vectors that define the ROI used in the experiments and Table 3.6 presents the success rate of each experiment. An experiment was considered successful if at least one solution of group 1 or 2 was found. Tables 3.1, 3.2, 3.3 and 3.4 present the average and the standard deviation of the IGD and HV metrics of the tests performed, highlighting the best and the worst result. A $\star$ marker is used when no results from group 1 or 2 were found.

Table 3.6 Success rate for the problems with 3, 5 and 10 objectives.

		3 objectives			5 objectives			10 objectives		
		DTLZ1	DTLZ2	WFG1	DTLZ1	DTLZ2	WFG1	DTLZ1	DTLZ2	WFG1
v1	gNSGAII	0.60	1.00	1.00	0.57	1.00	1.00	0.00	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
v2	gNSGAII	1.00	1.00	1.00	0.90	1.00	1.00	0.00	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
v3	gNSGAII	0.87	1.00	1.00	0.70	1.00	1.00	0.00	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
v4	gNSGAII	0.73	1.00	1.00	0.93	1.00	1.00	0.07	1.00	1.00
	MOEA/D	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

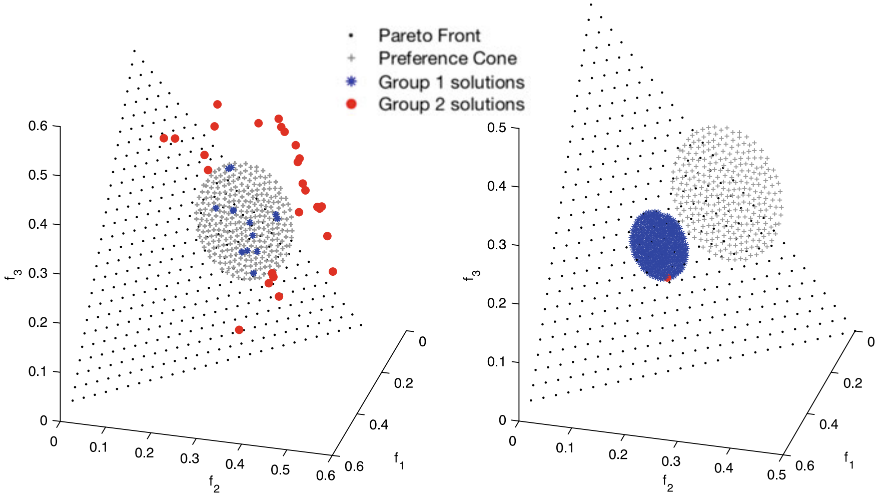
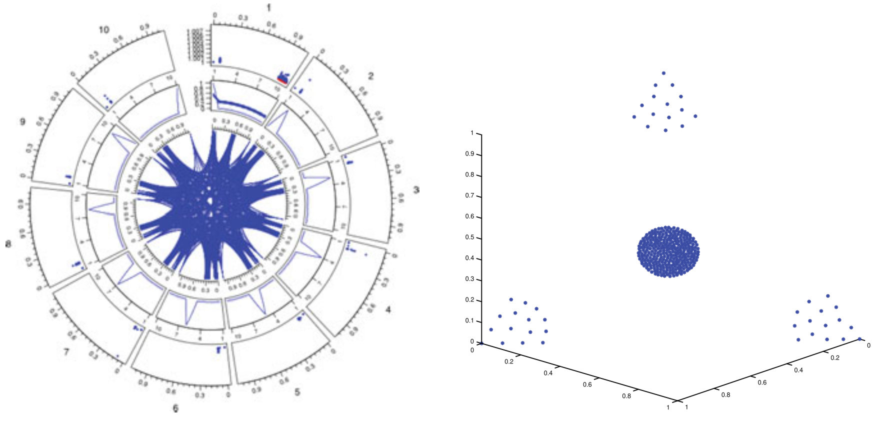


Fig. 3.2 Obtained solutions (blue dots) for the DTLZ1 problem with 3 objectives in $\mathbf{v}_1$ direction

Figure 3.3a presents the obtained solution (blue dots) for the DTLZ2 problem with 10 objectives in $\mathbf{v}_4$ direction and the ROI (red dots) using CAP-vis visualization tool [12]. The points represented at the beginning of track C of each sector represent the solutions found in the auxiliary cones. The ROI is entirely localized in sector 1 and it is possible to observe the alignment of the solutions obtained with this region.

3.4.3 Discussion

The decomposition algorithm proposed in this work, using the cone of weight vectors obtained better results than the dominance algorithm using ASF strategy. Figure 3.2 depicts the obtained solutions of a single run for each method in the v_1 direction, highlighting group 1 and group 2 solutions for the DTLZ1 problem. Algorithm



(a) (b)

Fig. 3.3 **a** obtained solution for the DTLZ2 problem with 10 objectives in v_4 direction. **b** cone and auxilliary cones of weigh vectors in v_1 direction

gNSGAII presented better performance only in the three objective WFG1 problem, regardless of the ROI analyzed. However, in other problems, this method presented a high variance of results in most cases. This phenomenon is caused by the prevalence of solutions of Group 2 which due to its penalty makes the value of its metric increased. Moreover, the results obtained reinforce the inadequacy of the algorithms based on dominance for problems with many objectives.

From the experiments conducted, it can be concluded that different regions of the objective space present different challenges for the optimizer of the same problem. As an example, the value of the IGD metric obtained in the region defined by the direction v_3 in the problem WFG1 indicates that in this region the algorithms used can obtain solutions with better convergence than others directions that were analyzed.

3.5 Conclusions

Due to the fundamental characteristics of many-objective optimization problems, obtaining a well-distributed and representative approximation of the Pareto Front is a hard task. Moreover, the analysis of the solutions obtained and the subsequent choice of a particular solution are challenging. Moreover, defining exactly the preferences of the DM in an a priori approach is difficult in most practical cases. Researching a noncommittal approach becomes attractive in such a scenario, in which the search for solutions in a specific region of the objective space that corresponds to the aspirations of the DM is a way to make this type of optimization problem less difficult.

This paper presented a new preference-based methodology to perform the search for solutions in the ROI, defined by a preference cone in the Objective Space. The exploration of the ROI using the preference cone presented good convergence and dispersion of the solutions, showing that this is an adequate methodology for preference-guided many-objective optimization. In addition, this approach of determining the ROI is more intuitive and is able to reflect the preferences of the decision-maker.

Acknowledgments This work was supported by the Brazilian funding agencies CAPES and CNPq.

References

1. Bader J, Zitzler E (2011) HypE: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19(1):45–76. https://doi.org/10.1162/evco_a_00009
2. Batista LS, Campelo F, Guimarães FG, Ramírez JA (2011) A comparison of dominance criteria in many-objective optimization problems. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp 2359–2366. https://doi.org/10.1109/CEC.2011.5949909
3. Bechikh S, Kessentini M, Said LB, Ghédira K (2015) Preference incorporation in evolutionary multiobjective optimization. In: Hurson AR (ed) Advances in computers, vol 98. Elsevier, pp 141–207. https://doi.org/10.1016/bs.adcom.2015.03.001
4. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5):773–791. https://doi.org/10.1109/tevc.2016.2519378
5. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, USA
6. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601
7. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multi-objective optimization. In: Abraham A, Jain L, Goldberg R (eds) Evolutionary multiobjective optimization: theoretical advances and applications. Springer London, pp 105–145. https://doi.org/10.1007/1-84628-137-7_6
8. Goulart F, Campelo F (2016) Preference-guided evolutionary algorithms for many-objective optimization. Inf Sci 329(Supplement C):236–255. https://doi.org/10.1016/j.ins.2015.09.015, http://www.sciencedirect.com/science/article/pii/S0020025515006696. Special issue on Discovery Science
9. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506. https://doi.org/10.1109/tevc.2005.861417
10. Keršulienė V, Zavadskas EK, Turskis Z (2010) Selection of rational dispute resolution method by applying new step-wise weight assessment ratio analysis (swara). J Bus Econ Manag 11(2):243–258. https://doi.org/10.3846/jbem.2010.12
11. Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: a survey. ACM Comput Surv 48(1):1–35. https://doi.org/10.1145/2792984
12. Meneghini IR, Koochaksaraei RH, Guimarães FG, Gaspar-Cunha A (2018) Information to the eye of the beholder: data visualization for many-objective optimization. In: 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE. https://doi.org/10.1109/cec.2018.8477889
13. Molina J, Santana LV, Hernández-Díaz AG, Coello CAC, Caballero R (2009) g-dominance: Reference point based dominance for multiobjective metaheuristics. Eur J Oper Res 197(2):685–692. https://doi.org/10.1016/j.ejor.2008.07.015
14. Ruiz AB, Saborido R, Luque M (2014) A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. J Global Optim 62(1):101–129. https://doi.org/10.1007/s10898-014-0214-y

15. Saaty TL (1980) The analytic hierarchy process: planning, priority setting, resource allocation. McGraw-Hill International Book Co., New York
16. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87. https://doi.org/10.1109/mci.2017.2742868
17. Trivedi A, Srinivasan D, Sanyal K, Ghosh A (2017) A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Trans Evol Comput 21(3):440–462. https://doi.org/10.1109/tevc.2016.2608507
18. Zhang Q, Li H (2007) MOEA/d: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731. https://doi.org/10.1109/tevc.2007.892759
19. Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol Comput 1(1):32–49. https://doi.org/10.1016/j.swevo.2011.03.001
20. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: Conference on Parallel Problem Solving from Nature (PPSN VIII). Springer, Berlin, pp 832–842
21. Zitzler E, Laumanns M, Thiele L (2002) SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou K et al (eds) Evolutionary methods for design, optimisation and control with application to industrial problems (EUROGEN 2001). International Center for Numerical Methods in Engineering (CIMNE), pp 95–100
22. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben AE, Bäck T, Schoenauer M, Schwefel HP (eds) Lecture notes in computer science. Springer, Heidelberg, pp 292–301. https://doi.org/10.1007/bfb0056872
23. Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132. https://doi.org/10.1109/tevc.2003.810758

Chapter 4
Solving Multiobjective Engineering Design Problems Through a Scalarized Augmented Lagrangian Algorithm (SCAL)

Lino Costa, Isabel Espírito Santo, and Pedro Oliveira

Abstract In this paper, a set of multiobjective engineering design problems is solved using a methodology that combines an Augmented Lagrangian technique to deal with the constraints and the Augmented Weighted Tchebycheff method to tackle the multiobjective nature of the problems to find the Pareto frontier. In order to compare and validate the performance of this strategy, the problems were also solved with `gamultiobj` from MATLAB™. We present the algorithm, as well as some results that seem very promising.

Keywords Engineering optimization · Multi-objective constrained optimization · Augmented weighted tchebycheff · Augmented lagrangian · Pattern search

4.1 Introduction

In multiobjective optimization no single optimal solution exists but a set of solutions that reflect different trade-offs between the different objectives. This problem is even difficult when constraints are imposed on the search space. Although several approaches exist for the handling of constraints in the context of multiobjective optimization, this is still an important issue.

Considering a set of n decision variables and a set of m objective functions, a multiobjective problem can be formulated as

L. Costa (✉) · I. Espírito Santo
ALGORITMI Center, University of Minho, Campus Gualtar, 4710-057 Braga, Portugal
e-mail: lac@dps.uminho.pt

I. Espírito Santo
e-mail: iapinho@dps.uminho.pt

P. Oliveira
EPIUnit, Instituto de Ciencias Biomedicas Abel Salazar, Universidade do Porto, Rua das Taipas, n 135, 4050-600 Porto, Portugal
e-mail: pnoliveira@icbas.up.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_4

$$\begin{array}{ll}\text{minimize:} & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^{\mathrm{T}} \\ \text{subject to:} & \\ & \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^n \end{array} \tag{4.1}$$

where $\boldsymbol{x}$ is the decision vector, $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0}, \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{0}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$ is the feasible decision space, $\boldsymbol{f}(\boldsymbol{x})$ is the objective vector defined in the objective space $\mathbb{R}^m$, $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0}$ are q equality constraints and $\boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{0}$ are p inequality constraints.

Inherent to the multiobjective nature of the problem, a solution represents different trade-offs between the different objectives and, therefore, the solution space is partially ordered. Accordingly, for two solutions $\boldsymbol{a}$ and $\boldsymbol{b}$, a solution $\boldsymbol{a}$ dominates solution $\boldsymbol{b}$ (denoted $\boldsymbol{a} \prec \boldsymbol{b}$) if

$$\forall i \in \{1, \ldots, m\} : f_i(\boldsymbol{a}) \leq f_i(\boldsymbol{b}) \wedge \exists j \in \{1, \ldots, m\} : f_j(\boldsymbol{a}) < f_j(\boldsymbol{b}). \tag{4.2}$$

Hence, the set of non-dominated solutions constitutes the so-called Pareto set (PS), being its approximation the fundamental task of a multiobjective algorithm:

$$PS = \{\boldsymbol{x} \in \Omega \mid \nexists \boldsymbol{y} \in \Omega : \boldsymbol{y} \prec \boldsymbol{x}\}. \tag{4.3}$$

Real world engineering problems are constrained in many different ways. Solving constrained multiobjective problems constitutes a major challenge, in particular, with respect to equality constraints. The presence of these constraints can alter greatly the Pareto frontier, making its approximation a very challenging problem. In this work, several engineering constrained multiobjective optimization problems are solved using a Scalarized Augmented Lagrangian Algorithm (SCAL) [4].

4.2 Augmented Weighted Tchebycheff Methods

In order to approximate the Pareto-optimal set, a scalarizing method is used, the Augmented Weighted Tchebycheff method, based on the proposal of Steuer and Choo [15]. This approach can be applied to nonlinear and nonconvex multiobjective optimization problems, and, at the same time, converge to non-extreme final solutions.

The Augmented Weighted Tchebycheff method can be formulated as:

$$\min \overline{f}(\boldsymbol{x}) \equiv \max_{i=1,\ldots,m} [w_i |f_i(\boldsymbol{x}) - z_i^\star|] + \rho \sum_{i=1}^{m} |f_i(\boldsymbol{x}) - z_i^\star| \tag{4.4}$$

where w_i are the weighting coefficients for objective i, $z_i^\star$ are the components of the ideal point, and ρ is a small positive value [6]. The problem formulated in Eq. 4.4 guarantees that the Pareto-optimal solutions are obtained by considering different combinations of weights. An approximation to the ideal vector is used as reference point $\boldsymbol{z}^\star = (z_1^\star, \ldots, z_m^\star)^T = (\min f_1, \ldots, \min f_m)^T$.

4.3 Augmented Lagrangian Technique Using the Hooke and Jeeves Pattern Search Method

In this work, a sequence of simple subproblems, defined by the Augmented Lagrangian technique, results in a objective function that considers the penalization of all or some of the constraint violation. Thus, the Augmented Lagrangian objective function, based on the proposal of [1, 3, 12] depends on a penalty parameter and multiplier vectors:

$$\Phi(\boldsymbol{x}; \boldsymbol{\lambda}, \boldsymbol{\delta}, \mu) = \overline{f}(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{x}) + \frac{1}{2\mu}\|\boldsymbol{c}(\boldsymbol{x})\|^2 + \frac{\mu}{2}\left(\left\|\left[\boldsymbol{\delta} + \frac{\boldsymbol{g}(\boldsymbol{x})}{\mu}\right]_+\right\|^2 - \|\boldsymbol{\delta}\|^2\right) \tag{4.5}$$

where μ is a positive penalty scalar, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_q)^T$, $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_p)^T$ are the Lagrange multiplier vectors associated with the equality and inequality constraints, respectively, and $\|.\|$ is the euclidean norm. Hence, function Φ penalizes equality and inequality constraints. Simple bounds, $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$, are not incorporated but the inner iterative process returns an approximate solution satisfying the bound constraints. Thus, the Hooke and Jeeves version of the pattern search [10] assures that any solution $\boldsymbol{x}$ that does not satisfy the bounds is projected, component by component.

The corresponding subproblem is formulated as:

$$\underset{\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}}{\text{minimize}} \ \ \Phi(\boldsymbol{x}; \boldsymbol{\lambda}^j, \boldsymbol{\delta}^j, \mu^j) \tag{4.6}$$

where, for each set of fixed $\boldsymbol{\lambda}^j$, $\boldsymbol{\delta}^j$ and μ^j, the solution of subproblem (4.6) provides an approximation $\boldsymbol{x}^j$ to the problem formulated in Eq. (4.4). Details of the approach can be found in [1]. In order to maintain the sequence of penalty parameters far away from zero usual safeguarded schemes are used, so that the solution of subproblem (4.6) is an easy task.

The following error function evaluates the equality and inequality constraint violation, and the complementarity conditions:

$$E(\boldsymbol{x}, \boldsymbol{\delta}) = \max\left\{\frac{\|\boldsymbol{c}(\boldsymbol{x})\|_\infty}{1 + \|\boldsymbol{x}\|}, \frac{\|[\boldsymbol{g}(\boldsymbol{x})]_+\|_\infty}{1 + \|\boldsymbol{\delta}\|}, \frac{\max_i \delta_i |g_i(\boldsymbol{x})|}{1 + \|\boldsymbol{\delta}\|}\right\}. \tag{4.7}$$

The Lagrange multipliers $\boldsymbol{\lambda}^j$ and $\boldsymbol{\delta}^j$ are estimated in this iterative process using the first-order updating formulae

$$\bar{\lambda}_i^{j+1} = \lambda_i^j + \frac{c_i(\boldsymbol{x}^j)}{\mu^j}, \ \ i = 1, \ldots, q \tag{4.8}$$

and

$$\bar{\delta}_i^{j+1} = \max\left\{0, \delta_i^j + \frac{g_i(\boldsymbol{x}^j)}{\mu^j}\right\}, \quad i = 1, \ldots, p \tag{4.9}$$

where: for all $j \in \mathbb{N}$, and for $i = 1, \ldots, q$ and $l = 1, \ldots, p$, λ_i^{j+1} is the projection of $\bar{\lambda}_i^{j+1}$ on the interval $[\lambda_{\min}, \lambda_{\max}]$ and δ_i^{j+1} is the projection of $\bar{\delta}_i^{j+1}$ on the interval $[0, \delta_{\max}]$, where $-\infty < \lambda_{\min} \leq \lambda_{\max} < \infty$ and $0 \leq \delta_{\max} < \infty$. After the new approximation $\mathbf{x}^j$ has been computed, the Lagrange multiplier vector $\boldsymbol{\delta}$ associated with the inequality constraints is updated, in all iterations, since $\boldsymbol{\delta}^{j+1}$ is required in the error function (4.7) to measure constraint violation and complementarity. We note that the Lagrange multipliers $\lambda_i, i = 1, \ldots, q$ are updated only when feasibility and complementarity are at a satisfactory level, herein defined by the condition

$$E(\boldsymbol{x}^j, \boldsymbol{\delta}^{j+1}) \leq \eta^j \tag{4.10}$$

for a positive tolerance η^j. It is required that $\{\eta^j\}$ defines a decreasing sequence of positive values converging to zero, as $j \to \infty$. This is easily achieved by $\eta^{j+1} = \pi\eta^j$ for $0 < \pi < 1$.

We consider that an iteration j failed to provide an approximation $\boldsymbol{x}^j$ with an appropriate level of feasibility and complementarity if condition (4.10) does not hold. In this case, the penalty parameter is decreased using $\mu^{j+1} = \gamma\mu^j$ where $0 < \gamma < 1$. When condition (4.10) holds, then the iteration is considered satisfactory. This condition says that the iterate $\mathbf{x}^j$ is feasible and the complementarity condition is satisfied within some tolerance η^j and, consequently, the algorithm maintains the penalty parameter value. The sequence of penalty parameters tends to zero if condition (4.10) is not observed infinitely many times. To prevent that the sequence $\{\mu^j\}$ reaches zero, the following update is used

$$\mu^{j+1} = \max\{\mu_{\min}, \gamma\mu^j\}, \tag{4.11}$$

where $\mu_{\min}$ is a sufficiently small positive real value.

Hooke and Jeeves (HJ) search method [10] is used to compute $\Delta^k \boldsymbol{s}^k$. The scalar Δ^k represents the step length and the vector $\boldsymbol{s}^k$ determines the direction of the step. The exploratory moves to produce $\Delta^k \boldsymbol{s}^k$ and the updating of Δ^k and $\boldsymbol{s}^k$ define a particular pattern search method and their choices are crucial to the success of the algorithm. Two types of moves, the exploratory move and the pattern move, distinguish this algorithm from the traditional coordinate search. An exploratory move is a coordinate search—a search along the coordinate axes—around a selected approximation, using a step length Δ^k. Being $\boldsymbol{z}^k$ the current approximation, a pattern move is a promising direction that is defined by $\boldsymbol{z}^k - \boldsymbol{z}^{k-1}$ when the previous iteration was successful and $\boldsymbol{z}^k$ was accepted as the new approximation. Thus, $\boldsymbol{z}^k + (\boldsymbol{z}^k - \boldsymbol{z}^{k-1})$ defines a new trial approximation, around which an exploratory move is performed. Being successful the new approximation is accepted as $\boldsymbol{z}^{k+1}$ (please refer to [10, 11] for details).

The HJ iterative process provides a new approximation $\boldsymbol{x}^j$ to the problems (4.4), $\boldsymbol{x}^j \leftarrow z^{k+1}$, when the following stopping condition is satisfied, $\Delta^k \leq \varepsilon^j$. If the process fails to satisfy this condition in $k_{\max}$ iterations, the last available approximation is maintained (implementation details are provided in [5]).

4.4 Engineering Design Problems

To test our solver, we selected 10 well-known two objectives engineering design problems. Table 4.1 summarizes the characteristics of these problems with the number of decision variables (n), and the number of constraints (m). All decision variables have simple bounds. Some of these problems have discrete decision variables. However, we chose to solve the relaxed problem since the main goal was to compare the SCAL and `gamultiobj`.

4.4.1 Four-Bar Plane Truss

The objective functions are the structural volume and joint displacement Δ (Fig. 4.1) [2]. The simple bounds of this problem are related to member stresses. The decision variables are the areas of the member cross-sections.

$$\min f_1(\boldsymbol{x}) = L\left(2x_1 + \sqrt{2x_2} + \sqrt{x_3} + x_4\right)$$
$$\min f_2(\boldsymbol{x}) = \frac{FL}{E}\left(\frac{2}{x_2} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right)$$

Table 4.1 Engineering design problems

Problem	n	m
Four-bar plane truss	4	0
Cantilever beam	2	2
Disc break	4	5
I-Beam	4	1
Pressure vessel	4	2
Speed reducer	7	11
Two-bar truss	3	3
Welded beam	4	4
Spring	3	7
Gear train	4	0

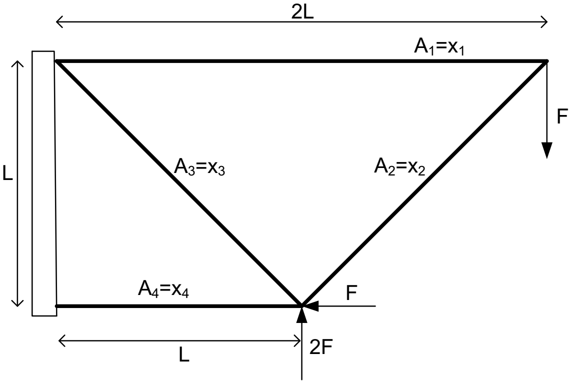


Fig. 4.1 Four-bar truss problem

subject to

$$\frac{F}{\tau} \le x_1 \le 3\frac{F}{\tau}, \quad \sqrt{2}\frac{F}{\tau} \le x_2 \le 3\frac{F}{\tau}, \quad \sqrt{2}\frac{F}{\tau} \le x_3 \le 3\frac{F}{\tau}, \quad \frac{F}{\tau} \le x_4 \le 3\frac{F}{\tau}$$

where $F = 10$ kN, $\tau = 10$ kN/cm^3, $L = 200$ cm, and $E = 2 \times 10^5$ kN/cm^2.

4.4.2 Cantilever Beam

The objective functions are the weight and deflection [14]. This problem has four constraints related to maximum stress. The decision variables are the diameter (x_1) and length (x_2) (Fig. 4.2).

$$\begin{aligned} \min f_1(\boldsymbol{x}) &= \rho\pi\frac{x_1}{4x_2} \\ \min f_2(\boldsymbol{x}) &= \delta \end{aligned}$$

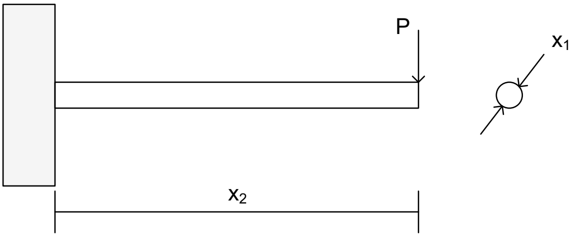


Fig. 4.2 Cantilever problem

subject to

$$\sigma_{\max} \le S_y,\ \ \delta \le \delta_{\max},\ \ 10 \le x_1 \le 50,\ \ 200 \le x_2 \le 1000$$

where $\sigma_{\max} = \dfrac{32Px_2}{\pi x_1^3}$ and $\delta = \dfrac{64Px_2^3}{3E\pi x_1^4}$, $\rho = 7800$ kg/m^3, $P = 1$ kN, $E = 207$ GPa, $Sy = 300$ MPa, and $\delta_{\max} = 5$ mm.

4.4.3 Disc Break

The objective functions are the mass and the stopping time [9]. This problem has nine constraints related to the minimum distance between the radii, the maximum length of the brake, pressure, temperature and torque limitations. The decision variables are the inner radius of the discs (x_1), the outer radius of the discs (x_2), the engaging force (x_3), and the number of friction surfaces and length (x_4).

$$\begin{aligned}
\min f_1(\boldsymbol{x}) &= 4.9 \times 10^{-5}\left(x_2^2 - x_1^2\right)(x_4 - 1)\\
\min f_2(\boldsymbol{x}) &= \frac{9.82 \times 10^{6}\left(x_2^2 - x_1^2\right)}{x_3 x_4\left(x_2^3 - x_1^3\right)}
\end{aligned}$$

subject to

$$\begin{aligned}
&x_2 - x_1 \ge 20, \quad 2.5(x_4 + 1) \le 30, \quad \frac{x_3}{3.14\left(x_2^2 - x_1^2\right)} \le 0.4,\\
&\frac{2.22 \times 10^{-3} x_3\left(x_2^3 - x_1^3\right)}{\left(x_2^2 - x_1^2\right)^2} \le 1, \quad \frac{2.66 \times 10^{-2} x_3 x_4\left(x_2^3 - x_1^3\right)}{x_2^2 - x_1^2} \ge 900,\\
&55 \le x_1 \le 80, \quad 75 \le x_2 \le 110, \quad 1000 \le x_3 \le 3000, \quad 2 \le x_4 \le 20
\end{aligned}$$

4.4.4 I-Beam

The objective functions are the cross-sectional area of the beam and the static deflection of the I-Beam [8]. This problem has five constraints related to geometry and strength. The decision variables are the four dimensions of the I-Beam $(x_1, \ldots, x_4)$ (Fig. 4.3).

$$\begin{aligned}
\min f_1(\boldsymbol{x}) &= 2x_2x_4 + x_3\ \ (x_1 - 2x_4)\\
\min f_2(\boldsymbol{x}) &= \frac{PL^3}{48EI}
\end{aligned}$$

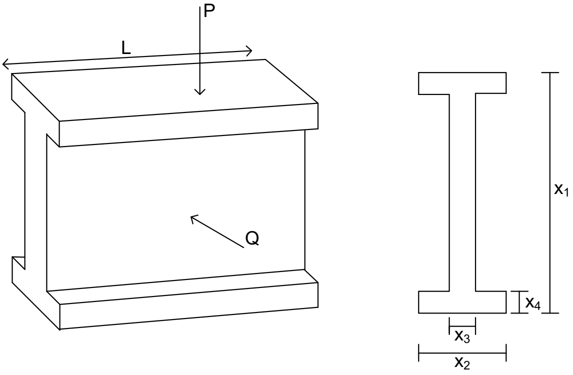


Fig. 4.3 I-Beam problem

subject to

$$\frac{M_y}{Z_y} + \frac{M_z}{Z_z} \leq \sigma_a \quad 10 \leq x_1 \leq 80, \quad 10 \leq x_2 \leq 50, \quad 0.9 \leq x_3 \leq 5, \quad 0.9 \leq x_4 \leq 5$$

where

$I = \frac{1}{12}\left(x_3\,(x_1 - 2x_4)^3 + 2x_2x_4\left(4x_4^2 + 3x_1\,(x_1 - 2x_4)\right)\right),\; M_y = 0.25PL,\; M_z = 0.25QL,$
$Z_y = \frac{1}{6x_1}\left(x_3\,(x_1 - x_4)^3 + 2x_2x_4\left(4x_4^2 + 3x_1\,(x_1 - 2x_4)\right)\right),$
$Z_z = \frac{1}{6x_2}\left((x_1 - x_4)\,x_3^3 + 2x_4x_2^3\right), E = 2 \times 10^4\,\text{kN/cm}^2, P = 600\,\text{kN}, L = 200\,\text{cm},$
$Q = 50$ kN, and $\sigma_a = 16$ kN/cm^2.

4.4.5 *Pressure Vessel*

The objective functions are the total cost and the storage capacity of a cylindrical vessel capped at both ends by hemispherical heads [13]. This problem has two constraints related to the dimensions. The decision variables are the thickness of the shell (x_1), the thickness of the head (x_2), the inner radius (x_3) and the length of the cylindrical section not including the head (x_4) (Fig. 4.4).

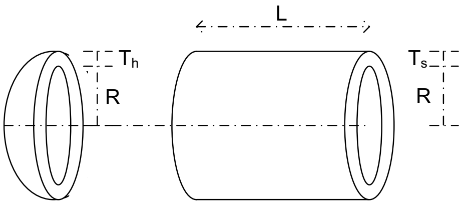


Fig. 4.4 Pressure vessel problem

$$\begin{aligned}\min f_1(\boldsymbol{x}) &= 0.6224x_1x_4x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3\\ \max f_2(\boldsymbol{x}) &= \pi x_3^2x_4 + 1.333\pi x_3^3\end{aligned}$$

subject to

$$\begin{aligned}&0.0193x_3 - x_1 \le 0, \quad 0.00954x_3 - x_2 \le 0\\ &0.0625 \le x_1 \le 5, \quad 0.0625 \le x_2 \le 5, \quad 10 \le x_3 \le 200, \quad 10 \le x_4 \le 240\end{aligned}$$

4.4.6 Speed Reducer

The objective functions are the weight of the gear assembly and the transverse deflection of the shaft [16]. This problem has 18 constraints related to the bending stress of the gear teeth, surfaces stress, transverse deflections of the shafts and stresses in the shafts. The decision variables are the face width (x_1), module of teeth (x_2), number of teeth in the pinion (x_3), length of the first shaft between bearings (x_4), length of the second shaft between bearings (x_5) and the diameter of the first (x_6) and second (x_7) shafts respectively (Fig. 4.5).

$$\begin{aligned}\min f_1(\boldsymbol{x}) = {} & 0.7854x_1x_2^2\left(3.333x_3^2 + 14.933x_3 - 43.0934\right)\\ & - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.477\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)\\ \min f_2(\boldsymbol{x}) = {} & \frac{\sqrt{\left(745\frac{x_4}{x_2x_3}\right)^2 + 1.69 \times 10^7}}{0.1x_6^3}\end{aligned}$$

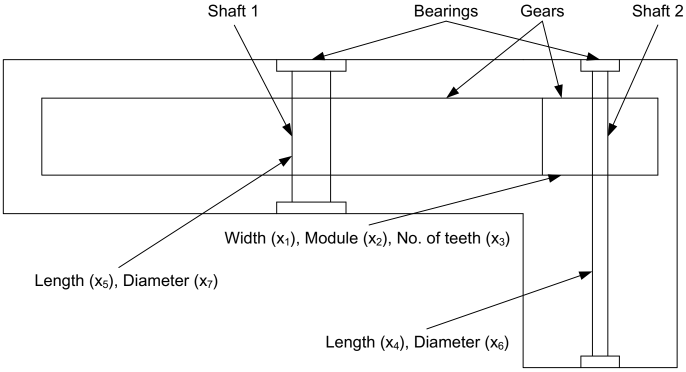

Shaft 1
Bearings
Gears
Shaft 2
Width (x₁), Module (x₂), No. of teeth (x₃)
Length (x₅), Diameter (x₇)
Length (x₄), Diameter (x₆)

Fig. 4.5 Speed reducer problem

subject to

$$\frac{1}{x_1x_2^2x_3} \le \frac{1}{27}, \quad \frac{1}{x_1x_2^2x_3^2} \le \frac{1}{397.5}, \quad \frac{x_4^3}{x_2x_3x_6^4} \le \frac{1}{1.93}, \quad \frac{x_5^3}{x_2x_3x_7^4} \le \frac{1}{1.93},$$
$$x_2x_3 \le 40, \quad \frac{x_1}{x_2} \le 12, \quad \frac{x_1}{x_2} \ge 5, \quad x_4 - 1.5x_6 \ge 1.9, \quad x_5 - 1.1x_7 \ge 1.9$$
$$\frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69 \times 10^7}}{0.1x_6^3} \le 1300, \quad \frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 1.575 \times 10^8}}{0.1x_7^3} \le 1100$$

$$2.6 \le x_1 \le 3.6, \quad 0.7 \le x_2 \le 0.8, \quad 17 \le x_3 \le 28, \quad 7.3 \le x_4 \le 8.3,$$
$$7.3 \le x_5 \le 8.3, \quad 2.9 \le x_6 \le 3.9, \quad 5 \le x_7 \le 5.5$$

4.4.7 Two-Bar Truss

The objective functions are the truss volume and the stress in the longer bar [16]. This problem has 6 constraints related to the truss volume, the stress in the longer and shorter bar. The decision variables are the length of the longer bar (x_1), the length of the shorter bar (x_2), and the vertical distance between the fixation point and the point where the force is applied (x_3) (Fig. 4.6).

$$\min f_1(\boldsymbol{x}) = x_1\left(16 + x_3^2\right)^{0.5} + x_2\left(1 + x_3^2\right)^{0.5}$$
$$\min f_2(\boldsymbol{x}) = \frac{20\left(16 + x_3^2\right)^{0.5}}{x_1x_3}$$

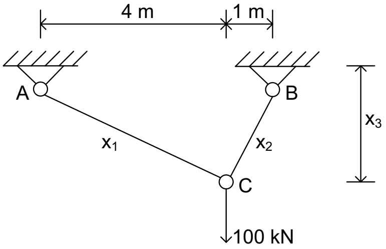


Fig. 4.6 Two-bar truss problem

subject to

$$x_1\left(16+x_3^2\right)^{0.5}+x_2\left(1+x_3^2\right)^{0.5}\le 0.1,\quad \frac{20\left(16+x_3^2\right)^{0.5}}{x_1x_3}\le 100000$$
$$\frac{80\left(1+x_3^2\right)^{0.5}}{x_2x_3}\le 100000,\quad 0\le x_1\le 25,\quad 0\le x_2\le 10,\quad 1\le x_3\le 3$$

4.4.8 Welded Beam

The objective functions are the cost and the end deflection [16]. This problem has eight constraints related to the shear and bending stress, weld length and buckling load. The decision variables are the height (x_1) and the length (x_2) of the welded joint and the thickness (x_3), and the width of the beam (x_4) (Fig. 4.7).

$$\begin{aligned}&\min f_1(\boldsymbol{x}) = 1.10471x_1^2x_2+0.04811x_3x_4\left(14+x_2\right)\\&\max f_2(\boldsymbol{x})=\delta(\boldsymbol{x})\end{aligned}$$

subject to

$$\tau(\boldsymbol{x})\le\tau_{\max},\quad \sigma(\boldsymbol{x})\le\sigma_{\max},\quad x_1-x_4\le 0,\quad P_c(\boldsymbol{x})\ge P,\quad 0.125\le x_1\le 5,$$
$$0.1\le x_2\le 10,\quad 0.1\le x_3\le 10,\quad 0.125\le x_4\le 5$$

where

$$\tau(\boldsymbol{x})=\sqrt{(\tau'(\boldsymbol{x}))^2+\frac{2\tau'(x)\tau''(\boldsymbol{x})x_2}{2R}+(\tau''(\boldsymbol{x}))^2},\quad \tau'(\boldsymbol{x})=\frac{P}{\sqrt{2}x_1x_2},$$
$$\tau''(\boldsymbol{x})=\frac{M(\boldsymbol{x})R(\boldsymbol{x})}{J(\boldsymbol{x})},$$
$$M(\boldsymbol{x})=P(\boldsymbol{x})\left(L+\frac{x_2}{2}\right),\ R(\boldsymbol{x})=\sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2},$$
$$J(\boldsymbol{x})=2\frac{x_1x_2}{\sqrt{2}}\left(\frac{x_2^2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right),\ \sigma(\boldsymbol{x})=\frac{6PL}{x_4x_3^2},\quad \delta(\boldsymbol{x})=\frac{4PL^3}{Ex_4x_3^3},$$
$$P_c(\boldsymbol{x})=\frac{4.013\sqrt{\frac{EGx_3^2x_4^6}{36}}}{L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\ P=6000\text{ lb},\ L=14\text{ in},\ \delta_{\max}=0.25\text{ in},$$
$E=30\times10^6$ psi, $G=12\times10^6$ psi, $\tau_{\max}=13600$ psi, and $\sigma_{\max}=30000$ psi.

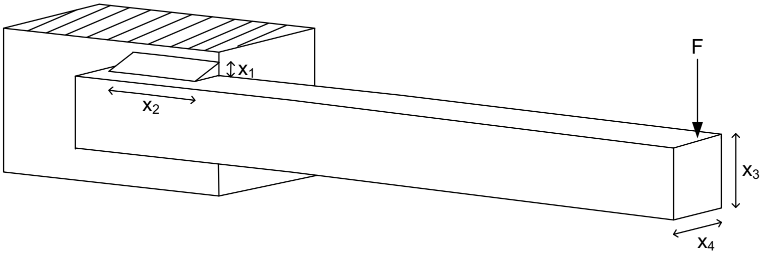


Fig. 4.7 Welded beam problem

4.4.9 Spring

The objective functions are the stress developed due to the application of a load and the volume of the spring [7]. This problem has ten constraints related to the minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The decision variables are the number of active coils (x_1), the wire diameter (x_2), and mean coil diameter (x_3) (Fig. 4.8).

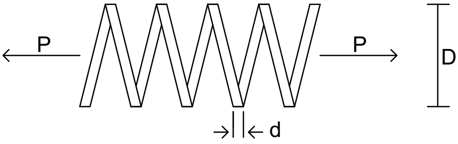
$$\begin{aligned}
&\min f_1(\boldsymbol{x}) = 0.25\pi^2 x_2^2 x_3\,(x_1+2)\\
&\max f_2(\boldsymbol{x}) = \frac{8\,K\,P_{\max} x_3}{\pi x_2^3)}
\end{aligned}$$

subject to

$$\begin{aligned}
&1.05x_2\,(x_1+2) + \frac{P_{\max}}{k} \le l_{\max}, \quad x_2 + x_3 \le D_{\max}, \quad C \ge 3, \quad \delta_p \le \delta_{pm},\\
&\frac{P_{\max} - P}{k} \ge \delta_w, \quad 8\,K\,P_{\max}\frac{x_3}{\pi x_2^3} \le S, \quad 0.25\pi^2 x_2^2 x_3\,(x_1+2) \le V_{\max},\\
&1 \le x_1 \le 32, \quad d_{\min} \le x_2 \le d_{\max}, \quad 1 \le x_3 \le 30
\end{aligned}$$

where $P = 300\,\text{lb}$, $D_{\max} = 3\,\text{in}$, $V_{\max} = 30\,\text{in}^3$, $P_{\max} = 1000\,\text{lb}$, $\delta_w = 1.25\,\text{in}$, $l_{\max} = 14$ in, $\delta_{pm} = 6$ in, $\delta_{\min} = 0.2$ in, $\delta_{\max} = 0.5$ in, $S = 189$ si, $G = 11500000$ lb/in^2, $C = \frac{x_3}{x_2}$, $k = G\frac{x_2^4}{8x_1x_3^3}$, $\delta_p = \frac{P}{k}$, and $K = \frac{4C-1}{4*C-4} + \frac{0.615x_2}{x_3}$.

Fig. 4.8 Spring problem



4.4.10 Gear Train

The objective functions are the gear ratio error with reference gear ratio $\frac{1}{6.931}$ and the maximum size of any of the gear [16]. The decision variables are the number of teeth on each gear $(x_1, \dots, x_4)$.

$$\begin{aligned} \min f_1(\boldsymbol{x}) &= \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right) \\ \min f_2(\boldsymbol{x}) &= \max\{x_1, x_2, x_3, x_4\} \end{aligned}$$

subject to

$$12 \le x_1 \le 60,\ 12 \le x_2 \le 60,\ 12 \le x_3 \le 60,\ 12 \le x_4 \le 60$$

4.5 Results and Discussion

The problems were coded and run in MATLAB™programming language, version R2017a. For each problem, 30 independent runs were performed using SCAL (coded in MATLAB™) and `gamultiobj` (Global Optimization Toolbox from MATLAB™). Table 4.2 shows the parameters of the augmented Lagrangian used in all experiments. The objective functions were normalized using the ideal and nadir vectors. The weights were uniformly varied according to a number of subintervals of 30, i.e., $(w_1, w_2) \in \{(0, 1), (0.03, 0.97), \dots, (1, 0)\}$. The parameter ρ for the Augmented Weighted Tchebycheff method was 10^{-8}. The maximum number of iterations $k_{\max}$ for Hooke and Jeeves pattern search was set on 100 iterations. The maximum number of function evaluations was 20,000. The solutions with a constraint violation superior to 10^{-3} were considered infeasible. For a fair comparison, we set the population size of `gamultiobj` to 50. The remainder parameters were set to the default values.

The hypervolume indicator was used to measure the performance of the algorithms [17]. The higher the value of hypervolume, the more preferable an approximation set is. This measure evaluates algorithm performance in terms of convergence to the Pareto front as well as the diversity of the approximation along the frontier.

Table 4.3 presents the statistical performance comparison of SCAL and `gamultiobj` in terms of the average and standard deviation hypervolume values for 30 runs. The p-values of the pairwise comparison of the outcomes of the algorithms

Table 4.2 Augmented Lagrangian parameters.

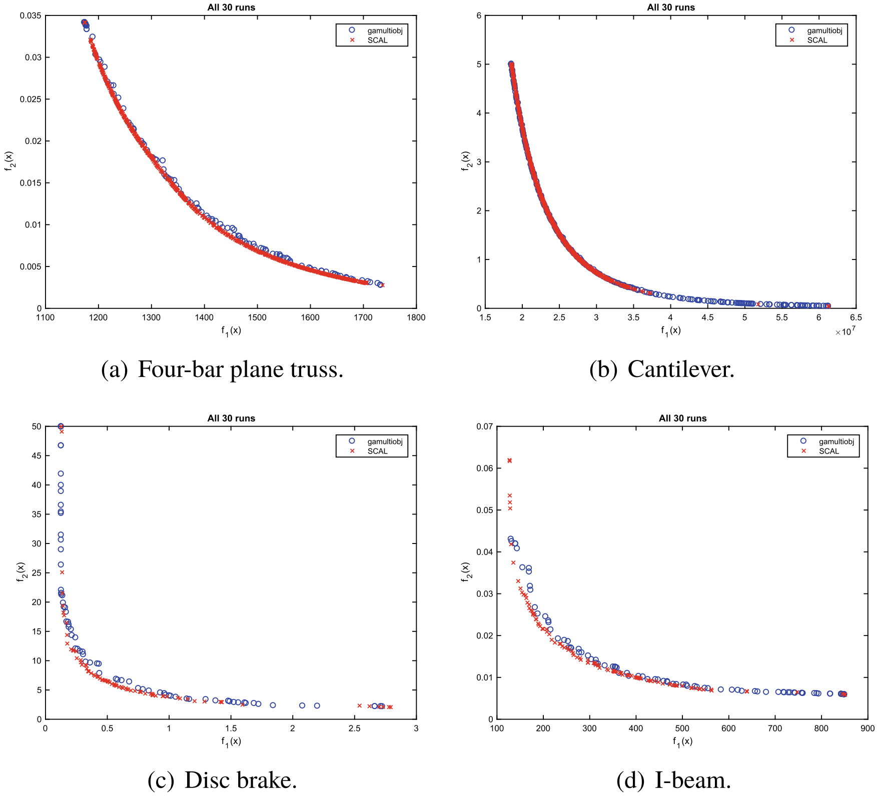
$\lambda_{\min}$	$\lambda_{\max}$	$\delta_{\max}$	μ^1	$\mu_{\min}$	γ	ε^*	η^*	$\lambda_i^1, \delta_i^1, \forall i$	η^1	$j_{\max}$	π	τ
-10^{12}	10^{12}	10^{12}	1	10^{-12}	0.5	10^{-12}	10^{-6}	0	1	300	0.5	0.5

Table 4.3 Performance of problems in terms of the average and standard deviation values of Hypervolume

Problem	SCAL		gamultiobj		
	Average	Std. Dev.	Average	Std. Dev.	*p*-value
Four-bar plane truss	**3.915×10^1**	8.283×10^{-2}	3.795×10^1	6.470×10^{-1}	3.020×10^{-11}
Cantilever beam	7.517×10^8	6.315×10^7	**8.001×10^8**	1.470×10^7	9.514×10^{-6}
Disc brake	1.443×10^2	2.764×10^0	1.454×10^2	1.230×10^0	3.112×10^{-1}
I-beam	**8.083×10^1**	1.201×10^0	7.840×10^1	2.043×10^0	1.254×10^{-7}
Pressure vessel	**2.815×10^{13}**	7.928×10^{11}	2.425×10^{13}	2.474×10^{12}	1.957×10^{-10}
Speed reducer	**1.141×10^6**	4.476×10^3	1.433×10^5	3.560×10^5	3.168×10^{-11}
Two-bar truss	**7.585×10^3**	1.899×10^2	0.000×10^0	0.000×10^0	1.212×10^{-12}
Welded beam	8.925×10^0	1.364×10^{-1}	**9.010×10^0**	1.175×10^{-1}	1.273×10^{-2}
Spring	**2.530×10^6**	3.032×10^4	2.499×10^6	3.866×10^4	6.765×10^{-5}
Gear train	**3.285×10^1**	4.702×10^{-2}	3.224×10^1	1.630×10^{-1}	3.020×10^{-11}

using a standard two-sided Wilcoxon rank sum test is also indicated ($\alpha = 0.05$). Figures 4.9 and 4.10 show the non-dominated feasible solutions obtained by the two algorithms for 30 runs when solving the considered engineering optimization problems.

For the four-bar plane truss, the i-beam, the pressure vessel, the speed reducer, the spring and gear train problems, the obtained hypervolume is significantly better for SCAL (Table 4.3). This result can be confirmed by Figs. 4.9a, d, 4.10a, b, e, f where a much better distribution of the non-dominated solutions can be observed. Also, these solutions dominate a considerable number of the solutions obtained by `gamultiobj`. Thus, SCAL performed better in terms of convergence to the Pareto optimal front. For the speed reducer problem, in Fig. 4.10b, this result is not as clear as the other ones, but some superiority in terms of the non-dominated solutions can be observed. For the two-bar truss problem (Fig. 4.10c), `gamultiobj` was not able to find any feasible solution. Thus, SCAL performed better. In the disc brake problem, there are no significant differences between the solvers. Figure 4.9c shows that `gamultiobj` is better in terms of distribution while SCAL is better in terms of convergence. According to Table 4.3, `gamultiobj` performed significantly better in terms of hypervolume only for the cantilever and welded beam problems. Despite the good distribution of the non-dominated solutions obtained by SCAL (Fig. 4.10d) in the first part of the curve for the welded beam problem, these solutions are dominated by the ones from `gamultiobj`.



(a) Four-bar plane truss.

(b) Cantilever.

(c) Disc brake.

(d) I-beam.

Fig. 4.9 Non-dominated feasible solutions obtained by the two algorithms for the 30 runs for Four-bar plane truss, Cantilever beam, Disc brake and I-beam problems

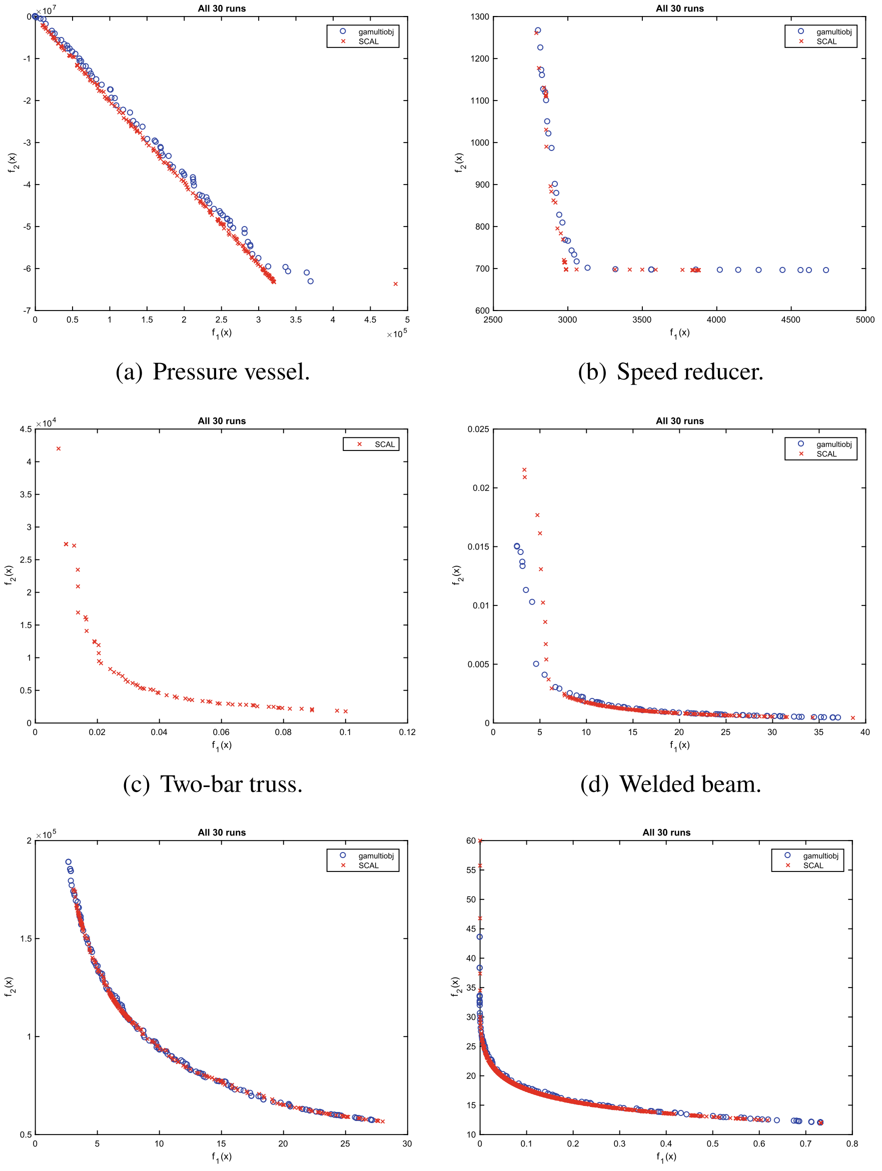
4.6 Conclusions

In this work, ten two-objective engineering design problems were solved by our solver, SCAL and the well-established solver from MATLAB™, `gamultiobj`.

The results obtained show that SCAL is a very promising solver for two-objective optimization problems. This reinforces the results of our previous work [4], now with engineering design problems.

From the ten problems, SCAL performed significantly better in seven problems in terms of average hypervolume. `gamultiobj` performs better in cantilever beam and welded beam problems. For the disc brake problem, there are no significant differences in the hypervolume.

In the near future, we intend to include discrete variables in the solver as well as to improve its performance by using other achievement scalarizing functions.



(a) Pressure vessel.

(b) Speed reducer.

(c) Two-bar truss.

(d) Welded beam.

(e) Spring.

(f) Gear train.

Fig. 4.10 Non-dominated feasible solutions obtained by the two algorithms for the 30 runs for Pressure vessel, Speed reducer, Two-bar truss, Welded beam, Spring and Gear train problems

Acknowledgements This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and the European project MSCA-RISE-2015, NEWEX, with reference 734205.

References

1. Bertsekas DP (1999) Nonlinear programming, 2nd edn. Athena Scientific, Belmont
2. Coello Coello C, Pulido G (2005) Multiobjective structural optimization using a microgenetic algorithm. Struct Multi Optim 30(5):388–403. https://doi.org/10.1007/s00158-005-0527-z
3. Conn AR, Gould NIM, Toint PL (1991) A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. SIAM J Numer Anal 28(2):545–572
4. Costa L, Santo I, Oliveira P (2018) A scalarized augmented lagrangian algorithm (scal) for multi-objective optimization constrained problems. In: ICORES 2018—Proceedings of the 7th international conference on operations research and enterprise systems, pp 335–340
5. Costa L, Santo IAE, Fernandes EM (2012) A hybrid genetic pattern search augmented lagrangian method for constrained global optimization. Appl Math Comput 218(18):9415–9426. https://doi.org/10.1016/j.amc.2012.03.025, http://www.sciencedirect.com/science/article/pii/S0096300312002573
6. Dachert K, Gorski J, Klamroth K (2012) An augmented weighted tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. Comput Oper Res 39(12):2929–2943. https://doi.org/10.1016/j.cor.2012.02.021, http://www.sciencedirect.com/science/article/pii/S0305054812000470
7. Deb K, Sundar J (2006) Reference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, GECCO '06. ACM, New York, NY, USA, pp 635–642. https://doi.org/10.1145/1143997.1144112, http://doi.acm.org/10.1145/1143997.1144112
8. Erfani T, Utyuzhnikov SV, Kolo B (2013) A modified directed search domain algorithm for multiobjective engineering and design optimization. Struct Multi Optim 48(6):1129–1141. https://doi.org/10.1007/s00158-013-0946-1
9. Gong W, Cai Z, Zhu L (2009) An efficient multiobjective differential evolution algorithm for engineering design. Struct Multi Optim 38(2):137–157. https://doi.org/10.1007/s00158-008-0269-9
10. Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. J Assoc Comput 8:212–229
11. Lewis R, Torczon V (1999) Pattern search algorithms for bound constrained minimization. SIAM J Optim 9(4):1082–1099
12. Lewis RM, Torczon V (2002) A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. SIAM J Optim 12(4):1075–1089
13. Mirzakhani Nafchi A, Moradi A (2011) Constrained multi-objective optimization problems in mechanical engineering design using bees algorithm. J Solid Mech 3(4):353–364. http://jsm.iau-arak.ac.ir/article_514445.html
14. Nakayama H (2005) Multi-objective optimization and its engineering applications. In: Branke J, Deb K, Miettinen K, Steuer RE (eds) Practical approaches to multi-objective optimization, no. 04461 in Dagstuhl seminar proceedings. Internationales Begegnungs- und Forschungszentrum f"ur Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany. http://drops.dagstuhl.de/opus/volltexte/2005/234
15. Steuer RE, Choo EU (1983) An interactive weighted tchebycheff procedure for multiple objective programming. Math Program 6(3):326–344. https://doi.org/10.1007/BF02591870

16. Tawhid MA, Savsani V (2018) ϵ-constraint heat transfer search (ϵ-hts) algorithm for solving multi-objective engineering design problems. J Comput Des Eng 5(1):104–119. https://doi.org/10.1016/j.jcde.2017.06.003, http://www.sciencedirect.com/science/article/pii/S228843001730026X
17. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Proceedings of the conference on parallel problem solving from nature, PPSN'98, pp 292–304

Chapter 5
Many-Objective Multidisciplinary Evolutionary Design for Hybrid-Wing-Body-Type Flyback Booster on an Entirely Automated System

Taiki Hatta, Masataka Sawahara, and Kazuhisa Chiba

Abstract The study aims to create pragmatic geometries of flyback booster on reusable launch systems with a high degree of freedom efficiently by evolutionary computations and to present its design candidates based on physics. This article performed a second optimal design that we sophisticated the first trial on many-objective multidisciplinary evolutionary design. The result has revealed that the surface discontinuity of the body back evaded in the hypersonic range could be beneficial for improving the lift-drag ratio in the transonic range. We hypothesized that deliberately dug grooves must be adequate to accomplish flyback boosters generally requires aerodynamic performance in the low-speed range.

Keywords Many-objective multidisciplinary design optimization · Evolutionary computation · Scatter plot matrix · Reusable launch vehicle · Flyback booster

5.1 Introduction

National collaborative research on reusable launch systems (RLSs) is evolving among several Japanese universities. Our university is in charge of a multidisciplinary multi-objective optimal design of the 3D geometry of hybrid-wing-body-type flyback booster by evolutionary computations. In this paper, we focus on two-stage-to-orbit (TSTO) reusable launch vehicles (RLVs). We ultimately constituted a full automated

T. Hatta (✉) · M. Sawahara · K. Chiba
The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
e-mail: h2032085@edu.cc.uec.ac.jp

M. Sawahara
e-mail: s1832049@edu.cc.uec.ac.jp

K. Chiba
e-mail: kazchiba@uec.ac.jp

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_5

multidisciplinary optimization system (FAMOS) for originating the reusable flyback booster. Now we have to describe why we concentrate on these points.

Multi-stage expendable launch vehicles (ELVs) are the primary method for space transportation in the world. In contrast, RLVs: single-stage-to-orbit (SSTO) and TSTO was restudied intensively in recent years to alleviate launch costs further. Generally, to use ELVs always entails considerable costs. ELVs utilize many disposable components to transport payloads, although payloads merely arrive into orbit. Multi-stage launch systems are also necessary because of a limit to the achievable engine performance. To solve the problem, researches on RLVs become active recently. SSTO must be the best way of RLS due to its simple concept. To realize SSTO requires both the lighter body and higher performance engine than up-to-date ELVs. To accomplish single-stage RLVs should enable weight reduction, but there are several technical challenges. RLVs have to thoroughly equip takeoff, atmospheric re-entry, deceleration, and landing functions in a wide speed range. In contrast, TSTO must be a more reasonable approach to fulfill the required conditions. Two advantages exist to adopt TSTO; (1) It can reduce ΔV (changes of velocity) of each stage; (2) TSTO enables each stage to share respective roles.

Against this background, some Japanese universities launched a collaborative research project for developing a space transport system. The testbed called “WInged REusable Sounding rocket (WIRES)” [7] is one of the projects’ fruits, which Kyushu Institute of Technology (KIT) has been developing since 2005. Under the KIT’s initiative, the development of WIRES involves JAXA (Japan Aerospace Exploration Agency), companies, and several domestic and American universities. A blended-wing-body-type vehicle anticipates an ideal form because of its high potential of reusability, operational flexibility, and abort capability. Thus, we reflect the blended-wing-body-type geometry to deliberate reusable flyback booster using FAMOS.

Aerodynamics and structural dynamics are integral elements in aircraft design. On the other hand, it is indispensable to examine not only aerodynamics/structural dynamics but also aerothermodynamics in the RLV design because flyback flight in a wide range of speeds from subsonic to hypersonic is conceivable. Therefore, it is also vital for the design of blended-wing-body-type boosters to recognize the high degree of freedom in expressing geometries, as in aircraft design. Then, we executed the first attempt of multidisciplinary design optimization (MDO) [25]. The first MDO noticed that discontinuous grooves produced on the back surface of geometries due to a glitch in the spline-curve definition. This second MDO trial would correct this matter because a surface discontinuity must provoke an abrupt rise in temperature due to aerodynamic heating in hypersonic conditions. This paper aims to examine how the modification expands the objective-function space and to reveal how aerodynamic heating alleviates.

5.2 Problem Definition

At present, we are promoting research on space transportation systems at several domestic universities including Kyushu Institute of Technology which designs and develops fully reusable space transportation WIRES. Try to design the spacecraft starting from the flight path optimized with WIRES. Originally, although the optimum flight route is also changed according to the change of the geometry, the hurdle for generating the aerodynamic performance matrix necessary for flight route optimization is high, so the optimization of the flight route is a future subject [25].

The flight path used in this study assumes an injection of 10 t payload into the orbit from the Tanegashima Space Center and a circular orbit at an altitude of 350 km. Based on this, defining the aerodynamic performance optimization in three points of transonic, supersonic, and hypersonic speeds. At the defined trans/supersonic design point, the booster is the point where we want to earn a range for fly back to the launch field. At the hypersonic (highest Mach number) design point where the booster and the orbiter are separated, at this point earn the altitude and take a sequence to secure the range margin.

Before we change the topic to our achievement, it is necessary to describe specific problem definitions. The fundamental parts of the problem definitions are kept the same as the previous research [25], and only the geometrical part has been changed this time. We would like to emphasize that we purposely kept the same problem definitions to verify the influence of the geometry modification.

5.2.1 Objective Functions: 6

The objective-functions that we defined are the following [25].

1. Aerodynamics
   The purposes of these objective functions are to expand the options of landing points and landing methods by maximizing the lift to drag (L/D).

   a. Maximizing L/D ($M = 0.65$)
   b. Maximizing L/D ($M = 2.3$)
   c. Maximizing L/D ($M = 6.8$)

2. Structure
   The purposes of this objective function are to increase the weight of payload and to decrease the amount of fuel.

   d. Minimizing empty weight

3. Aerodynamic heating
   These objective functions are to decline the vehicle's surface temperature for diminishing thermal damage to a booster; it is a vital function to develop RLVs. We assume to use carbon fiber reinforced plastics (CFRP) as the material of the

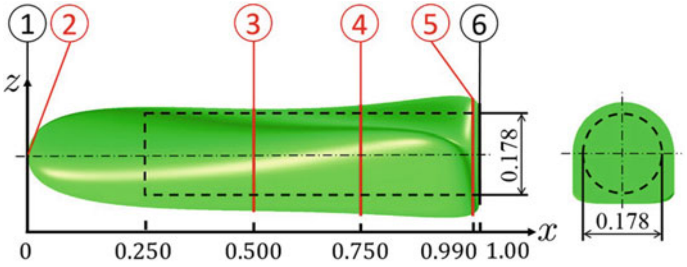


Fig. 5.1 Left: cross-section positions, right: tail surface. The dotted line describes a fuel tank to be a fixed size [25]

booster. The maximum temperature allowed for CFRP is 300 °C. We have to use the material for the thermal protection system (TPS) to protect the booster if the temperature exceeds the limit. The last objective function is to decrease the TPS area on the surface of a booster for reducing its weight and cost.

e. Minimizing surface maximum temperature ($M = 6.8$)
f. Minimizing TPS area on body surface area ($M = 6.8$).

5.2.2 *Design Variables: 40*

Figure 5.1 shows six cross-sectional shapes generated for the x-axis direction. We use section numbers 1 and 6 only to satisfy later-described constraint conditions, and the cross-sectional shape change in evolutionary computations (ECs) is four cross-sections with 2–5. Since each section utilizes ten design variables, the total of design variables is 40.

The first MDO trial brought an issue to make a discontinuous connection at a symmetric plane of bodies because of $\mathrm{d}z/\mathrm{d}y|_{\mathrm{P}_1} \neq 0$ due to available regions of the control points P_3 and P_4 to make wing shapes. It suspects to raise aerodynamic heating around the problem under the hypersonic condition, so influences might not be negligible for evaluations of the objective functions. Thus, we added an operation to adjust $z|_{\mathrm{P}_1}$ to be $\mathrm{d}z/\mathrm{d}y|_{\mathrm{P}_1} = 0$ after arranging all the control points (Table 5.1).

5.2.3 *Constraints: 5*

We define only one constraint on the geometry with five items; provide no constraint on the objective functions. We discard violation individuals at the stage of individual generation by the MOEA Framework; repeat creating individuals for population size

Table 5.1 Definition of design variables [25]

Section number	Design-variable S/N number	Parameter sense	Symbol (refer to Fig. 5.2)	Applicable range [-]	
				lower	upper
②	1, 3, 5, 7, 9	y-direction increment	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}	0	0.073
	2, 4, 6, 8, 10	z-direction increment	V_{z1}, V_{z2}, V_{z3}, V_{z4}, V_{z5}	0	0.087
③	11, 13, 15, 17, 19	y-direction increment	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}	0	0.073
	12,	z-direction increment	V_{z1}	0	0.218
	14, 16, 18,		V_{z2}, V_{z3}, V_{z4}	−0.218	0.218
	20		V_{z5}	−0.218	0
④	21, 23, 25, 27, 29	y-direction increment	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}	0	0.35
	22,	z-direction increment	V_{z1}	0	0.218
	24, 26, 28,		V_{z2}, V_{z3}, V_{z4}	−0.218	0.218
	30		V_{z5}	−0.218	0
⑤	31, 33, 35, 37, 39	y-direction increment	V_{y1}, V_{y2}, V_{y3}, V_{y4}, V_{y5}	0	0.35
	32,	z-direction increment	V_{z1}	0	0.218
	34, 36, 38,		V_{z2}, V_{z3}, V_{z4}	−0.218	0.218
	40		V_{z5}	−0.218	0

until they satisfy the constraint. The following items are specific descriptions of constraints.

1. A body must secure the columnar space for the fuel tank. Figure 5.1 shows the specific lengths.
2. Wavy geometries on a rear surface usually interfere with surface mesh generation. We added the process to avoid this geometrical problem.
3. Each control point connects with a B-spline curve in a single stroke. We added the process to avoid geometry, which has a crossing curve. The crossing curve will become the reason of crushed geometry.
4. We make the tip cross-section geometry closer to a circular shape to keep a smooth surface. We avoid a crushed geometry.
5. The cross-section geometry of section number ⑥ is fixed for the space to equip rocket nozzles.

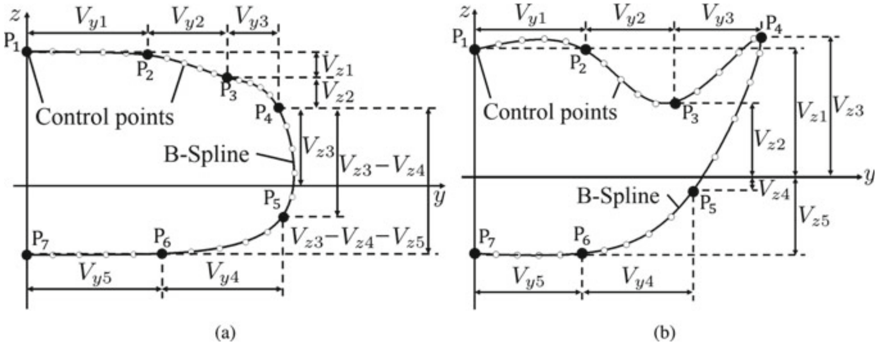


Fig. 5.2 Relationship among design variables V_n and control points P_m at **a** ② and **b** from ③ to ② [25]

5.3 Full Automated Multidisciplinary Optimization System

We adopt Eclipse[1] [1] in the integrated development environment (it is only a development environment; it does not affect the operation of the execution of programs). Then build an optimization system using the MOEA Framework[2] environment. ECs available within the MOEA Framework consist of open-source Java libraries. In the following, we will proceed with the content in line with the problem definition of this research, assuming aerodynamic performance evaluation.

When the optimization system is activated, the MOEA Framework generates population-size individuals. Then, FAMOS processes each individual in parallel after generating the population. The contents to be processed in parallel are (1) pre-processing for evaluating objective functions, (2) evaluating objective functions, and (3) post-processing for computing objective functions. FAMOS generates folders `/G#_I`i`/` with generation numbers # and personal identification numbers i; evaluates each objective in standalone in it.

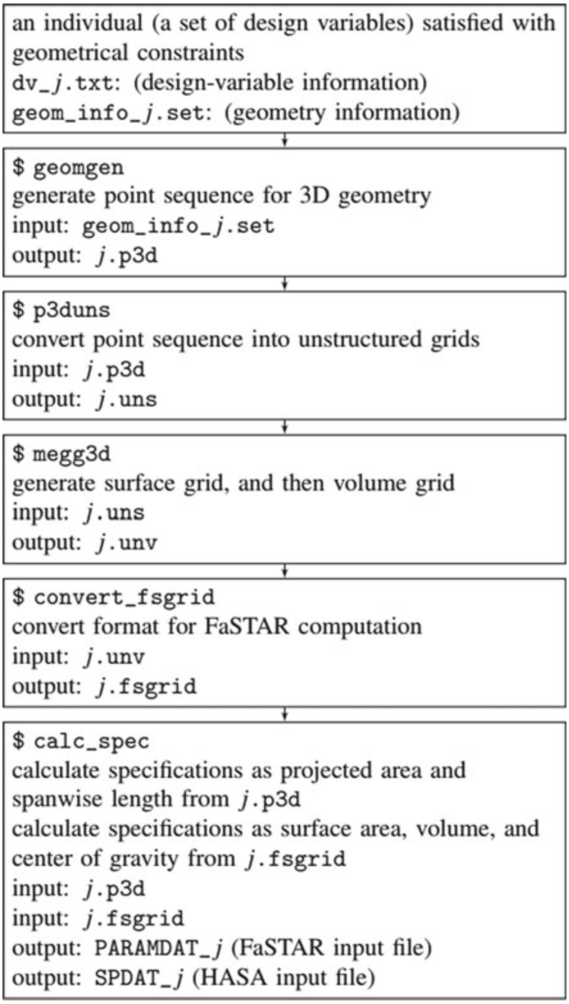
What is vital for running the system is a computing environment that performs a CFD analysis. We construct an integrated development environment on the terminal at hand, but the CFD analysis throws the job to appropriate computers. Currently, it is possible to use various information infrastructure systems. However, as there was a hurdle in uploading mesh and downloading the result, we have an issue regarding the security of a communication gateway. Hence we decided to close the system in the laboratory by occupying the system (Intel Xeon E5-26xx series: 9 nodes 156 cores, the parallel number is twice the number of cores) and implement CFD analyses.

[1] "Eclipse Foundation" available online at https://www.eclipse.org/ [retrieved 5 Dec. 2018].

[2] "MOEA Framework" available online at http://moeaframework.org/ [retrieved 5 Dec. 2018].

5.3.1 Pre-process

For an aerodynamic analysis that accurately captures phenomena and requires precision, it is necessary to define the geometry precisely and to discretize the surface/space of airframes (surface/volume mesh generation). This pre-processing is the most obstacle part of automation. The pre-process flowchart shown in Fig. 5.3 assembles the following description.



an individual (a set of design variables) satisfied with geometrical constraints
dv_j.txt: (design-variable information)
geom_info_j.set: (geometry information)

$ geomgen
generate point sequence for 3D geometry
input: geom_info_j.set
output: j.p3d

$ p3duns
convert point sequence into unstructured grids
input: j.p3d
output: j.uns

$ megg3d
generate surface grid, and then volume grid
input: j.uns
output: j.unv

$ convert_fsgrid
convert format for FaSTAR computation
input: j.unv
output: j.fsgrid

$ calc_spec
calculate specifications as projected area and spanwise length from j.p3d
calculate specifications as surface area, volume, and center of gravity from j.fsgrid
input: j.p3d
input: j.fsgrid
output: PARAMDAT_j (FaSTAR input file)
output: SPDAT_j (HASA input file)

Fig. 5.3 Flowchart of pre-processing for computing objective-function evaluation. j denotes the serial number of individuals in evolutionary computation, which uses generation number # and the individual number i in a generation [25]

5.3.1.1 After Generating Population by MOEA Framework

The design-variable information is stocked to dv_*j*.txt. Based on this, FAMOS prepares an input file geom_info_*j*.set for generating the connection data of body surface points.

5.3.1.2 Generating Point Sequence of Geometry Surface

We generate a point sequence structurally arranged in the direction of x, y, and z with the name of *j*.p3d to express the outline of the body surface. FAMOS forms point sequence data generated by computer-aided design (CAD) data.

5.3.1.3 Unstructured Point Sequence to Structurally Arrange

The system converts the format of *j*.p3d to a format for unstructured surface mesh and stock it as *j*.uns.

5.3.1.4 Discretizing Surface and Volume by Unstructured Mesh Method

FAMOS generates a ridge point sequence at the surface region (zone) boundary described in *j*.uns; generates an unstructured surface mesh together with the *j*.uns information (in other words, it prepares the part where we would generate ridgelines to be the boundary of the zones). This way provides the symmetry plane and the outer boundary; the computational space closes. Then, the system generates an unstructured volume mesh using this surface mesh. FAMOS makes prism layers on the wall surface (the thickness of the 1st layer of $y^+ \sim 1$ and at least 41 layers are laminated) to resolve the boundary layer and outputs as *j*.unv. Finally, *j*.fsmesh outputs according to the format of the solver used this time.

5.3.1.5 Generating Body Specification

FAMOS created files of geometry specification data (projected area, span length, body surface area, body volume, and center of gravity position) for the post-process.

5.4 Method of Numerical Functions

5.4.1 Optimizer

Since one of the information desired by the multiobjective design optimization is the executable structure of objective-function space, FAMOS utilizes ECs for the optimization method to perform a global search. Many ECs become prominent with steady progress; this study adopts SPEA2 [5] so that we compare the results with those of the first trial on an equal footing. Moreover, we respectively chose simulated binary crossover [2] and polynomial mutation [3] for crossover and mutation.

If we can acquire various solutions in a real-world problem, the diversity of design candidates must expand, and the range of design information procured by data mining should widen. The prior study [25] indicated that SPEA2, which does not stipulate search directions, can obtain more distinct individuals than IBEA [6], which prescribes search directions by hypervolume [4].

The results suppose that to search champions of each objective function should expedite convergence for a large-scale optimum design problem to be executed only with small population size and a small number of generations; a two-step search algorithm should be efficient, which accommodates regions between champions and gains diversity in nondominated solutions.

5.4.2 Data Mining

This research employed the scatter plot matrix (SPM) [26] for data mining; SPM is valid to compare the distribution of solutions and the correlation coefficient between each objective-functions space as a bird's eye view. SPM declares tradeoffs between the objective functions. We set the range of values based on the maximum and minimum value obtained by the optimization results.

5.4.3 Evaluation Methods of Objective Functions

1. Aerodynamics

   a. Mesh generations: We would assume blended-wing-body type geometries with high degrees of freedom. We applied the hybrid unstructured mesh automatic generation software: Mixed-Element mesh Generator in 3D (MEGG3D) [11–21] for the mesh generation. It laminated prismatic layers on the body surface within 99% of the boundary layer thickness; at least 41 prism layers constitute.

Table 5.2 Usage range and density of TPS material [25]

TPS material	Range (°C)	Value (kg/m^2)
LI-900	$300 \leq T < 1260$	144
RCC	$1260 \leq$	1986

b. Computational methods: We utilized a cell-vertex finite volume method: FaST Aerodynamic Routines (FaSTAR) [9, 10] developed by Japan Aerospace Exploration Agency (JAXA) for assessing aerodynamic performances.
   i. The governing equation: Compressible Reynolds-averaged Navier-Stokes equation
   ii. A turbulence model: shear stress transport (SST) -2003sust [22]
   iii. The numerical flux computations method: Harten-Lax-van Leer-Einfeldt-Wada (HLLEW) method [23]
   iv. The time integration method: Lower/Upper Symmetric Gauss-Seidel (LU-SGS) implicit method [24].

2. Structure: We applied the hypersonic aerospace sizing analysis (HASA) [8] developed by the National Aeronautics and Space Administration (NASA) for weight estimation. Since the original HASA assesses a wing and fuselage separately, we modified it for the blended wing body.
3. Aerodynamic heating: FaSTAR also evaluates the aerodynamic heating. The state equation analyzed the node temperature on the surface. This study alters the materials of TPS according to the temperature that the booster reaches. Table 5.2 presents which materials we affix for each temperature range.

5.5 The Modifications of Problem Definitions in This Paper

5.5.1 The Geometrical Problems in the Previous Research

The irregular body surface emerged on the surface in the previous results, as shown in Fig. 5.4. Also, the wavy geometry seldom generated between the cross-sections 5 and 6. We suspect that these geometrical traits induce sharp surges in temperature on the surface of the boosters under the hypersonic condition. The correction of the geometrical definition additionally anticipates expanding the objective-function space.

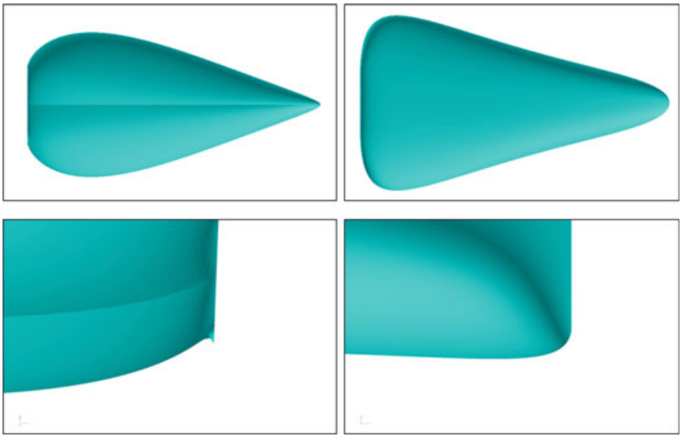


Fig. 5.4 The modificated geometry on the surface between left and right-hand side (images on upper half) and the modificated dent between the cross section ⑤ and ⑥ (images on lower half)

5.5.2 *Modification Manner of the Geometrical Subjects*

We disposed of seven control points for one cross-section of the booster body on the x-z plane; performed the B-spline interpolation to these points. However, the connecting points of B-spline curves on the x-z plane were still intermittent. Therefore, we added the process to move the coordinate points on the connecting point of the B-spline and to make continuous surfaces on the x-z plane.

For the specific description of the process, we have to explain a cubic function to represent the B-spline curve. A coefficient of the cubic function determines the gradient of the B-spline curve, so we solve the inverse problem to search for coefficients that make the gradient of B-spline to zero on the discontinuous surface at the two control points as P_1 and P_7 shown in Fig. 5.2 for each cross-section.

Additionally, we have to revise boosters with wavy geometry between cross-sections 5 and 6. We supplemented a new condition into the fifth constraint to prevent undulating surfaces near the cross-section 6. Figure 5.4 exhibits consequence examples of these alterations.

5.5.3 *SBX Modification*

We resolve SBX issues when changing discontinuous surfaces. General SBX uses two parents; creates two children. The previous study stipulated that both children have to fulfill the geometrical constraints to add children in SBX. The geometrical modifications induced it tough to satisfy this rule, so we alleviated it as follows.

New rule permitted a child who fulfilled the geometrical constraints to retain as a candidate of next-generation individual, even when another child does not satisfy the constraints.

5.6 Results and Discussion

We run the FAMOS until 30 generations using a population size of 10 individuals; it took roughly one month. Figure 5.5 compares the previous and present distributions of all the solutions in the objective-function space on SPMs. The following subjects are noticeable results.

1. The present geometrical modifications restrained extremely high temperatures on the body surface (over 5000 K) drastically.
2. The value of L/D declined in all speed ranges, and the objective-functions space expanded in a negative direction.

5.6.1 Cause of the Extreme High Temperature Area

First, we need to verify the reason for the drop in temperature. We hypothesized that discontinuous surfaces and wavy geometries provoked temperature boosts. We visualized the surface temperature of a broken area in Fig. 5.6a; the figure indicated that the irregular surface on the body back did not cause an extremely high temperature.

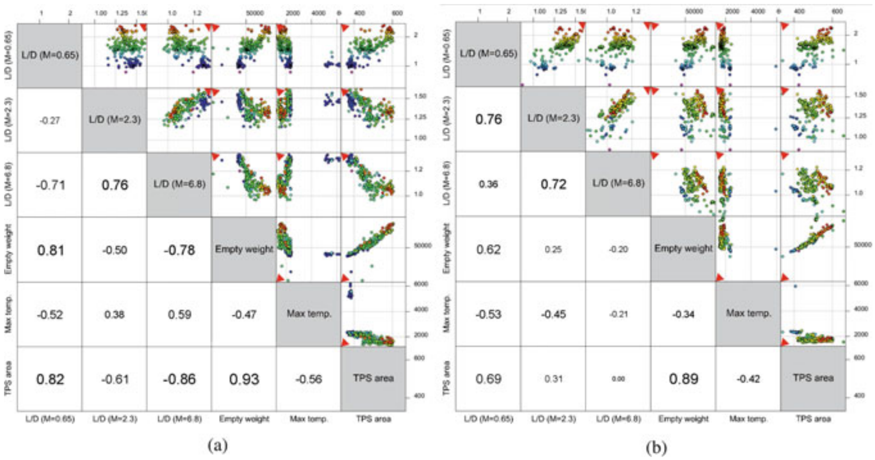


Fig. 5.5 SPMs of all individuals in the six objective-functions space; **a** the previous results [25]; **b** the present results. The red triangles denote optimum directions. We colored the plots with values of the objective function1

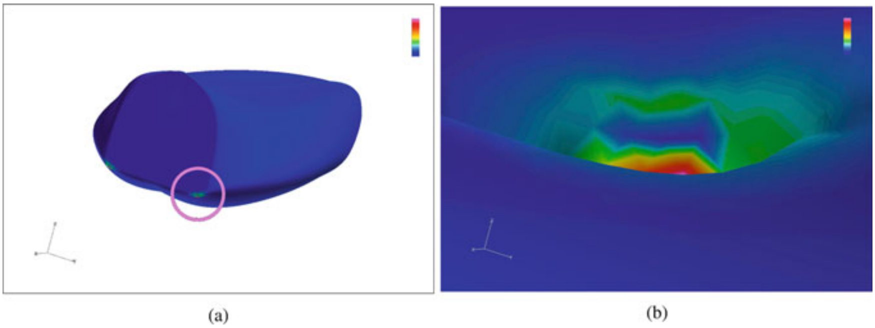


Fig. 5.6 The surface temperature distribution of the geometry with the highest body surface temperature (roughly 6200 K) obtained by the first MDO; **a** overall view; **b** enlarged view of the area surrounded by the circle in (**a**)

In contrast, the wavy dent between the cross-section demonstrates the reason for extremely high temperatures on the surface of the booster. Figure 5.6b indicates the high temperatures on the edge of the cross-section 6. The pink region in Fig. 5.6b is the most high-temperature area. The individuals with ultimate heats have significantly diminished by the extra constraint.

5.6.2 *The Temperature Restraint on Boosters' Surface*

Apart from boosters with an extremely high temperature (over 5000 K) area, we would focus on the causes of high temperatures (over 2000 K) on the surface. Figures 5.7a, b show the four boosters from above, which have the highest temperature in the range of 2000–2500 K in the previous and present results; there is no significant variation in temperature between the discontinuous and continuous surface. On the other hand, the area that signifies the temperature around 2000 K is all on boosters' nose. Figure 5.7 indicates the principal factor of high temperature (over 2000 K) on the body surface must result from the intense heat on boosters' nose.

5.6.3 *Negative Expansion of Objective-Functions Space*

Figure 5.5 implies L/D declines in all speed range after the optimization; the results deliver us a hypothesis regarding the discontinuous surface in the prior research. We presume that the irregular facade prompted the growth of L/D, especially reducing the drag in previous research. The present research removed the surface discontinuity

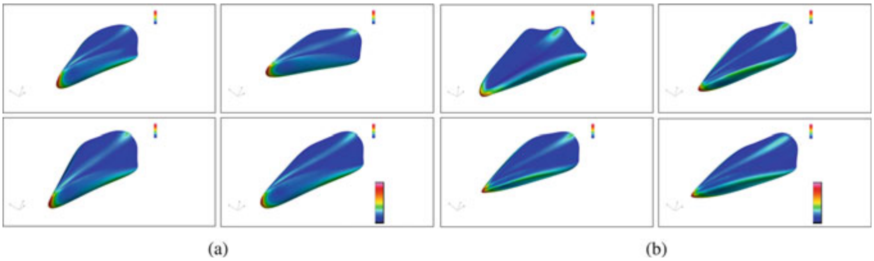


Fig. 5.7 The example of two individuals whose body surface temperature are around 2000 K; **a** the result from the previous research; **b** the result from the present study

from all individuals. We suspect that this modification makes the L/D decrease and causes the objective-function space to expand in a negative direction. To examine the evidence of the hypothesis is one of the future assignments.

5.7 Conclusions

This study designed a two-stage-to-orbit booster stage as part of a reusable space transport project in Japan. We implemented a second multidisciplinary many-objective optimal design using a fully automatic multidisciplinary optimization system. We examined the effects of correcting two problems regarding shaping the definition that emerged in the previous first optimal design, i.e., discontinuity at the back of bodies and waving at the rear of bodies.

As a result, the discontinuity at the back of bodies did not affect the surface temperature, while the wavy shape of the tail eliminated the sharp rise in the surface temperature. Moreover, to correct the discontinuity at the body tail caused a reduction in the lift-drag ratio over the entire speed range and expanded the objective-function space in the negative direction. Hypersonic range evades surface discontinuity due to a dramatic temperature growth, but the consequence has suggested that it could contribute to raising the lift-drag ratio at low speeds.

Acknowledgments This research was conducted under the support of JHPCN (jh180057). A part of the work was implemented under the Collaborative Research Project of the Institute of Fluid Science, Tohoku University (J19I076). The images for CFD results were created by using FieldView18.0 as provided by Intelligent Light via its University Partners Program.

References

1. Chiba K, Sumimoto T, Sawahara M (2019) Completely automated system for evolutionary design optimization with unstructured computational fluid dynamics. In: Proceedings of international conference on intelligent systems. Metaheuristics and Swarm Intelligence. ACM (2019)
2. Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. Complex Syst 9:115–148
3. Deb K, Agrawal RB (1996) A Combined Genetic Adaptive Search (GeneAS) for engineering design. Comput Sci Inf 26:30–45
4. Eckart Z, Lothar T (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271
5. Eckart Z, Marco L, Lothar T (2001) SPEA2: improving the performance of the strength pareto evolutionary algorithm. In: Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich
6. Eckart Z, Simon K (2004) Indicater-based selection in multiobjective search. Parallel Problem Solving Nat—PPSN VIII 3242:832–842
7. Fujikawa T et al (2017) Research and development of winged reusable rocket: current status of experimental vehicles and future plans. APISAT2017 12(I5-3):1826–1837
8. Harloff GJ, Berkowitz BM (1988) HASA—Hypersonic Aerospace Sizing Analysis for the Preliminary Design of Aerospace Vehicles. NASA CR-182226
9. Hashimoto A, Murakami K, Aoyama T, Hishida M, Sakashita M, Lahur P (2015) Development of Fast Unstructured-Grid Flow Solver FaSTAR. J Jpn Soc Aeronaut Space Sci 63(3):96–105
10. Hashimoto A, Murakami K, Aoyama T, Ishiko K (2012) Toward the Fastest Unstructured CFD Code 'FaSTAR'. In: AIAA Paper 2012–1075. 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition
11. Ito Y, Murayama M, Yamamoto K, Shih AM, Soni BK (2009) Efficient computational fluid dynamics evaluation of small device locations with automatic local remeshing. AIAA J 47(5):1270–1276. https://doi.org/10.2514/1.40875
12. Ito Y, Murayama M, Yamamoto K, Shih AM, Soni BK (2011) Efficient Hybrid Surface and Volume Mesh Generation for Viscous Flow Simulations. In: AIAA Paper 2011–3539, 20th AIAA computational fluid dynamics conference, pp 490–496
13. Ito Y, Nakahashi K (2002) Direct surface triangulation using stereolithography data. AIAA J 40(3):490–496
14. Ito Y, Nakahashi K (2002) Surface triangulation for polygonal models based on CAD data. Int J Numer Methods Fluids 39(1):75–96. https://doi.org/10.1002/fld.281
15. Ito Y, Nakahashi K (2004) Improvements in the reliability and quality of unstructured hybrid mesh generation. Int J Numer Methods Fluids 45(1):79–108. https://doi.org/10.1002/fld.669
16. Ito Y, Nakahashi K (2004) Reliable isotropic tetrahedral mesh generation based on an advancing front method. In: Proceedings of the 13th international meshing roundtable, pp 95–105
17. Ito Y, Shih AM, Koomullil RP, Kasmai N, Jankun-Kelly M, Thompson D (2009) Solution adaptive mesh generation using feature-aligned embedded surface meshes. AIAA J 47(822):1879–1888. https://doi.org/10.2514/1.39378
18. Ito Y, Shih AM, Soni BK (2009) Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. Int J Numer Methods Eng 77(13):1809–1833. https://doi.org/10.1002/nme.2470
19. Ito Y, Shih AM, Soni BK (2011) Hybrid mesh generation with embedded surfaces using a multiple marching direction approach. Int J Numer Methods Fluids 67(1):1–7. https://doi.org/10.1002/fld.1962
20. Ito Y, Shih AM, Soni BK (2011) Three dimensional automatic local remeshing for two or more hybrid meshes. Int J Numer Methods Fluids 66(12):1495–1505. https://doi.org/10.1002/fld.2324
21. Ito Y, Shih AM, Soni BK, Nakahashi K (2007) Multiple marching direction approach to generate high quality hybrid meshes. AIAA J 45(1):162–167. https://doi.org/10.2514/1.23260

22. Menter FR (1994) Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J 1598–1605
23. Obayashi S, Guruswamy GP (1994) Convergence acceleration of an aeroelastic Navier-Stokes solver. AIAA J 33(6):1134–1141
24. Sharov D, Nakahashi K (1998) Reordering of hybrid unstructured grids for lower-upper symmetric gauss-seidel computations. AIAA J 36:484–486
25. Sumimoto T, Chiba K, Kanazaki M, Fujikawa T, Yonemoto K, Hamada N (2019) Evolutionary multidisciplinary, design optimization of blended-wing-body-type flyback booster. in: Proceeding on the 57th AIAA Aerospace Science Meeting, AIAA paper-2019-0703, AIAA, San Diego, California, USA
26. Tatsukawa T, Oyama A, Kohira T, Kemmotsu H, Miyachi H (2017) iSPM—an interactive scatterplot matrix for visualizing multidimensional engineering data. In: Proceedings of the IEEE visualization conference. IEEE, Phoenix, Arizona, USA

Chapter 6
A Neuroevolutionary Approach to Feature Selection Using Multiobjective Evolutionary Algorithms

René S. Pinto, M. Fernanda P. Costa, Lino A. Costa, and António Gaspar-Cunha

Abstract Feature selection plays a central role in predictive analysis where datasets have hundreds or thousands of variables available. It can also reduce the overall training time and the computational costs of the classifiers used. However, feature selection methods can be computationally intensive or dependent of human expertise to analyze data. This study proposes a neuroevolutionary approach which uses multi-objective evolutionary algorithms to optimize neural network parameters in order to find the best network able to identify the most important variables of analyzed data. Classification is done through a Support Vector Machine (SVM) classifier where specific parameters are also optimized. The method is applied to datasets with different number of features and classes.

Keywords Neuroevolutionary · Multi-objective optimization · Feature selection

6.1 Introduction

In predictive analysis, feature selection is the process of identifying the most important, preferably a few, variables or parameters which are relevant in predicting the outcome. Other motivations can exist, such as: feature set reduction, to reduce

R. S. Pinto (✉) · A. Gaspar-Cunha
Institute of Polymers and Composites, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: b8057@dep.uminho.pt

A. Gaspar-Cunha
e-mail: agc@dep.uminho.pt

M. F. P. Costa
Centre of Mathematics, University of Minho, Campus Gualtar, 4710-057 Braga, Portugal
e-mail: mfc@math.uminho.pt

L. A. Costa
ALGORITMI Center, University of Minho, Campus Gualtar, 4710-057 Braga, Portugal
e-mail: lac@dps.uminho.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_6

resource utilization on future data collections; general data reduction, to increase algorithm speed; or performance improvement, to increase predictive accuracy [1]. For a n-dimensional dataset there exist 2^n possible feature subsets, becoming impractical to evaluate all possible solutions for a large n, leading to an NP-Hard combinatorial problem [2].

Several studies have been proposed to tackle feature selection problems. Simultaneously, there is research work using multiobjective evolutionary algorithms (MOEA) applied to different data classifiers. However, according to [3] most of the approaches for feature selection concerning optimization techniques are based on a single objective. There are a few studies which use multiobjective optimization for feature selection problems.

In [4], the authors proposed a framework for SVM based on multiobjective optimization to minimize the risk of the classifier. The same approach is presented in [5] with the aim of minimizing the number of features of the model. In [6], the authors used hierarchical MOEA to perform feature selection by generating a set of classifiers and selecting the best set of them. In [7], a MOEA optimization methodology is proposed to deal with feature selection problems using a SVM classifier. The proposed approach is applied and validated in a problem of cardiac Single Proton Emission Computed Tomography (SPECT).

In [8–10] authors apply successfully neuroevolutionary approaches in different kinds of problems concerning multiobjective optimization.

The present study suggests a neuroevolutionary approach to deal with feature selection problems. In order to reduce complexity of the optimization, artificial neural networks (ANNs) are used to map the most relevant features of analyzed data. MOEA is applied to optimize and find the best classifier parameters and ANNs which gives the most relevant features. The methodology is applied in datasets with different numbers of features, samples and classes. To compare the results, a binary approach, i.e., without using ANNs, is also applied.

6.2 Methodology

Regarding feature selection problems, that usually leads with thousands of features, the binary representation can increase drastically the computational costs necessary to evaluation because the search space increases with the number of features, since each feature is represented as one single bit in the chromosome of genetic algorithm. Usually, bit 0 means that the feature should not be considered by the classifier and bit 1 means the opposite, i.e., feature should be considered in the classification process. Therefore, this study proposes an alternative codification scheme, based on ANNs. Each chromosome encodes the weights and biases of an ANN instead of considering all the binary features for classification. The ANN is structured in three layers, where the Input Layer receives the number of a single feature and the output is the probability of the input feature being considered by the classifier. The number of inputs is the number of bits necessary to encode the number of features. For instance, if a dataset is composed by samples with 2000 features, 11 bits are required. On the other hand, the same example using binary representation it will requires a chromosome of at least 2000 genes to encode each feature. Although this study use a fixed topology for the ANNs (with 20 neurons in the hidden layer), different topologies can be used by the MOEA. Figure 6.1 illustrates the ANN considering the topology for the given example. The chromosome (without classifier parameters) will need only 272 genes to encode all ANN parameters instead of 2000 genes necessary by the binary chromosome. Figures 6.2 and 6.3 illustrate the structure of chromosome for binary and neuroevolutionary approaches, respectively.

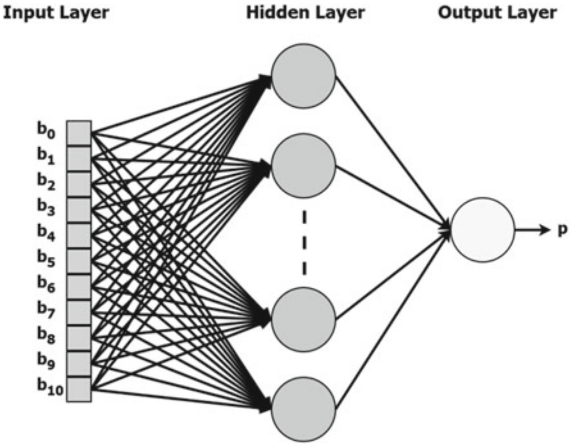


Fig. 6.1 Neural Network partially represented. Input layer receives a feature number in binary form (bits b_0, b_1… b_{10}). Hidden layer has a total of 20 neurons (only four are show on the figure). Output layer is composed by one single neuron that gives output p, which is the probability of input feature be relevant (selected) to the classifier

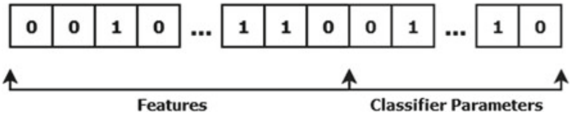


Fig. 6.2 Example of a chromosome for binary representation. The use information of each feature is encoded in one single bit, parameters for the classifier should be encoded at the end of the chromosome using binary representation

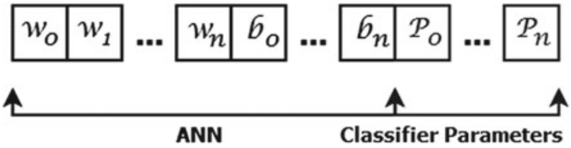


Fig. 6.3 Chromosome representation for neuroevolutionary approach. Each gene encodes a real number which might represent a weight or bias (of the ANN) or a parameter for the classifier

6.2.1 Classifier

It is important to point out that any classifier can be used with the proposed methodology. However, in this study a Support Vector Machine classifier was considered for the experiments.

Support Vector Machines (SVMs) are a set of models with associated learning algorithms that can be applied to classification and regression. The samples in a dataset are represented as points in space, so points of different categories can be separated by a hyper-plane or a set of hyper-planes. Although SVMs are binary linear classifiers, additional methods, such as kernel methods, can be applied to perform non-linear classifications. SVMs classifiers had been successfully applied in many machine learning problems.

The SVM classifier performance heavily depends on the selection of the right parameters, such as kernel function, kernel coefficients and regularization. In this study, a SVM non-linear classifier with Radial Basis Function (RBF) was considered with two different parameters to be optimized: the regularization (C) and the kernel gamma parameter (γ). This type of classifier was already used by [7] in feature selection problems with multiobjective optimization.

6.2.2 Performance Measure for Classification

A systematic analysis of performance measurements for classification can be found in [11]. When dealing with binary classification, *i.e.*, when datasets are composed by samples of two distinct (non-overlapping) classes, the precision metric of the classifier can be expressed by equation:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

where TP is the number of true positives, *i.e.*, the number of samples correctly classified and FP is the number of false positives, *i.e.*, the number of samples that belongs to a given class, but were incorrectly assigned to the other class.

For multi-class datasets the precision P can be expressed by the equation:

$$P = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$$

where tp_i is the number of true positives for a given class, fp_i is the number of false positives, *i.e.*, the number of samples of the given class that were incorrectly classified in another class, and l is the total number of possible classes.

6.2.3 Multiobjective Optimization

In feature selection problems there are two main conflicting objectives: the minimization of the number of features used for classification and the maximization of classifier precision. Thus, multiple solutions with different tradeoffs (number of features versus precision) can emerge from multiobjective optimization approaches.

The methodology proposed in this study combines the reduction of the search space (by using ANNs) with the minimization of objectives (number of features and classification error) into a single approach by using Neuroevolutionary MOEA (Multiobjective Optimization Evolutionary Algorithm). Figure 6.4 illustrates the overall algorithm.

The algorithm comprises a multiobjective optimization evolutionary process. It starts by an initial population of solutions which can be randomly generated. The ANNs are used in the evaluation phase to provide the features and parameters to be used by the classifier. The classifier is applied to the dataset considering the provided parameters and objective functions values are calculated from classification results. The process continues by sorting the solutions following a fitness criterion and deciding if convergence is reached or more iterations are needed. Evolution is promoted by selection and variation procedures.

At the end, a Pareto front composed by a set of non-dominated solutions which give different tradeoffs between the number of features used for classification and the precision of the classifier is expected. In this context, two objective functions can be defined:

f_1 = Number of features used for classification.

f_2= Classifier error defined as $\mathrm{f}_2 = 1 - \mathrm{P}$, where P is the classifier precision expressed between [0.0, 1.0].

By defining f_2 as the classifier error, the optimization problem becomes minimize (at the same time) f_1 and f_2.

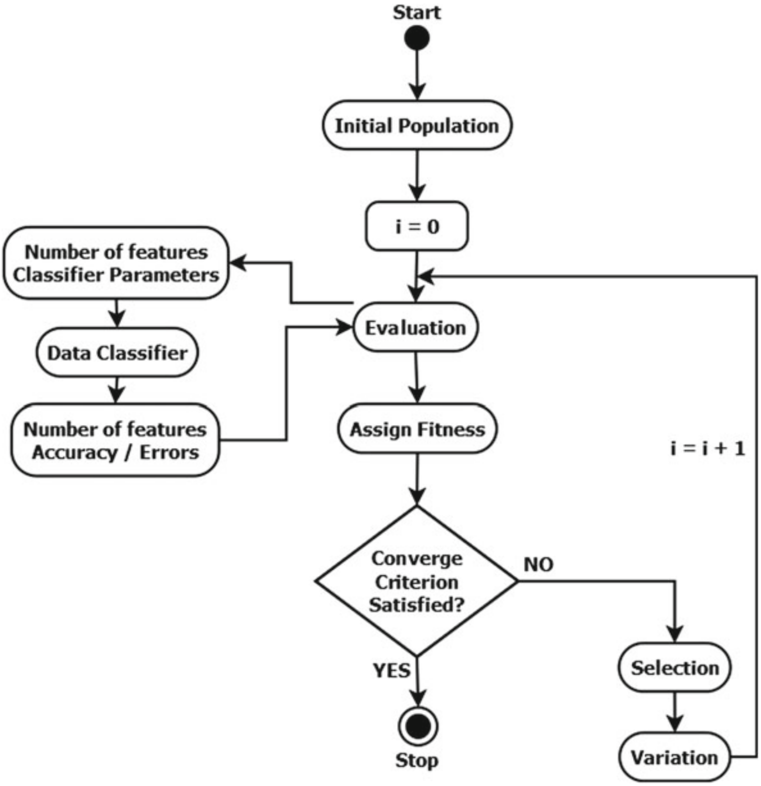


Fig. 6.4 Algorithm for the proposed approach for feature selection using neuroevolutionary and multiobjective optimization evolutionary methods

6.3 Experimental Design

To evaluate the proposed approach, eight datasets were chosen from UCI Machine Learning Repository[1] and one well known dataset (*colon*) was chosen from the literature in feature selection. All datasets comprise different number of features, samples and classes. Thus, a multiclass SVM classifier implementation was used in the experiments. Table 6.1 lists all datasets.

The proposed approach was implemented in MATLAB using the models and functions provided by the Statistics and Machine Learning Toolbox to perform SVM multiclass classification. The multiobjective optimization algorithm was implemented based on the SMS-EMOA algorithm [12]. In each generation, one single offspring is produced. The selection is done using a uniform distribution and variation is performed by the SBX-Crossover operator, which is designed to work with real number representations. Since the parameters of the classifier and of the

[1] Available at https://archive.ics.uci.edu.

Table 6.1 Datasets used in the experiments

Dataset	Features	Samples	Classes
Colon	2000	64	2
Ionosphere	34	351	2
Musk-1	166	476	2
Sonar	60	208	2
Semeion	256	1593	10
Yeast	8	1484	10
Libras	90	360	15
Wine1	12	1600	10
Solar	12	1066	7

neural networks are real numbers, this operator is adequate for the neuroevolutionary approach. The fitness of each solution and replacement strategy are based on Pareto front and *hypervolume* measure [13].

To compare the results, a binary approach was also applied to the datasets. The overall algorithm is the same, except by the evaluation and variation phases, where each solution is represented by a binary chromosome (Fig. 6.2) and a two point crossover operator is used instead of the SBX-Crossover.

Concerning the classifier parameters C (regularization) and γ (kernel gamma), after preliminary experiments with all datasets and based on former studies found in the literature, the following intervals were defined: [1, 500] for C and [0.01, 10] for kernel gamma, respectively. To encode these values in the binary representation, 10 bits were used for each parameter. This leads to 2^{10} possible integer values that are normalized into the respective parameter interval.

All classifications were performed using k-fold cross- validation with $k = 10$. The partitions for each dataset were pre-defined and used for both binary and neuroevolutionary approaches. The size of each population was set to 150 individuals (solutions) and the number of maximum generations was set to 300 due to computational time constraints.

6.4 Results and Discussion

Figures 6.5 and 6.6 show the evolution of the *hypervolume* for each generation for binary and neuroevolutionary approaches, respectively. All values were normalized concerning the origin and the maximum allowed point for all datasets. All curves are visually similar in both cases, but it can be seen that most of the curves in Fig. 6.6 (neuroevolutionary) converges slightly faster than Fig. 6.5.

Table 6.2 lists the *hypervolume* of Pareto front of final populations for both representations. Better results are highlighted. The neuroevolutionary approach presented

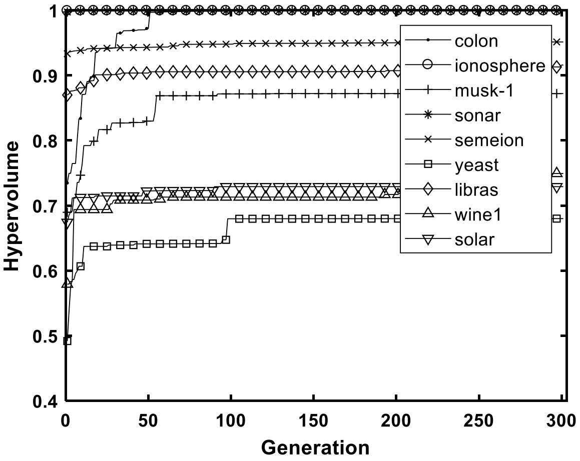


Fig. 6.5 *Hypervolume* evolution for each dataset using binary representation

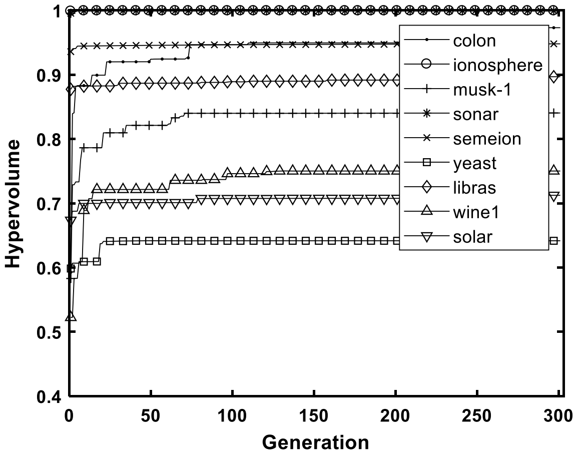


Fig. 6.6 *Hypervolume* evolution for each dataset using neuroevolutionary approach

Table 6.2 *Hypervolume* for Pareto front of final populations for binary and neuroevolutionary approaches

Dataset	Hypervolume	
	Binary	Neuroevolutionary
Colon	0.85	0.86
Ionosphere	0.99	0.99
Musk-1	0.77	0.78
Sonar	0.78	0.99
Semeion	0.05	0.08
Yeast	0.19	0.19
Libras	0.22	0.26
Wine1	0.50	0.46
Solar	0.46	0.46

better results for 5 of the 9 datasets, 3 datasets presented equal results and only one dataset (*wine1*) presented higher *hypervolume* for binary approach.

To illustrate the results of each optimization, Figs. 6.7 and 6.8 show the initial and final populations for datasets *semion* and *colon* (neuroevolutionary), respectively. Other datasets were omitted due to space constraints. It can be seen clearly the evolution of initial population to a set of optimal solutions which gives different tradeoffs between the number of features (f_1) and the classifier error (f_2).

For all datasets, an optimal solution (located in the knee of the Pareto curve) was selected from final population. Table 6.3 lists these solutions along with its classifier parameters, precision and number of features (better precision results are highlighted). In terms of classifier precision, for five of nine datasets, the neuroevolutionary approach presented better results. For the dataset *sonar*, neuroevolutionary reached 100% of precision using only one feature to classification against the binary

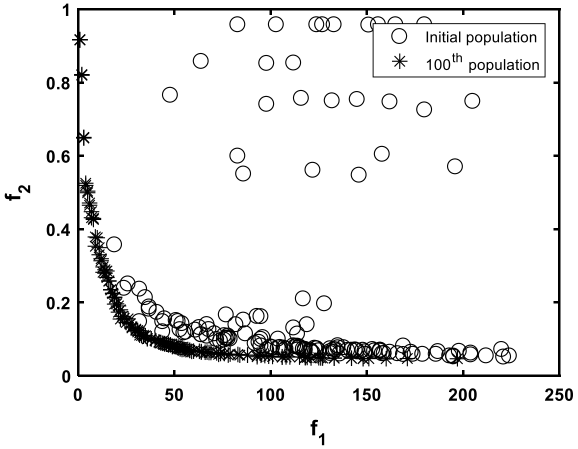


Fig. 6.7 Initial and final populations for dataset *semeion* (neuroevolutionary approach)

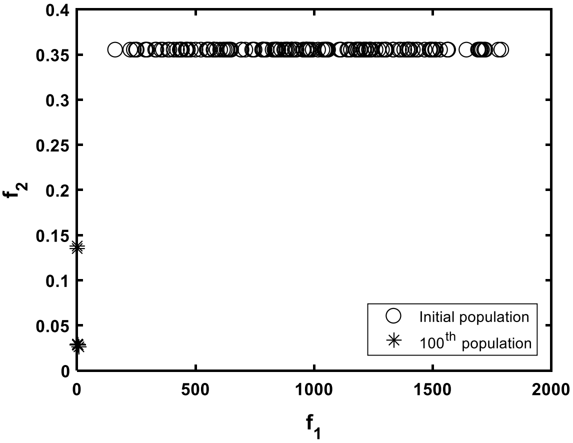


Fig. 6.8 Initial and final populations for dataset *colon* (neuroevolutionary approach)

Table 6.3 Optimal solutions selected from Pareto front of final population for each dataset (classifier parameters, precision and number of features are listed)

Dataset	Binary			Neuroevolutionary		
	C, γ	P	f_1	C, γ	P	f_1
Colon	324.08, 8.33	0.97	2	45.07, 8.24	0.98	2
Ionosphere	17.08, 0.47	1.00	1	354.72, 9.99	1.00	1
Musk-1	32.19, 9.71	0.82	2	124.01, 18.57	0.84	2
Sonar	90.67, 0.63	0.83	2	72.99, 3.00	1.00	1
Semeion	258.30, 1.58	0.83	17	474.86, 0.89	0.85	22
Yeast	1.00, 0.16	0.59	5	475.85, 1.89	0.58	5
Libras	1.00, 0.33	0.87	7	288.02, 0.20	0.85	6
Wine1	23.90, 0.01	0.72	3	218.43, 0.02	0.75	4
Solar	21.47, 2.52	0.71	3	126.47,3.31	0.71	3

approach, which found 2 features with 83% of precision. For datasets *semeion* and *wine1*, the neuroevolutionary approach presented better classifier precision, but the number of features was higher than the binary approach. The results for dataset *semeion* were 85% of precision (neuro) against 83% (binary) and the number of features were 22 (neuro) against 17 (binary). For dataset *wine1*, the results were 75% of precision (neuro) versus 73% (binary) and 4 features (neuro) versus 3 features (binary).

Concerning the dataset *libras*, the neuroevolutionary approach reached 85% of precision against 87% for binary approach, but only 6 features were used (against 7 features for binary). Datasets *ionosphere* and *solar* presented exactly the same results (precision and number of features) for both approaches. Only the dataset *yeast* presented better results for the binary approach: 59% of precision against 58% for neuroevolutionary, using 5 features in both approaches.

Table 6.4 shows the features that correspond to the optimal solutions obtained using the neuroevolutionary and binary approaches for the *colon* dataset. The precision, number of features and features selected in each solution are indicated. It can be observed that the number of solutions and the number of features of each solution using the neuroevolutionary approach are smaller. Feature 1 is present in all solutions. Feature 513 is selected for 2 neuroevolutionary solutions and 5 binary solutions. Features 2001, 2003, 2005, 2008, 2010, 2011, 2015, 2019 and 2020 are present in binary solutions. Solutions B6 to B10 have a precision of 1.000 and are very similar, sharing a large number of features.

Table 6.4 Optimal solutions from the final population for dataset *colon*

	Neuroevolutionary			Binary									
P	0.865	0.971	0.974	0.854	0.971	0.972	0.973	0.974	1.000	1.000	1.000	1.000	1.000
f_1	2	3	7	14	15	16	21	22	23	24	24	22	22
Feature	**N1**	**N2**	**N3**	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**
780	■												
1184		■											
769			■										
773			■										
785			■										
833			■										
837			■										
853			■										
513		■	■	■	■	■	■	■					
1	■	■	■	■	■	■	■	■	■	■	■	■	■
2001				■	■	■	■	■	■	■	■	■	■
2003				■	■	■	■	■	■	■	■	■	■
2005				■	■	■	■	■	■	■	■	■	■
2008				■	■	■	■	■	■	■	■	■	■
2010				■	■	■	■	■	■	■	■	■	■
2011				■	■	■	■	■	■	■	■	■	■
2015				■	■	■	■	■	■	■	■	■	■
2019				■	■	■	■	■	■	■	■	■	■
2020				■	■	■	■	■	■	■	■	■	■
2002				■	■	■	■	■	■	■	■	■	
2013				■	■	■	■	■	■	■	■		■
1740						■	■	■	■	■	■	■	■
42							■	■	■	■	■	■	■
187							■	■	■	■	■	■	■
498							■	■	■	■	■	■	■
1955							■	■	■	■	■	■	■
2007							■	■	■	■	■	■	■
544									■	■	■	■	■
632									■	■	■	■	■
1464									■	■	■	■	■
1466									■	■	■	■	■
1497									■	■	■	■	■
2017				■	■	■					■		
883							■	■					
1483					■	■							
1687							■	■					
102								■					
2004										■			

6.5 Conclusions

This study proposes a neuroevolutionary approach to deal with feature selection problems by using multiobjective evolutionary algorithms. Considering n-dimensional datasets, to perform feature selection using binary representations or exhaustive search becomes impractical for a large n. In this context, the proposed approach can drastically reduce the search space by using Artificial Neural Networks to provide the most important features to classify the data with maximum precision. Since the number of features and the classification precision are conflicting objectives, by using multiobjective optimization a set of solutions (Pareto front) with different tradeoffs between the objectives can be obtained.

The methodology was applied to nine datasets with different number of features, samples and classes. To compare the results, a binary representation was also applied. When comparing the Pareto front of both representations (in terms of *hypervolume*), the neuroevolutionary approach presented better (or equal) results for eight of nine datasets.

For each dataset, an optimal solution was selected from the Pareto front considering the point closest to the knee of the curve (to give an equal relationship between classifier precision and the number of features). When comparing these points in both representations, for seven of nine datasets the neuroevolutionary approach presented better (or equal) results in terms of classifier precision. Different results were also achieved for the number of features. Only one dataset presented better results for binary approach. However, it is important to point out that by using the neuroevolutionary approach, the search space is drastically reduced, since the parameters of ANN are being evolved instead of the binary representation for each feature.

By including classifier parameters in the optimization, the algorithm was able to find the best combination of C (regularization) and kernel gamma (of the SVM Classifier) for each dataset in order to reach better classification precision.

For each dataset, an optimal solution was selected from the Pareto front considering the point closest to the knee of the curve (to give an equal relationship between classifier precision and the number of features). When comparing these points in both representations, for seven of nine datasets the neuroevolutionary approach presented better (or equal) results in terms of classifier precision. Different results were also achieved for the number of features. Only one dataset presented better results for binary approach. A detailed analysis of the results for database *colon* showed that the neuroevolutionary approach presented consistency by finding two key features (1 and 513) in all non-dominated solutions. On the other hand, results for the binary approach contain these features among a higher number of others to achieve equal or slightly better classifier precision. Also, it is important to point out that by using the neuroevolutionary approach, the search space is drastically reduced, since the parameters of ANN are being evolved instead of the binary representation for each feature.

Future works can address different parameters or kernel functions for the SVM classifier, or even the use of other classifiers to perform the classification. Other ANN topologies can also be considered.

Acknowledgments This work has been supported by FCT - Fundação para a Ciência e Tecnologia in the scope of the projects: PEst-OE/EEI/UI0319/2014, UID/MAT/00013/2013, UID/CEC/00319/2019 and the European project MSCA-RISE-2015, NEWEX, with reference 734205.

References

1. Guyon I, Gunn S, Nikravesh M, Zadeh LA (2008) Feature extraction: foundations and applications, vol 207. Springer
2. Unler A, Murat A, Chinnam RB (2011) mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. Inf Sci 181:4625–4641
3. Hancer E, Xue B, Zhang M, Karaboga D, Akay B (2018) Pareto front feature selection based on artificial bee colony optimization. Inf Sci 422:462–479
4. Bi J (2003)Multi-objective programming in SVMs. In: Proceedings of the 20th international conference on machine learning (ICML-03)
5. Igel C (2005) Multi-objective model selection for support vector machines. In: International conference on evolutionary multi-criterion optimization
6. Oliveira LS, Morita M, Sabourin R (2006) Feature selection for ensembles using the multi-objective optimization approach. In: Multi-objective machine learning. Springer, pp 49–74
7. Gaspar-Cunha A (2010) Feature selection using multi-objective evolutionary algorithms: application to cardiac SPECT diagnosis. In: Advances in bioinformatics. Springer, pp 85–92
8. Pinto R, Silva H, Duarte F, Nunes J, Gaspar-Cunha A (2019) Neuroevolutionary multiobjective methodology for the optimization of the injection blow molding process. In: International conference on evolutionary multi-criterion optimization
9. Denysiuk R, Duarte FM, Nunes JP, Gaspar-Cunha A (2017) Evolving neural networks to optimize material usage in blow molded containers. In: EUROGEN - international conference on evolutionary and deterministic methods for design optimization and control with applications to industrial and societal problems
10. Denysiuk R, Gaspar-Cunha A, Delbem ACB (2019) Neuroevolution for solving multiobjective knapsack problems. Expert Syst Appl 116:65–77
11. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45:427–437
12. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur J Oper Res 181:1653–1669
13. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International conference on parallel problem solving from nature

Chapter 7
Multi-objective Optimization in the Build Orientation of a 3D CAD Model

Marina A. Matos, Ana Maria A. C. Rocha, Lino A. Costa, and Ana I. Pereira

Abstract Over the years, rapid prototyping technologies have grown and have been implemented in many 3D model production companies. A variety of different additive manufacturing (AM) techniques are used in rapid prototyping. AM refers to a process by which digital 3D design data is used to build up a component in layers by depositing material. Several high-quality parts are being created in various engineering materials, including metal, ceramics, polymers and their combinations in the form of composites, hybrids, or functionally classified materials. The orientation of 3D models is very important since it can have a great influence on the surface quality characteristics, such as process planning, post-processing, processing time and cost. Thus, the identification of the optimal build orientation for a part is one of the main issues in AM. The quality measures to optimize the build orientation problem may include the minimization of the surface roughness, build time, need of supports, maximize of the part stability in building process or part accuracy, among others. In this paper, a multi-objective approach was applied to a computer-aided design model using MATLAB® multi-objective genetic algorithm, aiming to optimize the support area, the staircase effect and the build time. Preliminary results show the effectiveness of the proposed approach.

Keywords Additive Manufacturing · 3D Printing · Multi-objective Optimization · Build Orientation

M. A. Matos (✉) · A. M. A. C. Rocha · L. A. Costa
ALGORITMI Center, University of Minho Campus Gualtar, 4710-057 Braga, Portugal
e-mail: aniram@live.com.pt

A. M. A. C. Rocha
e-mail: arocha@dps.uminho.pt

L. A. Costa
e-mail: lac@dps.uminho.pt

A. I. Pereira
Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
e-mail: apereira@ipb.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_7

7.1 Introduction

Additive manufacturing (sometimes called 3D printing) refers to a process by which 3D computer-aided design (CAD) models are used to build 3D objects, by adding layer-by-layer material. The manufacturing processes in layers are currently used in several areas to fabricate end-use products in aircraft industry, medical implants, jewelery, footwear industry, automotive industry and fashion products [1, 2]. Additive manufacturing technologies have grown over the years due to their effectiveness in the development of the prototype model in a reduced production time and cost. Depending on the specific 3D printing technology and the complexity of the 3D model, it is important to consider support structures and how they may affect the final result. In this work, a 3D printer using Fused Deposition Modeling (FDM) is considered. FDM extrudes a melted filament onto a build surface along a predetermined path. As the material is extruded, it cools, forming a solid surface providing the foundation for the next layer of material to be built upon. This is repeated layer-by-layer until the object is completed. With FDM printing, each layer is printed as a set of heated filament threads which adhere to the threads below and around it. Each thread is printed slightly offset from its previous layer. This allows a model to be built up to angles of 45°, allowing prints to expand beyond its previous layers width. When a feature is printed with an overhang beyond 45°, it can sag and requires support material beneath it to hold it up [3, 4]. Thus, the accuracy of the printed object depends on the orientation of the part on the printer platform, that is, the part must have the correct orientation in order to improve the quality of the surface. Different measures can be considered to determine the optimal build orientation taking into account factors such as staircase effect, model precision, build time, structure support and model stability [1, 5]. The optimal build orientation of a model helps in the accuracy of the part, reduces the number of supports generated and the build time of the parts, and consequently decreases the final costs.

Several approaches have been carried out to determine the orientation of a model based on single-objective optimization using objective functions such as the build height, staircase effect, volumetric error, volume of support structures and total contact area of the part with the support structures, surface quality, surface roughness and build deposition time [6, 7]. Recently, multi-objective approaches have been developed to determine the optimal object building orientation in the construction of CAD models, essentially by reducing the multi-objective problem to a single-objective one using classical scalarization methods [8–13]. A genetic algorithm was used in Brika et al. [8] for solving a multi-objective build orientation problem. They optimized several variables, yield and tensile strength, elongation and vickers hardness, for material properties used, surface roughness, support structure and build time and cost. The particle swarm optimization algorithm was used in Li et al. [10] to solve a multi-objective optimization problem in order to get the desired orientations for the support area, construction time and surface roughness. In the paper of Phatak and Pande [11] a genetic algorithm was used to optimize a weighted average of five normalized evaluation criteria (build height, staircase error factor, material utilization factor, part surface area in contact with support structures and volume of support structures) based on their relevance to the rapid prototyping process. In Das et al. [14], the errors related to the staircase effect and the support volume were studied, using weights to find the best orientation of a spherical model. Cheng et al. [15] formulated a multi-objective optimization problem focused on the surface quality and production cost of the parts, obtaining solutions for all types of surfaces, whether with complex geometries or not, or even for curved surfaces. The multi-objetive approach presented by Byun et al. [16] intend to reduce surface roughness, construction time and part cost. The goal was to find the ideal orientation of a 3D model by applying the Technique for Order Preference by Similarity to Ideal Solution and weight methods. Nezhad et al. [17] proposed an Optimized Pareto Based Part Orientation algorithm in order to optimize the minimum construction time, the support volume and surface finish. The applied method does not use weights and optimizes objectives simultaneously and independently. A multi-objective optimization approach, using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Multi-Objective Particle Swarm Optimization algorithm, considering as objective functions the surface roughness and the build time, for different models, was developed by Padhye and Deb [18]. Gurrala and Regalla [19] also applied the NSGA-II algorithm to optimize the strength of the model and its volumetric shrinkage as objective functions. Through the Pareto front, they concluded that with the shrinkage of the part their strength increases in the horizontal and vertical directions. A different study addressing how an easily removed support structure might be designed using less material and build time and leaving fewer artifacts on the specimen surface can be seen in Kuo et al. [20]. There, a cost-based formulation is employed to find a compromise between cost and surface profile error induced by specimen weight.

In this work, the optimization of the final printed object surface is addressed, based on the minimization of the staircase effect, the area of the object in contact with the supporting structures and the build time. Here, a multi-objective optimization approach is proposed to obtain the orientation of the Rear Panel Fixed model taking into account the compromise between combinations of two measures mentioned above. We present some preliminary experiments showing the Pareto fronts and discuss different trade-offs between the objectives.

This article is organized as follows. Section 7.2 introduces the orientation problem, the quality measures and the multi-objective optimization approach. The numerical experiments are presented and discussed in Sect. 7.3. Finally, Sect. 7.4 contains the conclusions of this study and some recommendations for future work.

7.2 Multi-objective Approach

7.2.1 *Optimization Problem*

In this study, a multi-objective optimization to determine the orientation of the construction of a 3D CAD model is used. It intends to simultaneously minimize more than one measure of the quality of the printed object.

The measures involved in this study are the staircase effect, the area of the object in contact with the support structures and the build time.

Although we intend to study three measures of quality, in this study we will perform the multi-objective optimization of the combinations of two objective functions and three objective functions simultaneously. Thus the multi-objective optimization problem is given by

$$\begin{array}{ll} \min & f\left(\theta_x, \theta_y\right) = \left\{ f_1\left(\theta_x, \theta_y\right), \ldots, f_k\left(\theta_x, \theta_y\right) \right\} \\ \text{s.t.} & 0 \leq \theta_x \leq 180 \\ & 0 \leq \theta_y \leq 180 \end{array} \tag{7.1}$$

where k is the number of objective functions and θ_x and θ_y are the rotation angles along the x-axis and the y-axis, respectively.

In the following, the quality measures based on staircase effect, support area and the required build time are described.

7.2.2 *Quality Measures*

7.2.2.1 Support Area

A measure of the quality of the printed object is the quantity of support area, since it affects post-processing and surface finish [9]. The support area is defined as the total area of the downward-facing facets that is equivalent to the total contact area of the external supports with the object [7, 9].

The support area, SA, is defined by

$$SA = \sum_i A_i \left| d^T n_i \right| \delta \tag{7.2}$$

where A_i is the area of the triangular face i, d is the unit vector of the direction of construction, n_i is the normal unit vector of the triangular face i and the initial function is given by $\delta = 1$ if $d^T n_i < 0$ and $\delta = 0$ if $d^T n_i > 0$ [9]. In this study, we used the direction vector $d = (0, 0, 1)$, because our 3D printer only moves on the x-axis and y-axis, since the base platform (z-axis) is fixed.

7.2.2.2 Staircase Effect

The orientation and layer thickness are the most important factors that affect the superficial roughness [21]. Kattethota et al. [21] studied the staircase effect (SE) of a 3D model based on the deviation between the actual and desired surfaces. It means that the greater the deviation between the two surfaces (real and desired), the greater the length of the layer and the lower the orientation of the construction of the part. The staircase effect, SE, is defined by

$$SE = \sum_{i} \begin{cases} \frac{t}{\tan(\theta_i)}, & \text{if } \tan(\theta_i) \neq 0 \\ 0, & \text{if } \tan(\theta_i) = 0 \end{cases} \tag{7.3}$$

where t is the layer thickness and θ_i is the angle between triangle facet i of model surface and build orientation (d).

7.2.2.3 Build Time

As considered in Jibin [9] the build time encompasses the scanning time and the preparation time. The scanning time includes solid scanning time, contour scanning time and support scanning time, where the solid and contour scanning times are independent of the part building direction and the support scanning time depends on the volume of supports. The preparation time of the model covers the time required for the platform to move down during the construction of each layer, the scraping time of this and other preparation times. Thus, the preparation time depends on the total number of slices of the solid, which is dependent on the height of the building direction. Therefore, minimizing this height and consequently the number of layers, can decrease the construction time of the part [7, 9].

The build time, BT, is given by

$$BT = \max_{i} \left(d^T v_i^1, d^T v_i^2, d^T v_i^3\right) - \min_{i} \left(d^T v_i^1, d^T v_i^2, d^T v_i^3\right) \tag{7.4}$$

where d is the direction vector and v_i^1, v_i^2, v_i^3 are the vertex triangle facets i.

7.2.3 Multi-objective Genetic Algorithm

In this work, the elitist Non-dominated Sorting Genetic Algorithm II (NSGA-II) proposed by Deb [22] is used. This is a multi-objective genetic algorithm that mimics the natural evolution of the species. Evolution starts from a population of individuals randomly generated. Each individual represents a potential solution of the multi-objective optimization problem. In NSGA-II, each individual in the current population is evaluated using a Pareto ranking and a crowding measure. First the best rank is assigned to all the non-dominated individuals in the current population. Solutions with the best rank are removed from the current population. Next, the second best rank is assigned to all the non-dominated solutions in the remaining population. In this manner, ranks are assigned to all solutions in the current population. The fittest individuals have a higher probability of being selected to generate new ones by genetic operators. NSGA-II uses a binary tournament selection based on non-domination rank and crowding distance to select a set of parent solutions. When two solutions are selected, the one with the lowest non-domination rank is preferred. Otherwise, if both solutions belong to the same rank, then the solution with the higher crowding distance is selected. Next, genetic operators such as recombination and mutation are applied to create an offspring population. Then, the two populations are merged together to form a combined population that is sorted according to different non-dominated fronts. If the size of the first non-dominated front is smaller then the population size, all members of this front are chosen for the new population. The remaining members of the population are chosen from subsequent non-dominated fronts in the order of their ranking.

The MATLAB® function `gamultiobj` [23] provided in the Global Optimization Toolbox will be used in order to approximate the Pareto fronts of the multi-objective problems with each combinations of two objective functions. The `gamultiobj` function implements a multi-objective genetic algorithm that is a variant of the elitist NSGA-II [22]. This function provides a set of algorithm options related with customizing randomization key properties, algorithm properties and termination criteria.

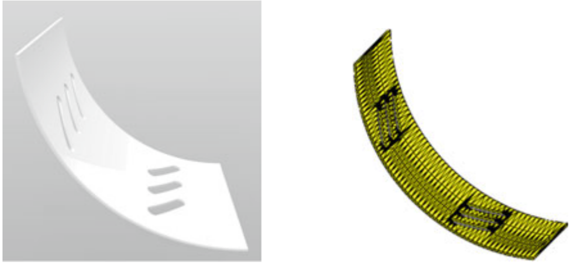
7.3 Experiments

7.3.1 Model

The 3D CAD model used in this study is a Rear Panel Fixed (see Fig. 7.1a) that has vents on either side. The size of the model is different from the side panels, but the side panels for left and right are equal.

Initially, the CAD model is converted into STL (STereoLithography), which is the default file type used by the most common 3D print file formats (see Fig. 7.1b).

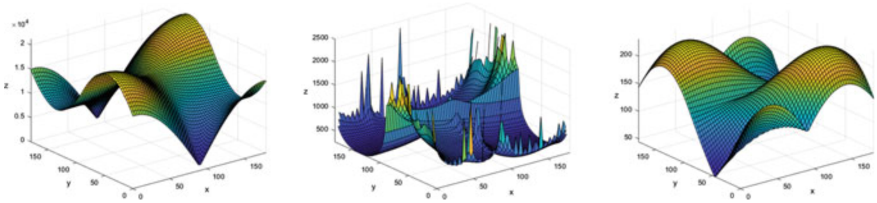
The STL file is an approximation (tessellation) of the CAD model, where the geometric characteristics of the 3D model are depicted. Thus, the model is represented by a mesh of triangles, describing only the surface geometry of a three-dimensional object without any representation of color, texture or other common attributes of the CAD model. It was defined using 3008 triangles, a volume of 46.2 cm^3 and 676 slices for a layer thickness of 0.2 mm (layer thickness used in this work). Figure 7.2a–c depict the SA, SE and BT objective functions landscapes for the Rear Panel Fixed model. These objective functions are nonconvex with multiple local optima. Moreover, it can be observed that the minimizers of each objective function are different.



(a) Rear Panel Fixed model

(b) Rear Panel Fixed STL model

Fig. 7.1 Rear Panel Fixed model



(a) *SA* objective function

(b) *SE* objective function

(c) *BT* objective function

Fig. 7.2 Rear Panel Fixed objective functions

Therefore, these objectives are conflicting each other and there exist different trade-off solutions that represent different compromises between the objectives.

7.3.2 Implementation Details

Firstly, the combination of two of the quality measures, the support area, the staircase effect and the build time of the part was considered, and the following three multi-objective optimization problems were formulated:

- SA *versus* SE—problem (7.1) with $f_1 = SA$ and $f_2 = SE$;
- SA *versus* BT—problem (7.1) with $f_1 = SA$ and $f_2 = BT$;
- SE *versus* BT—problem (7.1) with $f_1 = SE$ and $f_2 = BT$.

Secondly, a multi-objective optimization of the three objective functions simultaneously is considered, where SA *versus* SE *versus* BT denotes solving the problem (7.1) with $f_1 = SA$, $f_2 = SE$ and $f_3 = BT$.

In order to solve the multi-objective optimization problems, the MATLAB® `gamultiobj` function was used with default values, thus a population size and a maximum number of generations of 50 and 400, respectively. By default, the Pareto fraction is 0.35 and therefore, in each run, 18 non-dominated solutions are found

(0.35× population size). In addition, 30 independent runs were performed and the *Simplify 3D* software [24] (a 3D model printing simulator) was used to represent the solutions found for the Rear Panel Fixed model.

In the following sections the results for the different combinations of two objectives (SA *versus* SE, SA *versus* BT, and SE *versus* BT problems) as well as the results for the three objectives (SA *versus* SE *versus* BT problem) are presented.

In all figures, the set of non-dominated solutions obtained among the 30 independent runs are plotted with a blue dot. From this overall set of solutions, the non-dominated ones were selected and marked with a red circle. Representative solutions will be selected to discuss trade-offs between objectives and identify the characteristics associated with these solutions.

7.3.3 Results for the SA versus SE Problem

Figure 7.3 depicts the Pareto front for SA *versus* SE problem, where the set of non-dominated solutions for all runs are plotted with a blue dot.

Table 7.1 presents the representative non-dominated solutions selected from the Pareto front, that were marked with a red circle in Fig. 7.3.

Solutions A and G are the extremes of the Pareto front, where solution A has the best SA value and the worst SE value. Conversely, solution G is the worst in terms

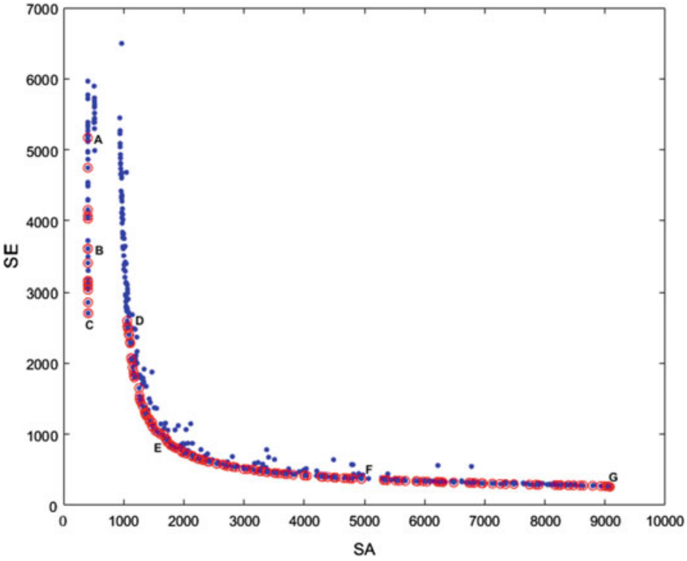
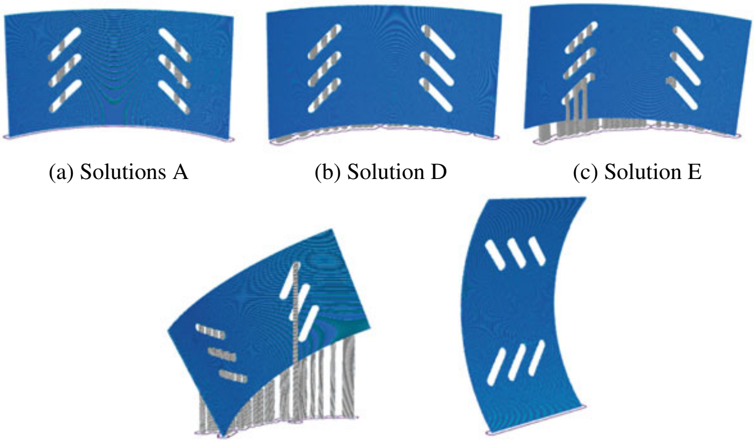


Fig. 7.3 Pareto front of the Rear Panel Fixed model for the SA *versus* SE problem

Table 7.1 Representative non-dominated solutions for the SA *versus* SE problem

Solutions	θ_x	θ_y	SA	SE
A	90.00	0.00	406.45	5168.99
B	90.00	0.01	407.05	3602.96
C	89.99	0.02	412.11	2708.20
D	90.99	0.95	1057.61	2596.88
E	94.44	2.76	1695.75	943.42
F	68.49	160.87	4951.71	372.42
G	180.00	44.95	9078.07	260.39


(a) Solutions A

(b) Solution D

(c) Solution E

(d) Solution F

(e) Solution G

Fig. 7.4 Representative solutions for the SA *versus* SE problem

of SA and the best in terms of SE. These solutions correspond to the lowest values of SA and SE that can be observed in Fig. 7.2a, b, respectively.

From Table 7.1, it is possible to observe that solutions A, B, C and D have very similar orientation angles, although different SA and SE values, verifying a reduction in the staircase effect and an increase in the support area, in particular in the solution D. Solutions D and E are visually very similar, as can be seen in Fig. 7.4b, c, respectively, but the solution E requires more supports. From solutions A to G, there is a significant change in the orientation of the part.

7.3.4 Results for the SA versus BT Problem

Figure 7.5 shows the Pareto front for the model Rear Panel Fixed when SA and BT are the objectives to minimize simultaneously (SA *versus* BT). Table 7.2 presents the orientation angles and objective values for representative non-dominated solutions selected from the Pareto front. These solutions are shown in Fig. 7.6. It is possible to see that from point A to point B there is no significant change in terms of BT, but there is a great increase in the support area. From solutions B to C, the value of SA increases, while the value of BT decreases. Solution D is one of the extremes of the Pareto front, being minimum of BT function (as it can also be seen in Fig. 7.2c), but it is a bad solution in terms of SA.

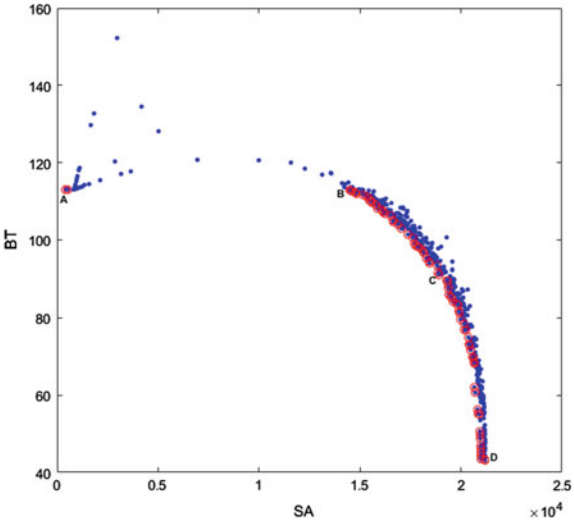
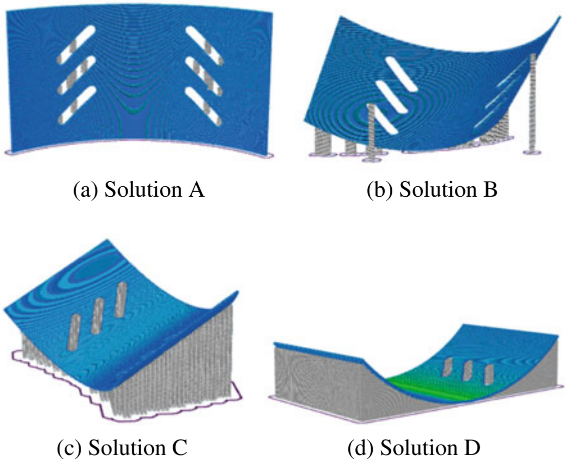


Fig. 7.5 Pareto front of the Rear Panel Fixed model for the SA *versus* BT problem

Table 7.2 Representative non-dominated solutions for the SA *versus* BT problem

Solutions	θ_x	θ_y	SA	BT
A	90.00	0.00	406.45	113.00
B	122.37	151.85	14457.00	112.98
C	144.87	140.93	19038.00	89.42
D	180.00	135.00	21190.00	43.17

Fig. 7.6 Representative solutions for the SA *versus* BT problem


(a) Solution A

(b) Solution B

(c) Solution C

(d) Solution D

7.3.5 Results for the SE versus BT Problem

In Fig. 7.7, the solutions obtained in the objective space when optimizing SE *versus* BT problem are presented. Solutions A and F are the extremes of the Pareto front. There is a significant improvement in the BT value, when comparing solutions B and C, although a negligible decrease in SE. From solutions A to F the part is placed lying down, as can be seen in Fig. 7.8, decreasing the height of the part (decreasing BT). Solutions D, E and F are visually similar as can be seen in Fig. 7.8d, e, f (Table 7.3).

7.3.6 Results for the SA versus SE versus BT Problem

In this section, the results of the multi-objective optimization of the three objective functions simultaneously (SA *versus* SE *versus* BT problem) are presented. The problem was optimized using the MATLAB® function `gamultiobj` with default values, as described in the Sect. 7.3.2.

Figure 7.9 shows the non-dominated solutions obtained for the Rear Panel Fixed model. Three solutions regarding the extreme solutions for each objective are represented by A, B and C, corresponding to the angles (90.00, 0.00), (180.00, 44.95), (180.00, 135.00), respectively (corresponding to Figs. 7.6a, 7.8a, f, respectively).

In Fig. 7.10 the two-dimensional projections of the Pareto front of the Rear Panel Fixed model for the SA *versus* SE *versus* BT problem are presented. It can be seen that the number of non-dominated solutions is larger than the one obtained with two objective combinations.

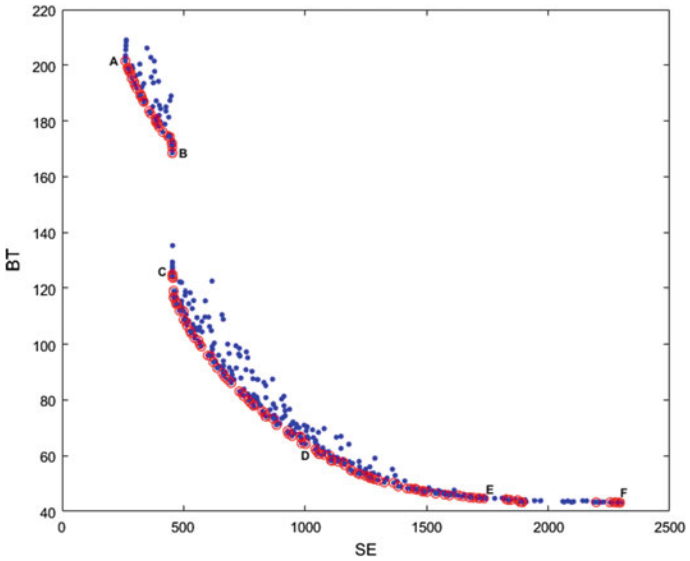
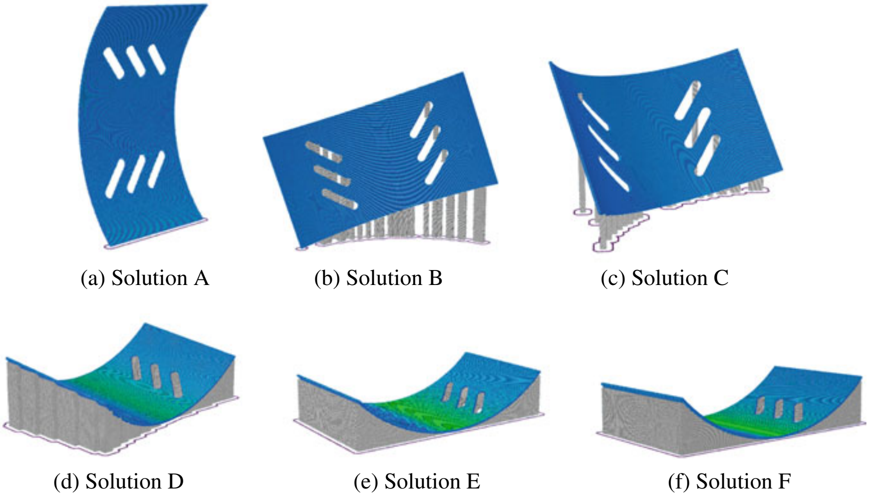


Fig. 7.7 Pareto front of the Rear Panel Fixed model for the SE *versus* BT problem



(a) Solution A (b) Solution B (c) Solution C

(d) Solution D (e) Solution E (f) Solution F

Fig. 7.8 Representative solutions for the SE *versus* BT problem

Table 7.3 Representative non-dominated solutions for the SE *versus* BT problem

Solutions	θ_x	θ_y	SE	BT
A	180.00	44.95	260.39	201.52
B	81.53	163.10	453.88	168.44
C	114.95	161.23	453.89	124.88
D	164.87	135.55	1002.11	64.24
E	178.93	134.99	1735.08	44.67
F	180.00	135.00	2295.19	43.15

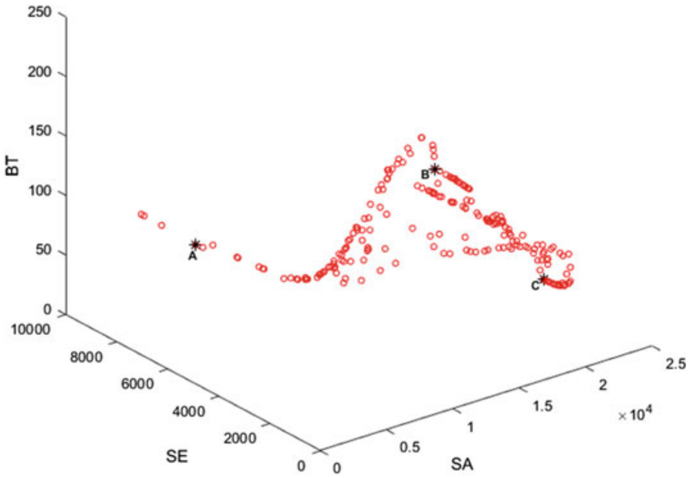
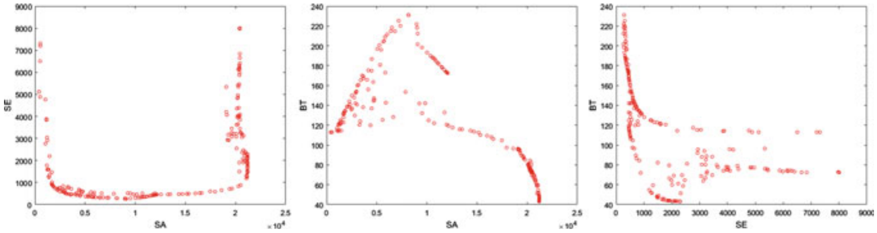


Fig. 7.9 Pareto front of the Rear Panel Fixed model for the SA *versus* SE *versus* BT problem



(a) *SA* and *SE*

(b) *SA* and *BT*

(c) *SE* and *BT*

Fig. 7.10 2D Projections of the Pareto front for the SA *versus* SE *versus* BT problem

7.3.7 Discussion of the Results

The first three combinations of multi-objective optimization problems solved allow to perceive the extremes and the compromise between objectives. Moreover, there are solutions that belong to the Pareto optimal set of the problems solved, i.e., their images belong to the Pareto fronts of the different multi-objective optimization problems. This is the case of solution (90.00, 0.00) that minimizes SA and appears on the Pareto fronts of SA *versus* SE and SA *versus* BT problems. In addition, the solution that corresponds to the (180.00, 44.95) orientation was found for the SA *versus* SE and SE *versus* BT problems, optimizing the SE function. It is also verified that the solution (180.00, 135.00) optimizes BT, as can be seen in the solutions of SA *versus* BT and SE *versus* BT. When BT is one of the objective functions involved in the multi-objective problem (combinations SA *versus* BT and SE *versus* BT), some representative solutions put the part lying down, as can be seen in solution D of Fig. 7.6 and in solutions D, E, and F of Fig. 7.8, as expected because BT function intends to minimize its height. However, with the combination SA *versus* SE, as expected, no solution that position the part lying down exists.

In the multi-objective simultaneous optimization of the three objective functions, all the solutions obtained with the combinations of two objectives and others that represent other trade-offs between objective functions were found.

7.4 Conclusions and Future Work

In this paper, the build orientation optimization of a given object - Rear Panel Fixed model—was addressed based on three quality measures: the total support contact area, the staircase effect and the build time.

First, a multi-objective optimization approach was proposed for three different combinations of two objectives: SA *versus* SE, SA *versus* BT, and SE *versus* BT. Some preliminary experiments were presented for the three different combinations. The Pareto fronts obtained and the different trade-offs between the objectives were discussed. It was also verified that some solutions were found repeatedly in different combinations of objective functions. The results showed the effectiveness of the proposed approach since it was possible to find different solutions to optimize the various combinations.

Then, the three objective functions were optimized simultaneously. From the Pareto front we may conclude that a larger number of solutions was obtained when comparing to the ones obtained through two objective combinations, as well as, new trade-off solutions were found. It was observed that, for all problems, the Pareto fronts have nonconvexities and discontinuities.

In the future, we intend to perform a multi-objective optimization using other objective functions and test more difficult models.

Acknowledgments This work has been supported and developed under the FIBR3D project - Hybrid processes based on additive manufacturing of composites with long or short fibers reinforced thermoplastic matrix (POCI-01-0145-FEDER-016414), supported by the Lisbon Regional Operational Programme 2020, under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). This work has been also supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Unit Project Scope: UIDB/00319/2020 and UIDB/05757/2020 and the European project MSCA-RISE-2015, NEWEX, with reference 734205.

References

1. Pandey P, Reddy NV, Dhande S (2007) Part deposition orientation studies in layered manufacturing. J Mater Process Technol 185(1–3):125–131
2. Gao W, Zhang Y, Ramanujan D, Ramani K, Chen Y, Williams CB, Wang CC, Shin YC, Zhang S, Zavattieri PD (2015) The status, challenges, and future of additive manufacturing in engineering. Comput-Aided Des 69:65–89
3. King WE, Anderson AT, Ferencz R, Hodge N, Kamath C, Khairallah SA, Rubenchik AM (2015) Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. Appl Phys Rev 2(4):041304
4. Turner BN, Gold SA (2015) A review of melt extrusion additive manufacturing processes: II. materials, dimensional accuracy, and surface roughness. Rapid Prototyping J 21(3):250–261
5. Wang WM, Zanni C, Kobbelt L (2016) Improved surface quality in 3d printing by optimizing the printing direction. In: Computer graphics forum, vol 35. Wiley Online Library, pp 59–70
6. Pereira S, Vaz A, Vicente L (2018) On the optimal object orientation in additive manufacturing. Int J Adv Manuf Technol 98(5–8):1685–1694
7. Rocha AMAC, Pereira AI, Vaz AIF (2018) Build orientation optimization problem in additive manufacturing. In: International conference on computational science and its applications. Springer, Berlin, pp 669–682
8. Brika SE, Zhao YF, Brochu M, Mezzetta J (2017) Multi-objective build orientation optimization for powder bed fusion by laser. J Manuf Sci Eng 139(11):111011
9. Jibin Z (2005) Determination of optimal build orientation based on satisfactory degree theory for RPT. In: Proceedings of the ninth international conference on computer aided design and computer graphics. CAD-CG '05, IEEE Computer Society, USA, pp 225–230
10. Li A, Zhang Z, Wang D, Yang J (2010) Optimization method to fabrication orientation of parts in fused deposition modeling rapid prototyping. In: 2010 international conference on mechanic automation and control engineering. IEEE, pp 416–419
11. Phatak AM, Pande S (2012) Optimum part orientation in rapid prototyping using genetic algorithm. J Manuf Syst 31(4):395–402
12. Ga B, Gardan N, Wahu G (2019) Methodology for part building orientation in additive manufacturing. Comput-Aided Des Appl 16(1):113–128
13. Matos MA, Rocha AMAC, Costa LA, Pereira AI (2019) A multi-objective approach to solve the build orientation problem in additive manufacturing. In: International conference on computational science and its applications. Springer, Berlin, pp 261–276
14. Das P, Chandran R, Samant R, Anand S (2015) Optimum part build orientation in additive manufacturing for minimizing part errors and support structures. Procedia Manuf 1:343–354
15. Cheng W, Fuh J, Nee A, Wong Y, Loh H, Miyazawa T (1995) Multi-objective optimization of part-building orientation in stereolithography. Rapid Prototyping J 1(4):12–23
16. Byun HS, Lee KH (2006) Determination of optimal build direction in rapid prototyping with variable slicing. Int J Adv Manuf Technol 28(3–4):307
17. Nezhad AS, Vatani M, Barazandeh F, Rahimi A (2009) Multi objective optimization of part orientation in stereolithography. In: WSEAS international conference. proceedings. mathematics and computers in science and engineering. WSEAS, p 5

18. Padhye N, Deb K (2011) Multi-objective optimisation and multi-criteria decision making in SLS using evolutionary approaches. Rapid Prototyping J 17(6):458–478
19. Gurrala PK, Regalla SP (2014) Multi-objective optimisation of strength and volumetric shrinkage of FDM parts: a multi-objective optimization scheme is used to optimize the strength and volumetric shrinkage of FDM parts considering different process parameters. Virtual Phys Prototyping 9(2):127–138
20. Kuo YH, Cheng CC, Lin YS, San CH (2018) Support structure design in additive manufacturing based on topology optimization. Struct Multi Optim 57(1):183–195
21. Kattethota G, Henderson M (1998) A visual tool to improve layered manufacturing part quality. In: Proceedings of solid freeform fabrication symposium, pp 327–334
22. Deb K (2001) Multi-Objective Optimization Using Evolutionary Algorithms, vol 16. Wiley, New York, NY, USA
23. MATLAB (2019) version 9.6.0.1214997 (R2019a). Natick, Massachusetts, The MathWorks Inc
24. SIMPLIFY3D, Integrated Software Solutions (2017). Simplify3D LLC., Legal Dept

Chapter 8
The Effects of Crowding Distance and Mutation in Multimodal and Multi-objective Optimization Problems

Mahrokh Javadi, Heiner Zille, and Sanaz Mostaghim

Abstract In this paper, we study the effects of a modified crowding distance method and a Polynomial mutation operator on multimodal multi-objective optimization algorithms. Our goal is to provide an in-depth analysis on these two modifications which we apply to NSGA-II: The weighted sum crowding distance and the neighborhood-based mutation operator. Furthermore, we examine the performance of the proposed weighted sum crowding distance method under different weight values to find a trend for the behaviour of the proposed algorithm. We compare the different variations of the proposed method with state-of-the-art algorithms and the baseline NSGA-II. The results show that our modifications can improve the functionality of NSGA-II on multimodal multi-objective problems.

Keywords Multi-modality · Multi-modal problems · Multi-objective Optimization · Evolutionary Algorithms · Non-dominated Sorting Genetic Algorithm

8.1 Introduction

In real-world applications, there are many problems involving several conflicting objectives which need to be optimized at the same time. These problems are usually referred to as *Multi-Objective Problems* (MOP). In such problems, improving one of the objectives can have a negative impact on other objectives [1].

Multi-objective optimization problems are mathematically formulated as follows (we consider minimization problems, without loss of generality):

M. Javadi (✉) · H. Zille · S. Mostaghim
Faculty of Computer Science, Otto von Guericke University, Magdeburg, Germany
e-mail: mahrokh1.javadi@ovgu.de

H. Zille
e-mail: heiner.zille@ovgu.de

S. Mostaghim
e-mail: sanaz.mostaghim@ovgu.de

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_8

$$\begin{aligned}
&\text{minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\
&\text{subject to} \qquad\quad \mathbf{x} \in S \subset \mathbb{R}^D \\
&\qquad\qquad\qquad g_i(\mathbf{x}) \leq 0, i = 1, 2, \ldots, k \\
&\qquad\qquad\qquad h_j(\mathbf{x}) = 0, j = 1, 2, \ldots, p
\end{aligned}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ is a D–dimensional decision vector, $(f_1, f_2, \ldots, f_M)$ a M–dimensional objective vector and $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are inequality and equality constraints in decision space. In order to deal with these problems, the concept of domination can be used. Given two vectors $\mathbf{x}$, $\mathbf{y} \in S$, $\mathbf{x}$ is said to be dominated by $\mathbf{y}$ (denoted by $\mathbf{y} \prec \mathbf{x}$) if and only if $\forall j \in \{1, \ldots, M\}$, $f_j(\mathbf{y}) \leq f_j(\mathbf{x})$, and $\exists k \in \{1, \ldots, M\}, f_k(\mathbf{y}) < f_k(\mathbf{x})$ [2]. A solution which is not dominated by any other solution in the decision space is called a Pareto-optimal solution. The set of such optimal solutions in decision space is called *Pareto-Set* (PS), and the corresponding solutions in objective space are called *Pareto-Front* (PF) [3]. The goal of multi-objective optimization algorithms is to find a set of non-dominated solutions with a good approximation of the PF both in terms of convergence and diversity [1]. By using the definition of domination, there is no guarantee that finding all of the solutions in the PF leads to finding all solutions which actually belong to the PS, since two solutions in decision space can map to one point in the objective space. This class of problems is referred to as multimodal multi-objective problems by Liang and Qu [2]. More precisely, in a multimodal problem, there are multiple subsets of the PS which map to the same objective function values, therefore the PF can be approximated by just finding one of these subsets of the PS. However, decision makers are often interested in high diversity both in decision and objective spaces. Hence, it might be practical to develop algorithms that can find multiple Pareto-optimal solutions in both decision and objective spaces for such multimodal problems.

In this paper, we modify the concept of crowding distance in both decision and objective spaces and investigate a neighborhood-based mutation operator. Both these approaches are based on our preliminary short study [4]. In this paper, we introduce them and evaluate their performances in both decision and objective spaces using various performance indicators and perform a detailed comparative study of the obtained PS and PF. In addition, we examine their effects separately and analyze the contribution of each of them on the approximation of PS and PF. We also evaluate the effects of different weight parameter values on the diversity of solutions in both decision and objective spaces. The remainder of this paper is organized as follows: In Sect. 8.2 the related works are briefly reviewed. Section 8.3 introduces the proposed algorithms and the novelty of the work in detail. In Sect. 8.4, the settings of the experiments are discussed. The results of the experiments and analysis are provided in Sect. 8.5. Finally conclusions and perspectives are presented in Sect. 8.6.

8.2 Related Work

In recent years, there has been an increasing amount of literature focusing on finding multimodal solutions for multi-objective optimization problems [5–7]. Some of these works aim to get a better approximation of the PS by increasing the diversity of solutions in decision space. However, this might not provide a better convergence to the PS [2].

The *Omni-optimizer Algorithm* [8] applies a crowding distance approach to the solutions in the decision space to preserve more solutions in decision space than the objective space. In this algorithm, the final crowding distance value for each solution is assigned based on the comparison of the average crowding distances in both decision and objective spaces: If the crowding distance value of each solution either in decision or objective space becomes larger than or equal to the average value, the maximum crowding distance is selected, otherwise the minimum of these two values is taken as the final crowding distance value. The provided modification was applied to the well-known *Non-dominated Sorting Genetic Algorithm (NSGA-II)* [1]. Zhou et al. [9] proposed a model resulting in enhanced approximations of both decision and objective spaces simultaneously. The population is classified into sub-populations in objective space, and the model increases the diversity of solutions in decision space by evaluating the diversity of the PS in each sup-population. The obtained solutions show better convergence to the PS and PF for the MOP in comparison with the Omni-optimizer algorithm. Liu et al. [10] proposed another approach called *Double-Niches Evolutionary Algorithm* (DNEA). This algorithm is an extension of the NSGA-II for multimodal MOPs. The performances of the DNEA are similar to those of the Omni-optimizer algorithm with the difference that it uses sharing functions required for fine-tuning the parameters in both decision and objective spaces.

Liang et al. [2] presented *DN-NSGA-II* which incorporates a niching algorithm in decision space. It contains two modifications of the original NSGA-II: The crowding distance method is used in decision space instead of objective space, and it creates a mating pool of solutions with a niching technique. The resulting algorithm was able to cover more solutions in the PS than the original NSGA-II algorithm.

Yue et al. [6] proposed *Multi-objective Particle Swarm Optimization using Ring topology by applying Special Crowding Distance* (MO-Ring-PSO-SCD). In this work, a ring topology is used to capture more optimal solutions in the decision space by making robust niches, and a special crowding distance method assists to preserve solutions in the PS. The results of this algorithm show significant improvement compared to NSGA-II, DN-NSGA-II and Omni-optimizer in terms of approximation of the PS in decision space.

In a recent work, Liang et al. [5] adopt the concept of a mutation bound process, which gives a second opportunity to perform mutation if the mutated solutions lie outside the boundaries of the decision space. They also use both non-dominated sorting in objective space and the crowding distance technique in decision space. Their proposed method, called *Multimodal Multi-Objective Differential Evolution* (MMODE), was applied to a differential evolution algorithm. The results of this algorithm show improvement in terms of diversity of solutions in both decision and objective spaces.

In the work presented by Liang et al. [7], an improved version of the SMPSO algorithm [11] was proposed. This SMPSO-MM makes use of creating neighborhoods in the decision space. Furthermore, they designed a special version of crowding distance on both decision and objective spaces to keep the obtained optimal solutions. The experimental results show that the mentioned algorithm could obtain better approximations of the PS than other state-of-the-art algorithms like the MO-Ring-PSO-SCD.

8.3 Proposed NSGA-II-WSCD-NBM

In this section, we modify the existing NSGA-II [1] with a weighted sum crowding distance method and a new Polynomial mutation operator in the so called NSGA-II-WSCD-NBM algorithm.

8.3.1 Weighted-Sum Crowding Distance Method

The classical Crowding Distance (CD) approach is typically used in the objective space to improve the diversity of the solutions in the objective space [1]. This approach leads to a better approximation of the PF, but it does not promise to preserve all the solution in the PS. Therefore, similar to the Omni-optimizer algorithm [8], we adopt the concept of crowding distance in both spaces to obtain the better approximation of the PS and PF. Our approach is called Weighted Sum Crowding Distance (*WSCD*) as it is calculated as the weighted sum of the crowding distances in objective and decision space. The WSCD method is shown in Algorithm 1. In WSCD, the calculation of crowding distance in the objective space is similar to the proposed CD calculations in NSGA-II. The extreme solutions in the objective space are assigned a large CD values (infinity). The CD values for the rest of the solutions are calculated by the sum of the normalized distances between the left-side and the right-side neighbors in the objective space [1].

In the proposed WSCD approach, first the calculation of the crowding distance in decision space is adopted from the Omni-optimizer from the literature (Lines 1 to 16). The maximum and minimum values for all solutions are calculated (Lines 2 and 3). The crowding distance values in decision space and the WSCD values for all solutions are first set to zero (Lines 6 and 7). Then the solutions are sorted based on the decision variable values for each variable (Line 10). The crowding distance value for the boundary solutions are calculated from the normalized distance values between the solution and its adjacent neighbors (Lines 11 and 12). The crowding distance values for the rest of the solutions are calculated by normalizing the distances between the left-side and right side neighbors for the solutions in decision space (Line 14). The novelty of our work is as follows: The crowding distance values in decision

Algorithm 1: Weighted Sum Crowding Distance approach.

```
   Input: List $S$ of non-dominated solutions with added Crowding Distance ($CD_{obj}$) values for
          each solution in objective space according to NSGA-II [1] algorithm with $s := |S|$,
   Number of Objectives: $M$,
   Number of Decision Variables
 1 : $D$
   Output: List $S$ with added Weighted Sum Crowding Distance ($CD_{WS}$) values for each
           solution
 2 for $i \in \{1, .., D\}$ do
 3     $x_{i,max}$ = maximum of values for $i$-th decision variable in $S$
 4     $x_{i,min}$ = minimum of values for $i$-th decision variable in $S$
 5 end
 6 for $j \in \{1, .., s\}$ do
 7     $S[j].CD_{dec} = 0$ //initialize $CD_{dec}$ of $j$-th solution in $S$
 8     $S[j].CD_{WS} = 0$ //initialize $CD_{WS}$ of $j$-th solution in $S$
 9 end
10 for $i \in \{1, .., D\}$ do
11     $S'$ = sort $S$ ascending based on $i$-th decision variable
12     $S'[1].CD_{dec} += 2 \cdot \frac{|S'[j+1].x_i - S'[j].x_i|}{|x_{i,max} - x_{i,min}|}$
13     $S'[s].CD_{dec} += 2 \cdot \frac{|S'[j].x_i - S'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
14     for $j \in \{2, .., s-1\}$ do
15         $S'[j].CD_{dec} += \frac{|S'[j+1].x_i - S'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
16     end
17 end
18 for $j \in \{1, .., s\}$ do
19     $S[j].CD.obj = norm(S[j].CD_{obj})$ //normalize $CD_{obj}$ of $j$-th solution using max and min
       values of $CD_{obj}$ in $S$
20     $S[j].CD.dec = norm(S[j].CD_{dec})$ //normalize $CD_{dec}$ of $j$-th solution using max and min
       values of $CD_{dec}$ in $S$
21     $S[j].CD_{WS} = w_1 \cdot S[j].CD_{dec} + w_2 \cdot S[j].CD_{obj}$;
22 end
23 return $S$
```

and objective spaces are normalized in order to make the scores of crowding distance values comparable for different dimensions in decision and objective space (Lines 18 and 19). Given the importance of having a good diversity of solutions in both decision and objective spaces, we allocate a final weighted sum crowding distance value based on the assigned weights w_1 and w_2 for the crowding distance in the decision and the objective spaces (Line 21).

8.3.2 Neighborhood Polynomial Mutation

In multi-objective evolutionary algorithms, the Polynomial mutation operator is often used and shown to be effective [12]. It was originally proposed by Deb and Goyal [13].

In this section, we propose a modification to this operator inspired by the concept of neighborhood mutation by Qu et al. [14] to make it more applicable on multimodal optimization problems. The neighborhood-based Polynomial mutation is presented in Algorithm 2. In this algorithm, a set of neighbors is computed for each solution, and the mutation operator is applied to each of them.

In Algorithm 2, at first the Euclidean distances between all solutions in the decision space are computed (Line 3). The neighborhood of each solution is composed out of the individual itself and its K nearest neighbors in terms of computed distances (Line 7). Afterwards, for each individual in the population, a Polynomial mutation is used to mutate the individual and its neighbors (Lines 9 to 26). The mutated offsprings are

Algorithm 2: Neighborhood Polynomial Mutation.

```
   Input: List $O$ of offspring of solutions of current generation with $o := |O|$,
   Neighborhood Size=$K$
   Probability of Mutation=$p_m$,
   Distribution Index=$\eta_m$
   Upper and lower bounds $x_k^u$ and $x_k^l$ for each variable $k$
   Output: Mutated Individuals $O$
 1 for $i \in \{1, .., o\}$ do
 2     for $j \in \{1, .., o\}$ do
 3         $Euc(i, j) = \|O[i].\mathbf{x} - O[j].\mathbf{x}\|_2$ //calculate Euclidean distances between solutions
 4     end
 5 end
 6 for $i \in \{1, .., o\}$ do
 7     $N(i)$ = list of indices of $K + 1$ smallest values in $Euc(i)$ //Set the neighborhood of each
       solution $i$ as itself and its $K$ nearest neighbors
 8 end
 9 for $i \in \{1, .., o\}$ do
10     for $j \in N(i)$ do
11         for $k \in \{1, .., D\}$ do
12             $b = U(0, 1)$
13             if $b \leq p_m$ then
14                 $\delta_1 = \frac{O[j].x_k - x_k^l}{x_k^u - x_k^l}$
15                 $\delta_2 = \frac{x_k^u - O[j].x_k}{x_k^u - x_k^l}$
16                 $b = U(0, 1)$
17                 if $b \leq 1/2$ then
18                     $\delta_q = [(2b) + (1 - 2b)(1 - \delta_1)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} - 1$
19                 else
20                     $\delta_q = [1 - (2(1 - b)) + 2(b - 0.5)(1 - \delta_2)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}}$
21                 end
22                 $O[j].x_k += \delta_q.(x_k^u - x_k^l)$
23             end
24         end
25     end
26 end
27 return $O$
```

returned (Line 27). Using this mutation operator implies that a neighboring solution which appears in the neighborhood of many solutions, has the chance to be mutated more often than other solutions. In that way, the solutions which are located in crowded areas in the search space have a higher chance of being mutated. As a result, this might lead to a better exploration in the decision space.

8.4 Experimental Setting

In order to evaluate the effectiveness of the modifications, we considered various versions of the proposed algorithm. The NSGA-II with the Neighbourhood-based Mutation operator (NSGA-II-NBM), the NSGA-II with the Weighted Sum Crowding Distance (NSGA-II-WSCD), and NSGA-II with both of the modifications (NSGA-II-WSCD-NBM). The results are compared with the results of the state-of-the-art multimodal optimization algorithm Mo-Ring-PSO-SCD [6]. We additionally compare the results with NSGA-II [1] as the baseline. The median and the interquartile range (IQR) of all the experimental results are calculated over 31 independent runs for a maximum of 10,000 function evaluations. The population size is set to 100 for all the experiments. The parameters of NSGA-II are set to be similar as in the literature [1]. We set the distribution index of both crossover η_c and mutation η_m to be 20. The probability of crossover is set to $p_c = 1.0$, and the probability of mutation is set to $p_m = 1/D$, where D is the number of decision variables. The neighborhood size for the neighborhood-based mutation in both the NSGA-II-WSCD-NBM and NSGA-II-NBM is set to 20. In both WSCD variations, NSGA-II-WSCD-NBM and NSGA-II-WSCD, the weights are equally divided for crowding distances in decision and objective spaces as $w_1 = 0.5$ and $w_2 = 0.5$. In the Mo-Ring-PSO-SCD, we use the same parameter values as in the literature [6]. Therefore, we set $C_1 = C_2 = 2.05$ and $W = 0.7298$. We used codes provided in Matlab-based PlatEmo [15] framework for the NSGA-II and the codes by the original authors for Mo-Ring-PSO-SCD [6].

8.4.1 Test Problems

We take the state-of-the-art test problems for multimodal multi-objective optimization [2, 6] to test our proposed algorithms. We use the SSUF1 and SSUF3 test problems [2] and MMF3, MMF4, MMF5,and MMF6 problems [6]. The problems contain different levels of complexity and different numbers of equivalent subsets of the PS to challenge the functionality of the proposed algorithms. The dimensions of decision and objective spaces are 2 in all of the problems. Since the problems are multimodal, one of the most important features of these test problems is that there are always multiple distinct subsets of the PS in each problem, where each of them covers the PF completely on its own.

8.4.2 Performance Measures

Since our primary focus lies in decision space, the Inverted Generational Distance in decision space (IGDX) [9] is adopted as a metric to measure the effectiveness of the algorithms. The IGDX performance metric is calculated as the average Euclidean distance between the set of obtained solutions and the PS in decision space. This metric demonstrates the diversity and convergence of obtained solutions in relation to the Pareto-optimal solutions set. A lower IGDX value indicates a better performance. Let P^* be a sample of the PS of the problem, and R a set of obtained solutions in decision space by an algorithm, the IGDX indicator is formulated as:

$$IGDX\,(P^*, R) = \frac{\sum_{v \in P^*} \|R - v\|_2}{|P^*|} \tag{8.1}$$

Where $\|R - v\|_2$ is the minimum Euclidean distance between the sampled point v and any point in R.

In addition, we calculated the Pareto Set Proximity (PSP) [6] performance indicator to also represent the overlap ratio between the obtained solution set and PS. This indicator is calculated by the division of the Cover Rate (CV) and the IGDX value $PSP = CR/IGDC$.

In this formula the CR value represent the maximum spread of obtained solutions in decision space. A higher CR value shows a better overlap ratio between the bounding box of the obtained set and the PS.

Additionally, in order to compare the performance of the algorithms with each other in the objective space, we use the Inverted Generational Distance (IGD) [16, 17]:

Table 8.1 PSP values of different algorithms. An asterisk (*) indicates statistical significance compared to the respective best algorithm

	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
SSUF1	**15.70245 (5.23527E–1)**	13.14347 (1.59755)*	12.50492 (1.24596)*	13.50486 (1.37166)*	9.29939 (1.41609)*
SSUF3	**59.21458 (9.79915)**	56.24433 (12.59192)	9.36013 (7.80405)*	30.63185 (9.39582)*	9.42434 (8.32393)*
MMF3	**67.17031 (5.65764)**	66.58847 (9.97048)	14.95064 (11.64698)*	38.33345 (9.46867)*	12.73248(0.0314)*
MMF4	**23.97306 (2.88038)**	16.89303 (3.19544)*	17.15216 (3.1857)*	21.85007 (2.94561)*	8.20806 (3.14241)*
MMF5	**8.72255 (3.8104E–1)**	7.66519 (9.6644E–1)*	6.8738 (5.9585–1)*	7.95066 (5.7642E–1)*	4.89867 (1.33815)*
MMF6	**10.03783 (6.3865E–1)**	9.20888 (9.6738E–1)	7.92392 (8.2706)*	9.24543 (8.2126E–1)*	5.14445 (8.2706E–1)*

$$IGD(P^*, R) = \frac{\sum_{v \in P^*} \|\mathbf{f}(R) - \mathbf{f}(v)\|_2}{|P^*|} \tag{8.2}$$

This indicator is formulated in the same way as the IGDX. The IGD value is calculated, with the difference that the distances are calculated in the objective space using a sample of the PF (which can be obtained by evaluating the PS as $\mathbf{f}(P^*)$) and $\mathbf{f}(R)$ accordingly. All experiments were run using MATLAB R2018a on a PC equipped with an Intel Core i7 CPU with 3 GHz, a 64-bit Operating System and 16 GB of RAM.

8.5 Analysis of Results

The experimental results (median and IQR) for the comparison of the used algorithms concerning IGDX, IGD and PSP indicators are shown in Tables 8.1, 8.2 and 8.3 respectively. Smaller IGDX and IGD values and larger PSP values indicate better performance. In order to test the statistical significance, we take the Mann-Whitney U statistical test with respect to the best algorithm on each test problem. That is, we test for each algorithm the hypothesis that the performance of the algorithm and the performance of the best algorithm on this problem have equal medians. A difference between the two results is regarded as significant for values of $p < 0.01$. The best values are highlighted in bold and significance compared to the best algorithm is shown by an asterisk (*) in the respective columns.

From the analysis of Tables 8.1 and 8.2 regarding the comparison of IGDX and PSP values, it can be concluded that the NSGA-II-WSCD-NBM algorithm outper-

Table 8.2 IGDX values of different algorithms. An asterisk (*) indicates statistical significance compared to the respective best algorithm

	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
SSUF1	**0.06321 (1.817E–3)**	0.0.07552 (9.316E–3)*	0.07923 (7.412E–3)*	0.0.07235 (7.26E–3)*	0.1051 (151E–2)*
SSUF3	**0.01688 (2.885E–3)**	0.01771 (3.956E–3)	0.08949 (7.2725E–2)*	0.0.03088 (8.32E–3)*	0.1021 (853E–2)*
MMF3	**0.01486 (1.309E–3)**	0.015017 (2.318E–3)	0.0.05839 (3.49894E–2)*	0.02478 (5.73E–2)*	0.07854(314E–2)*
MMF4	**0.04163 (4.949E–3)**	0.05895 (1.0412E–2)*	0.05793 (1.0978E–2)*	0.04493 (5.78E–3)*	0.11921 (4185E–1)*
MMF5	**0.11394 (4.388E–3)**	0.12952 (1.7379E–2)*	0.14473 (1.1124E–2)*	0.12442 (8.72E–3)*	0.19475 (3932E–2)*
MMF6	**0.09921 (6.021E–3)**	0.10812 (1.054E–2)*	0.12406 (1.2611E–2)*	0.10665 (9.2E–3)*	0.18852 (6103E–2)*

Table 8.3 IGD values of different algorithms. An asterisk (*) indicates statistical significance compared to the respective best algorithm

	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
SSUF1	5.53E–3 (6.2E–4)*	**4.6E–3 (1.2E–4)**	5.441E–3 (3.22E–4)*	6.49E–3 (7.6e–4)*	5.32E–3 (2.6E–4)*
SSUF3	1.455E–2 (2.516E–3)	**1.4452E-2 (2.497E–3)**	1.6955E–2 (1.4602E–2)*	1.877E–2 (5.89E–3)*	1.995E–2 (1.253E–2)
MMF3	1.2298 E–2 (2.12E–3)*	**1.098E-2 (2.007E–3)**	1.527E–2 (1.3291E–2)*	1,656E–2 (0.00485)*	1.497E–2 (9.72E–3)*
MMF4	5.347E–3 (7.6E–4)*	**4.762E-2 (2.4E–4)**	5.425E–3 (2.5E–4)*	7.02E–3 (9.2E–4)*	5.17E–3 (1.9E–4)*
MMF5	5.37E–3(3.9E–4)*	**4.6E-3 (1.7E–4)**	5.59E–3 (3.2E–4)*	6.52E–3(5.3E–4)*	5.34E–3 (3.2E–4)*
MMF6	5.43E–3 (4.57E–4)	**4.59 E-3 (2.01E–4)**	5.49E–3 (2.83E–4)*	6.43E–3 (7.5E–4)*	5.31E–3 (2.6E–4)*

forms the NSGA-II-NBM in four out of six test problems. It also shows its significant superiority for all the test problems compared with the results of the other algorithms. This means the proposed algorithm provides better approximations of PS in terms of the both diversity and convergence of the obtained solutions.

To analyze the performance in the objective space, Table 8.3 shows the IGD values for the different algorithms. As can be observed from the results, NSGA-II-NBM obtains a better IGD value than the others, while both the NSGA-II and NSGA-II-WSCD-NBM algorithms gained IGD values similar to each other. We can further observe that the proposed methods significantly outperform the original NSGA-II and the state-of-the-art Mo-Ring-PSO-SCD. In terms of IGDX, the proposed NSGA-II-WSCD-NBM performs significantly better than both algorithms from the literature on all of the six test problems. In the objective space, measured by the IGD indicator, NSGA-II-NBM outperforms the state-of-the-art in all of the used benchmarks, and the original NSGA-II on all but one test problem. According to the analysis of the results, the WSCD variants lead to preserving distinct solutions with the same objective function values. Therefore the NSGA-II-WSCD shows improvement compared to NSGA-II in terms of the decision space related metric. In addition, introducing neighborhood-based mutation helps to discover more Pareto-optimal solutions during the search by increasing the diversity of solutions.

In order to better understand the similarity between the obtained solutions in both decision and objective spaces, we present the obtained solutions for the NSGA-II-WSCD-NBM, NSGA-II-WSCD, NSGA-II-NBM and Mo-Ring-PSO-SCD in Figs. 8.1 and 8.2. The figures show the runs which achieved the median IGDX indicator for each of the algorithms. As an example, in Figs. 8.2 we illustrate the obtained solutions in the decision space for the MMF3 problem of the algorithms. The same is shown for the objective space. We can observe that all algorithms obtain an evenly spread solution set al.ong the PF in the objective space. However, when we look at the decision space we see differences. As can be seen from the Figs. 8.1 and 8.2, the obtained solutions in decision space for NSGA-II-WSCD-NBM are evenly distributed along the PS while covering more points in each of the subsets of the PS. This is because both the NBM and WSCD methods could help the algorithm locate and maintain the captured optimal solutions in decision space in each generation.

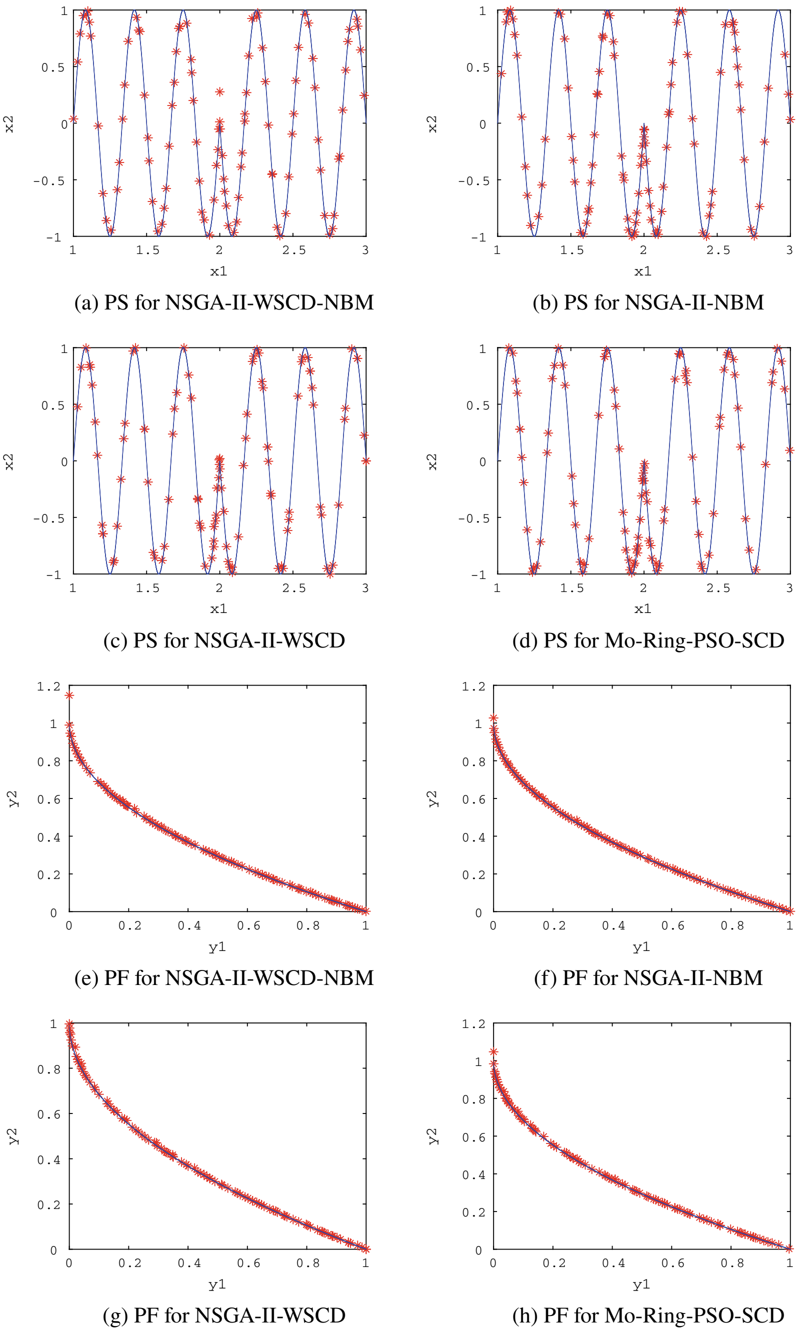
In NSGA-II-NBM, the obtained solutions are mostly located in one of the subsets. This means that this algorithm could not preserve the solutions in different subsets, since the crowding distance is only used in objective space. While the solutions in decision space are distributed in all the equivalent subsets of the PS in the NSGA-II-WSCD algorithm, we still lack an even spread along these subsets (Figs. 8.1c and 8.2c). Altogether, we conclude that NSGA-II-WSCD which uses the crowding distance in the decision space helps to maintain most of the so far found solutions. However, due to a lack of neighborhood mutation process, it could not find all the solutions of the PS. The results of the Mo-Ring-NSGA-II shown in Figs. 8.1 and 8.2 also reveal that the PS could not be fully covered by the algorithm and the solutions are not evenly distributed along the PS.

8.5.1 Influence of the Weight Values in WSCD

In addition to the overall performance of the algorithms, we investigate the impact of the weights (w) in the WSCD variants. Our preliminary studies show that increasing the weight value in either the objective or the decision space improves the distribution of solutions in the corresponding space, while deteriorating the distribution of solutions in the other space. The results of different w values on the performance of NSGA-II-WSCD are demonstrated in Fig. 8.3. The horizontal axis shows the different weight values used for the crowding distance in decision space (from 1 to 0). The vertical axis shows the IGDX, IGD and PSP values obtained by different weights. We see in Figs. 8.3a and 8.3b that the performance of the NSGA-II-WSCD algorithm is sensitive to the weight vector values on multimodal multi-objective test problems. However, as we expected, decreasing the share of the value of crowding distance in decision space, the performances of the NSGA-II-WSCD algorithm on approximating the PS deteriorates. On the other hand, we obtain a better approximation of PF by increasing the weight value in objective space. The obtained PSP values in Fig. 8.3c also support the idea that with decreasing the portion of crowding distance in decision space the diversity of approximation of obtained PS are deteriorated.

8.5.2 Influence of the Population Size in WSCD

Finally we examine the impact of the population size on the performance of the NSGA-II-WSCD-NBM algorithm. In most algorithms, increasing the population



(a) PS for NSGA-II-WSCD-NBM

(b) PS for NSGA-II-NBM

(c) PS for NSGA-II-WSCD

(d) PS for Mo-Ring-PSO-SCD

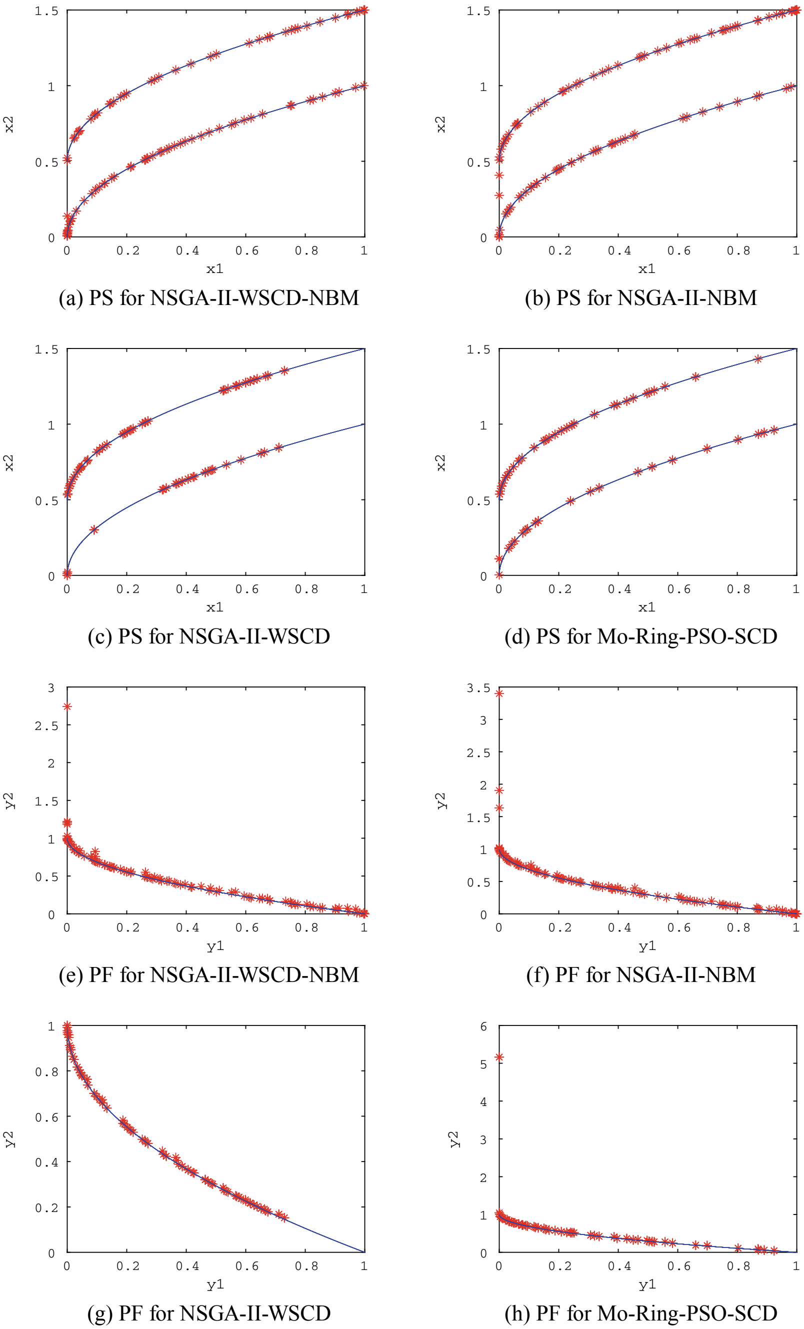
(e) PF for NSGA-II-WSCD-NBM

(f) PF for NSGA-II-NBM

(g) PF for NSGA-II-WSCD

(h) PF for Mo-Ring-PSO-SCD

Fig. 8.1 Obtained solutions in decision and objective space for SSUF1 problem



(a) PS for NSGA-II-WSCD-NBM

(b) PS for NSGA-II-NBM

(c) PS for NSGA-II-WSCD

(d) PS for Mo-Ring-PSO-SCD

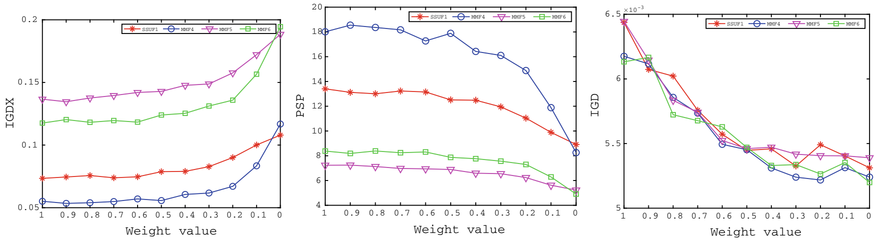
(e) PF for NSGA-II-WSCD-NBM

(f) PF for NSGA-II-NBM

(g) PF for NSGA-II-WSCD

(h) PF for Mo-Ring-PSO-SCD

Fig. 8.2 Obtained solutions in decision and objective space for MMF3 problem

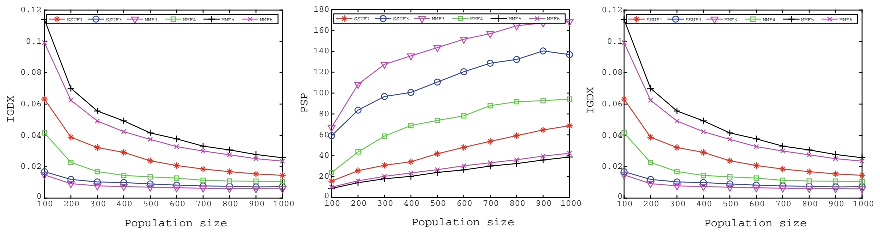


(a) IGDX values (b) PSP values (c) IGD values

Fig. 8.3 Achieved **a** IGDX values, **b** PSP values and **c** IGD values by NSGA-II-WSCD using different weight values for the crowding distance in the decision space

size results in better approximations of optimal solutions [18]. However, increasing the population size also comes with an increase in computational costs. We preformed experiments with different population sizes for the NSGA-II-WSCD-NBM algorithm on the six different test problems to evaluate the effects of the population size on the approximation of the PS and PF. The number of function evaluations is set to 10, 000 and the used population sizes are 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000. In order to make the trade off between both approximation of optimal solutions in decision and objective spaces we equally divided the weight values on both decision and objective spaces.

The results of these experiments are shown in Fig. 8.4, where the median IGDX, PSP and IGD values of the experiments based on 31 independent runs are shown on the vertical axis. As expected, we observe in Fig. 8.4a, b that larger population sizes lead to a higher probability of locating more diverse solutions. Therefore, the algorithm provides better approximations of the PS with larger population sizes. However, as we can observe in Fig. 8.4c the rate of improvement regarding the IGD indicator decreases for larger population sizes. This shows that locating more solutions in decision space does not guarantee well-distributed solutions in objective space.



(a) IGDX values (b) PSP values (c) IGD values

Fig. 8.4 Achieved **a** IGDX values, **b** PSP values and **c** IGD values by NSGA-II-WSCD using different population sizes

8.6 Conclusion

The purpose of the current study is to propose two mechanisms for acquiring better approximations of the PS in multimodal multi-objective problems. These two mechanisms are (1) the WSCD method which combines crowding distance in both objective and decision spaces, and (2) a neighborhood Polynomial mutation. The two proposed operators were included in different combinations into the existing NSGA-II algorithm. In order to examine the performance of the presented combinations of operators, we compare the algorithms with the original NSGA-II algorithm as well as a state-of-the-art multimodal algorithm from the literature (Mo-Ring-PSO-SCD) on six different test problems. The IGDX, PSP and IGD performance indicators are used to compare the performance of the algorithms in decision and objective spaces. The results show significant differences between the proposed variants of the NSGA-II-WSCD-NBM, NSGA-II-WSCD and NSGA-II-NBM algorithms compared to the existing methods in terms of approximations of the PS and PF. The proposed algorithm NSGA-II-WSCD-NBM is able to outperform the state-of-the-art Mo-Ring-PSO-SCD and the standard NSGA-II on all of the test problems in terms of approximating the PS, while at the same time obtaining comparable IGD values. For future work, we want to compare the proposed NSGA-II-WSCD-NBM with other state-of-the-art multimodal algorithms like those recently proposed in the literature [5, 7, 19]. In addition, we will evaluate the potential of the proposed algorithm on solving more complex test problems and real world examples.

References

1. Deb K, Pratap A (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
2. Liang J, Yue C, Qu B (2016) Multimodal multi-objective optimization: a preliminary study. In: Evolutionary Computation (CEC). IEEE, Vancouver, pp 2454–2461
3. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8:256–279
4. Javadi M, Zille H, Mostaghikm S (2016) Modified crowding distance and mutation for multimodalmulti-objective optimization. ACM, Prague, To appear
5. Liang J, Xu W, Yue C, Yu K, Song H et al (2019) Multimodal multiobjective optimization with differential evolution. Swarm Evol Comput 44:1028–1059
6. Yue C, Qu B, Liang, J (2018) A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Trans Evol Comput 22:805–817
7. Liang J, Guo Q, Yue C, Qu B, Yu K (2018) A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In: Broy M, Dener E (eds) Software pioneers. Springer, Shanghai, pp 550–560
8. Deb K, Tiwari S (2005) Omni-optimizer: a procedure for single and multi-objective optimizationn. In: International conference on evolutionary multi-criterion optimization. Springer, Guanajuato, pp 47–61
9. Zhou A, Zhang Q, Jin Y (2009) Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. IEEE Trans Evol Comput 13:1167–1189

10. Liu Y, Ishibuchi H, Nojima Y, Masuyama N, Shang K (2018) A double-niched evolutionary algorithm and its behavior on polygon-based problems. In: International conference on swarm intelligence. Springer, Coimbra, pp 262–273
11. Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello CA et al (2009) Smpso: a new pso-based metaheuristic for multi-objective optimization. In: Computational intelligence in multi-criteria decision-making. IEEE, Nashville, pp 66–73
12. Hamdan M (2012) The distribution index in polynomial mutation for evolutionary multiobjective optimisation algorithms: An experimental study. In: Broy M, Dener E (eds) Software pioneers. IEEE, Kanyakumari
13. Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inf 26:30–45
14. Qu BY, Suganthan PN, Liang J (2012) Differential evolution with neighborhood mutation for multimodal optimization. IEEE Trans Evol Comput 16:601–614
15. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12:73–87
16. Reyes-Sierra M, Coello CA (2005) A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization: an experimental study. In: International conference on electronics computer technology. IEEE, Edinburgh, pp 65–72
17. Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. IEEE Trans Evol Comput 12:41–63
18. Hu Y, Wang J, Liang J, Yu K et al (2019) A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm. Sci China Inf Sci 62:70206
19. Tanabe R, Ishibuchi H (2018) A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. In: International conference on parallel problem solving from nature. Springer, Coimbra, pp 249–261

Chapter 9
Combining Manhattan and Crowding Distances in Decision Space for Multimodal Multi-objective Optimization Problems

Mahrokh Javadi, Cristian Ramirez-Atencia, and Sanaz Mostaghim

Abstract This paper presents a new variant of the Non-dominated Sorting Genetic Algorithm to solve Multimodal Multi-objective optimization problems. We introduce a novel method to augment the diversity of solutions in decision space by combining the Manhattan and crowding distance. In our experiments, we use six test problems with different levels of complexity to examine the performance of our proposed algorithm. The results are compared with NSGA-II and NSGA-II-WSCD algorithms. Using IGDX and IGD performance indicators, we demonstrate the superiority of our proposed method over the rest of competitors to provide a better approximation of the Pareto Set (PS) while not getting much worse results in objective space.

Keywords Multimodality · Multi-modal problems · Multi-objective optimization · Evolutionary algorithms · Solution space diversity

9.1 Introduction

In real world, there are many Multi-objective Optimization Problems (MOP) with at least two conflicting objectives in nature. This means that improving one of the objectives leads to deteriorating the value for the other objectives. Without loss of generality, a multi-objective minimization problems is formulated as follows:

M. Javadi (✉) · C. Ramirez-Atencia · S. Mostaghim
Faculty of Computer Science, Otto von Guericke University, Magdeburg, Germany
e-mail: mahrokh1.javadi@ovgu.de

C. Ramirez-Atencia
e-mail: cristian.ramirez@ovgu.de

S. Mostaghim
e-mail: sanaz.mostaghim@ovgu.de

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_9

$$\begin{aligned} \text{minimize } & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\ \text{subject to } & \quad \mathbf{x} \in S \subset \mathbb{R}^D \\ & g_i(\mathbf{x}) \leq 0, i = 1, 2, \ldots, G \\ & h_j(\mathbf{x}) = 0, j = 1, 2, \ldots, H \end{aligned} \tag{9.1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ is considered as a D–dimensional decision vector and $(f_1, f_2, \ldots, f_M)$ is a M–dimensional objective vector. $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are inequality and equality constraint functions in decision space.

In order to deal with these problems, the concept of domination can be used. Given two vectors $\mathbf{x}, \mathbf{y} \in S$, $\mathbf{x}$ is said to be dominated by $\mathbf{y}$ (denoted by $\mathbf{y} \prec \mathbf{x}$) if and only if $\forall j \in \{1, \ldots, M\}, f_j(\mathbf{y}) \leq f_j(\mathbf{x})$, and $\exists k \in \{1, \ldots, M\}, f_k(\mathbf{y}) < f_k(\mathbf{x})$ [1].

The solution of multi-objective optimization problems, is a set of non-dominated solutions called Pareto-Set (PS), which the corresponding set of these solutions in the objective space is called the Pareto-Front (PF).

In Multimodal Multi-Objective Optimization Problems, there are two or more distinct solutions in the PS, which correspond to the same value in the PF. In this area, most of the available literature deals with multimodal single objective optimization problems and there is a relatively small number of published research on Multimodal Multi-objective Optimization Problems (MMOP) [2]. In the current paper, we propose a new method for this type of problems, which is based on the combination of the Manhattan Distance and Crowding Distance in decision space (MDCD). The performance of our proposed method is examined on a number of available multimodal multi-objective test functions. We study the influence of the proposed method on finding a better approximation of optimal solutions in decision space, and the results are compared with the state-of-the-art algorithms.

The remaining parts of the paper proceed as follows: In Sect. 9.2 the related works on MMOPs are investigated. The proposed algorithm is presented in Sect. 9.3. In Sect. 9.4, the setting of the experiments is explained. The experiments and analysis are presented in Sect. 9.5. In the end, Sect. 9.6 concludes the paper and provides the future research direction.

9.2 Related Work

In the field of Evolutionary Multi-objective Optimization (EMO), the main concern is to find a good approximation of PF with a good diversity of solutions in objective space [3]. However, there is not much literature focusing on increasing the diversity of solutions in the decision space to handle MMOPs.

One of the first works dealing with MMOPs was proposed by Deb and Tiwari [4] who introduce the omni-optimizer algorithm. This algorithm is a modified version of the well-known *Non-dominated Sorting Genetic Algorithm-II* (NSGA-II) [5]. The aim of this work was dealing with a wider range of optimization problems (i.e: single or multi-objective and uni or multimodal problems). They proposed a modified crowding distance by comparing the crowding distance value of each individual with its average value (in both spaces), and take the larger value of the two distances.

To achieve a better distribution of solutions both in decision and objective spaces, Zhou et al. [6] proposed a model where the population is classified into sub-populations in the objective space, and the diversity of solutions is increased in the decision space by evaluating the diversity of PS in each sub-population. The obtained solutions show a better convergence to PS and PF for the MOP compared to the Omni-optimizer algorithm.

The concept of niching in MMOPs is used by Liang et al. [1]. They proposed *Decision-based Niching NSGA-II* (DN-NSGA-II), where they applied the crowding distance technique to the decision space instead of the objective space as a secondary selection criteria. Even though this algorithm could find more Pareto optimal solutions than NSGA-II, the solutions are not well distributed in decision space.

Another perspective is found in an algorithm called *Multi-objective Particle Swarm Optimization using Ring topology by applying Special Crowding Distance* (MO-Ring-PSO-SCD) proposed by Yue et al. [7]. They used a ring topology and a special crowding distance method to locate and maintain more Pareto optimal solutions. This algorithm is able to provide better approximation of PS in comparison with NSGA-II, DN-NSGA-II and Omni-optimizer algorithms.

Multimodal Multi-Objective Differential Evolution algorithm (MMODE) was proposed by Liang et al. [8]. The mutation-bound process was introduced to provide a second opportunity to perform mutation for infeasible solutions (those outside the boundaries) of the decision space. In their presented algorithm, the crowding distance method is applied to the solutions in the decision space to maintain the diversity of solutions.

In a recent study, another contribution is proposed by Liu et al. [2], called *Double-Niches Evolutionary Algorithm* (DNEA). The main focus of this method is the calculation of Euclidean distance in both decision and objective spaces. Then, a double-niched method is applied to diversify the solutions on both decision and objective spaces.

In a previous work, we proposed a modified version of NSGA-II algorithm called Weighted Sum Crowding Distance using NSGA-II algorithm (NSGA-II-WSCD) [9]. To obtain a good diversity of solutions both in the decision and objective spaces, we compute the Crowding distance value of solutions by taking the weighted sum value of crowding distances in both spaces.

9.2.1 Crowding Distance in the NSGA-II Algorithm

The NSGA-II is a population-based algorithm that was introduced by Deb et al. in 2002 [5], and was described as one of the most popular multi-objective algorithms by a study in 2011 [10].

This algorithm provides a selection process consisting in two steps: first, the sorting of the population through a fast non-dominated sorting method; second, for each front obtained in the previous step, the crowding distance method is applied in the objective space in order to decide which solutions provide a better diversity. The algorithm keeps the solutions with lower rank and higher crowding distance in successive generations. The maintenance of diversity in crowding distance is based on the selection of solutions in less crowded areas in the objective space. The crowding distance method used in NSGA-II is presented in Algorithm 1.

In this method, the first step consist on computing the maximum and minimum values for each objective (Lines 2 and 3) among all the solutions of the front. Then, the crowding distance values are initialized as zero for every solution (Line 6). Following that, for each objective function, solutions are sorted according to their fitness values in that objective function (Line 9), and the first and last individuals (i.e. the extreme points) are assigned a crowding distance value of infinity (Lines 10 and 11), in order to keep them for next generation, as they preserve the spread of the front. Then, for the rest of solutions of the front, the normalized distance between the left-side and right-side neighbors in that objective function is added to the crowding distance in each solution (Line 13).

Algorithm 1: Crowding Distance method used in NSGA-II algorithm.

```
   Input: List $P$ of non-dominated solutions with $p := |P|$,
   Number of Objectives $M$
   Output: List $P$ with added Crowding Distance ($CD$) values for each solution
1  for $i \in \{1, .., M\}$ do
2      $f_{i,max}$ = maximum of values for $i$-th objective in $P$
3      $f_{i,min}$ = minimum of values for $i$-th objective in $P$
4  end
5  for $j \in \{1, .., p\}$ do
6      $P[j].CD = 0$ //initialize $CD$ of $j$-th solution in $P$
7  end
8  for $i \in \{1, .., M\}$ do
9      $P'$ = sort $P$ ascending based on $i$-th objective
10     $P'[1].CD = \infty$
11     $P'[p].CD = \infty$
12     for $j \in \{2, .., p-1\}$ do
13         $P'[j].CD += \frac{P'[j+1].f_i - P'[j-1].f_i}{f_{i,max} - f_{i,min}}$
14     end
15 end
16 return $P$
```

9.3 Proposed NSGA-II-MDCD Algorithm

In this section we propose a modified distance measurement technique that can be used to obtain a better diversity of solutions in decision space, and therefore make a better approximation of PS.

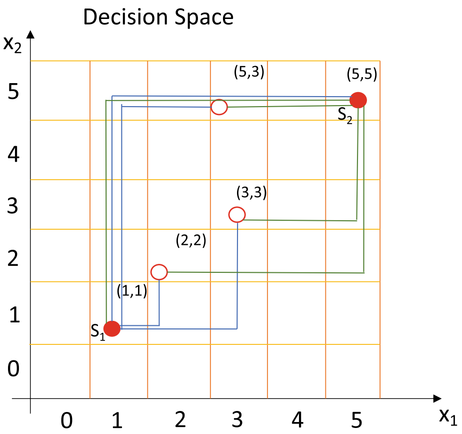


Fig. 9.1 An example of the computation of MDCD, and its influence on the diversity of solutions in the decision space

In the proposed method, due to the natural capability of grids to represent the distribution of solutions, we took the Manhattan distance metric (also called p_1 metric) as a distance measurement method in the decision space. For each solution, we calculate the Manhattan distance to all other solutions in the current front. Then, our global Manhattan distance metric is computed as the summation of all of these distances between each solution and the rest of solutions:

$$MD_{global}(\mathbf{a}) = \sum_{p \in P} \|\mathbf{a} - \mathbf{p}\| = \sum_{p \in P} \sum_{i=1}^{D} |a_i - pi| \qquad (9.2)$$

where P is the current front of solutions, D is the dimension of decision variables, and a_i and p_i represent the grid index values of solutions **a** and **p** in dimension i.

For a better diversity of solutions, we multiply the obtained global Manhattan distance metric value with its crowding distance value in decision space (as defined in [4], this distance only takes into account its nearest neighbor for boundaries).

In Fig. 9.1, an example is used to better illustrate the influence of both Manhattan and crowding distance on obtaining a good diversity of solutions in decision space.

In Fig. 9.1, the global Manhattan distance values of S_1 and S_2 are both equal to 20. Both of the solutions are located far from the rest of the solutions and both make a good coverage of solutions in decision space. In this example, solution S_1 is located in a more crowded neighborhood area than S_2. Therefore, the crowding distance value for S_1 is smaller than for the other solutions. By multiplying both Manhattan and crowding distance values, S_2 gets a larger value than the other solutions. Therefore, we could guarantee a better diversity of solutions by using both distance metrics.

In Algorithm 2 we present our proposed method (NSGA-II-MDCD). We modify NSGA-II by changing the crowding distance with our MDCD metric. First we calculate the global Manhattan distance value (Lines 1 to 12). Then, the crowding distance value for all solutions in the decision space is calculated (Lines 8 to 21). The Final *MDCD* value for each solution is calculated by multiplying the two distances.

Algorithm 2: Combining Manhattan Distance and Crowding Distance (MDCD) approach.

```
Input: Number of Objective functions: $M$,
Number of Decision variables : $D$,
List $P$ of solutions of current front (with GridIndex values for each dimension), of size
$p = |P|$
Output: List $P$ of solutions of current front with extra property of Combined Manhattan
        Distance and Crowding Distance (MDCD) for each solution
for $j \in \{1, .., p\}$ do
    $P[j].MD_{global} = 0$
    $P[j].CD_{dec} = 0$
    $P[j].MDCD = 0$
end
for $i \in \{1, .., p\}$ do
    for $j \in \{1, .., p\}$ do
        for $k \in \{1, .., D\}$ do
            $P[i].MD_{global}+ = P[i].GridIndex(k) - P[j].GridIndex(k))$
        end
    end
end
for $i \in \{1, .., D\}$ do
    $x_{i,min}$ = minimum of values for $i$-th decision variable in $P$
    $x_{i,max}$ = maximum of values for $i$-th decision variable in $P$
end
for $i \in \{1, .., D\}$ do
    $P'$ = sort $P$ ascending based on $i$-th decision variable
    $P'[1].CD_{dec} += 2 \cdot \frac{|P'[j+1].x_i - P'[j].x_i|}{|x_{i,max} - x_{i,min}|}$
    $P'[p].CD_{dec} += 2 \cdot \frac{|P'[j].x_i - P'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
    for $j \in \{2, .., p-1\}$ do
        $P'[j].CD_{dec} += \frac{|P'[j+1].x_i - P'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
    end
end
for $i \in \{1, .., p\}$ do
    $P[j].MDCD = P[j].CD_{dec} \cdot P[j].MD_{global}$
end
return $P$
```

9.4 Experiments

This section is dedicated to investigate the effectiveness of the proposed method (NSGA-II-MDCD) for obtaining a good approximation of solutions in both decision and objective spaces. We compare the proposed approach with the state-of-the-art, NSGA-II-WSCD algorithm and NSGA-II as a base-line algorithm (Table 9.1).

Table 9.1 Featurs of test problems

Problem name	No. of pareto subsets	PF shape
SSUF1	2	Concave
SSFU3	2	Concave
MMF3	2	Concave
MMF4	4	Concave
MMF5	4	Convex
MMF6	4	Convex

9.4.1 Test Problems

We took 6 multimodal multi-objective test functions from the literature SSUF1, SSUF3 [1] and MMF3-MMF6 [7]. These test problems have different shapes and properties of the PF (concave and convex).

9.4.2 Parameter Settings

In the following we explain the parameter setting used in the comparisons. The population size is set to 100 and we used 10,000 function evaluations as a termination criterion in all the experiments. We calculate the median and interquartile (IQR) ranges out of 31 independent runs. We used Simulated Binary Crossover (SBX) and Polynomial Mutation (PM) as variation operators. The distribution index for both crossover and mutation is set to 20. The recombination probability $P_c = 1$ and the mutation probability $P_m = 1/D$. In a first study, in order to decide the optimal grid size of MDCD, the performance with different grid sizes is compared, and the best grid size reported is successively used in the following experiments. For the literature algorithms used to compare, we used the parameter setting as in the literature where the WSCD value is obtained by equally division of weights in both decision and objective spaces. The implementation of these algorithms, as well as NSGA-II, is provided in the Matlab-based platform PlatEmo [7].

9.4.3 Performance Measures

To assess the performance of the proposed method and the compared algorithms, we used the Inverted Generational Distance in decision space (IGDX) [6]. The obtained values demonstrate both the diversity and convergence of solutions in decision space by calculating the Euclidean distance between the PS and the set of obtained solutions in decision space. The mathematical definition of IGDX is:

$$IGDX(P^{*}, R) = \frac{\sum_{v \in P^{*}} \|R - v\|_2}{|P^{*}|} \tag{9.3}$$

Where R and P^* accordingly are a set of obtained solutions in decision space and a sample of the PS, and $\|R - v\|_2$ is the minimum Euclidean distance between the sampled point v and any point in R.

We also look at the diversity and convergence of the obtained solutions in objective space, by calculation of Inverted Generational distance (IGD) [11, 12], which is mathematically formatted in the same way as IGDX as follows:

$$IGD(P^{*}, R) = \frac{\sum_{v \in P^{*}} \|\mathbf{f}(R) - \mathbf{f}(v)\|_2}{|P^{*}|} \tag{9.4}$$

where R and P^* respectively are a set of obtained solutions in objective space and a sample of the PF.

Moreover, in order to better demonstrate the performance of the proposed method, we used the Pareto Set Proximity (PSP) performance indicator to evaluate the approximation of the obtained solutions in decision space [7]. PSP is computed as follows:

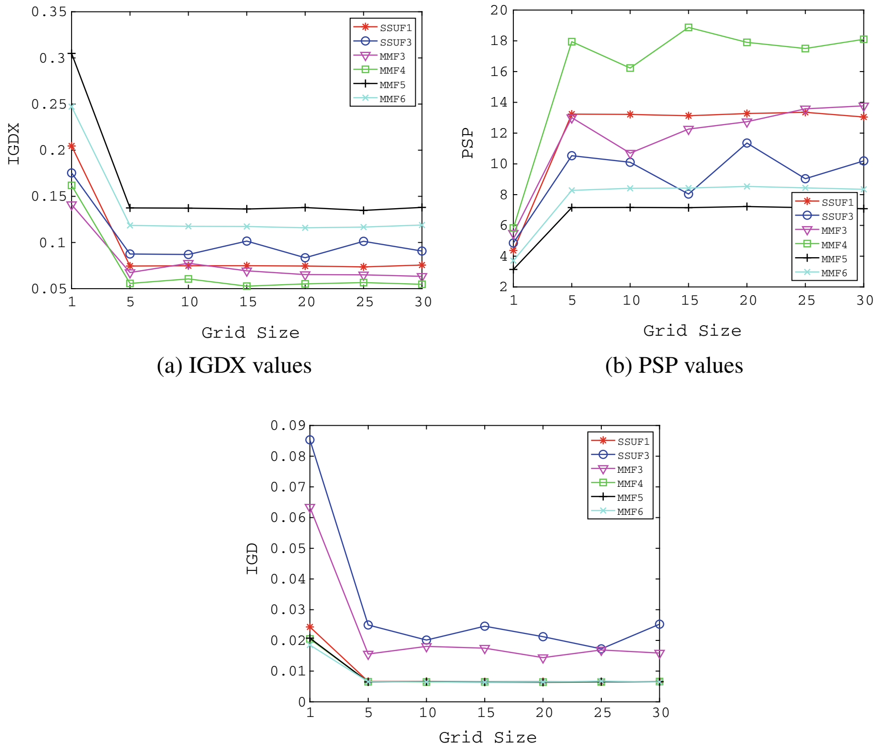
$$PSP(P^{*}, R) = \frac{CR(P^{*}, R)}{IGDX(P^{*}, R)} \tag{9.5}$$

where CR, i.e. the cover rate, is a modification of the Maximum Spread (MS) for decision space. A high value of the PSP indicator represents a better performance of the algorithm in terms of diversity in decision space.

9.5 Analysis of Results

First, in order to evaluate the impacts of grid size on the performance of the proposed algorithm, we conducted a experimental comparison with 1, 5, 10, 15, 20, 25 and 30 grids. To perform this experiment, the population size is fixed to 100, and all the parameter values are as explained in the subsection of Parameter Settings. The results of the grid size comparison are presented in Fig. 9.2. As can be seen from Fig. 9.2a, b, c, when increasing the size of the grids from 1 to 5, the IGDX and IGD values decrease, while the PSP values increase accordingly. On the other hand, by increasing the grid sizes from 5 to 30, the IGD and IGDX on most problems keep their steady states and do not show changes. In some of the problems, some changes can be appreciated between different sizes, but there is no clear increasing or decreasing behaviour as the grid size increases.

A simple explanation for these observations is that by increasing the size of the grids, the Manhattan distance between all the solutions in decision space is expanding accordingly. Therefore, by increasing the value of multiplication of the Manhattan distance and crowding distance, the same solutions are selected for the selection



(a) IGDX values

(b) PSP values

(c) IGD values

Fig. 9.2 Achieved **a** IGDX values, **b** PSP values and **c** IGD values by NSGA-II-MDCD algorithm with different grid sizes

process for the different grid sizes. Following this results, a grid size of 10 is selected for the rest of experiments.

Now, the IGDX, IGD and PSP results for the different algorithms compared are presented in Tables 9.2, 9.3 and 9.4, respectively. The Mann-Whitney U statistical test is taken to test statistical significance according to the best algorithm on each test problem, and the significance is assumed for a value of $p \leq 0.05$. The values highlighted in bold represents the best values for each problem, and the asterisks (*) demonstrate the significance compared to the best algorithm for each test problem.

As can be observed in Tables 9.2 and 9.4, NSGA-II-MDCD performs the best in terms of IGDX and PSP compared to the rest of the algorithms for four out of six test problems, which means that the proposed algorithm provides better distribution of solutions in the decision space. Even though NSGA-II-WSCD algorithm is getting better results for MMF3 and MMF4 compared to the proposed method, no statistical significance was observed between these two algorithms. A possible explanation for this might be that by griding the decision space, in MMF3 and MMF4, as the optimal solutions are more concentrated in concrete grids, then the NSGA-II-MDCD

Table 9.2 IGDX values for comparison of different algorithms

	NSGA-II-MDCD	NSGA-II-WSCD	NSGA-II
SSUF1	**0.07478 (0.00849)**	0.07923 (0.00741)*	0.1051 (0.0151)*
SSFU3	**0.08699 (0.072)**	0.08949 (0.07273)	0.1021 (0.0853)
MMF3	0.07747 (0.03521)	**0.05839 (0.03499)**	0.07854(0.0314)
MMF4	0.06053 (0.01059)	**0.05793 (0.01098)**	0.11921 (0.04185)*
MMF5	**0.13723 (0.01042)**	0.14473 (0.01112)*	0.19475 (0.03932)*
MMF6	**0.11752 (0.00682)**	0.12406 (0.01261)*	0.18852 (0.06103)*

Table 9.3 IGD values for comparison of different algorithms

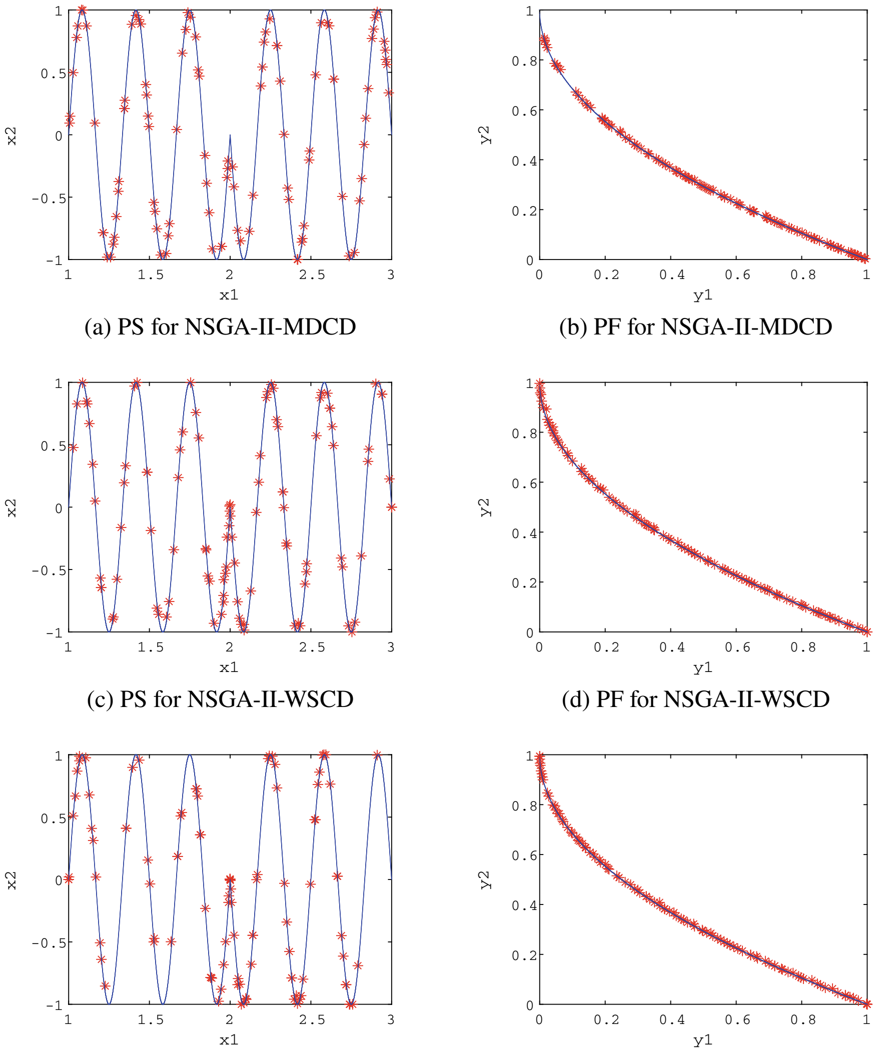
	NSGA-II-MDCD	NSGA-II-WSCD	NSGA-II
SSUF1	0.00662 (0.00053)*	0.00544 (0.00032)	**0.00532 (0.00026)**
SSFU3	0.02011 (0.02278)	0.01696 (0.0146)	**0.01995 (0.01253)**
MMF3	0.01805 (0.01474)	0.01527 (0.01329)	**0.0149 (0.00972)**
MMF4	0.00645 (0.00035)*	0.00542 (0.00025)*	**0.00517 (0.00019)**
MMF5	0.00655 (0.00034)*	0.00559 (0.00032)*	**0.00534 (0.00032)**
MMF6	0.00647 (0.0005)*	0.00549 (0.00028)*	**0.00531 (0.00026)**

Table 9.4 PSP values for comparison of different algorithms

	NSGA-II-MDCD	NSGA-II-WSCD	NSGA-II
SSUF1	**13.2156 (1.48997)**	12.50492 (1.24596)*	9.29939 (1.41609)*
SSFU3	**10.10445 (7.29567)**	9.36013 (11.64698)	9.42434 (8.32393)
MMF3	10.69778 (0.03521)	**14.95064 (0.03499)**	12.73248(0.0314)
MMF4	16.23044 (3.10691)	**17.15216 (3.1857)**	8.20806 (3.14241)*
MMF5	**7.16681 (0.45601)**	6.8738 (0.59585)*	4.89867 (1.33815)*
MMF6	**8.41297 (0.53495)**	7.92392 (1.67318)*	5.14445 (0.82706)*

is getting worse results compared to other problems with optimal solutions involved in larger number of grids. As we expected from Table 9.3, the IGD value of the NSGA-II algorithm shows its superiority in comparison with the proposed algorithm. The reason is that the main focus of NSGA-II algorithm is to get a better diversity of solutions in objective space, while neglecting decision space, therefore a lower IGD value is expected. According to the further analysis of results we could claim that NSGA-II-MDCD algorithm provides a better approximation of PS while not disturbing that much the approximation of PF.

Moreover, the results solutions obtained in the median execution for the three algorithms over the different data sets are represented in Figs. 9.3, 9.4, 9.5 and 9.6, both in decision and objective spaces. In these figures, the true PS and PF are represented in blue, so the clustered solutions can be appreciated. As can be seen



(a) PS for NSGA-II-MDCD

(b) PF for NSGA-II-MDCD

(c) PS for NSGA-II-WSCD

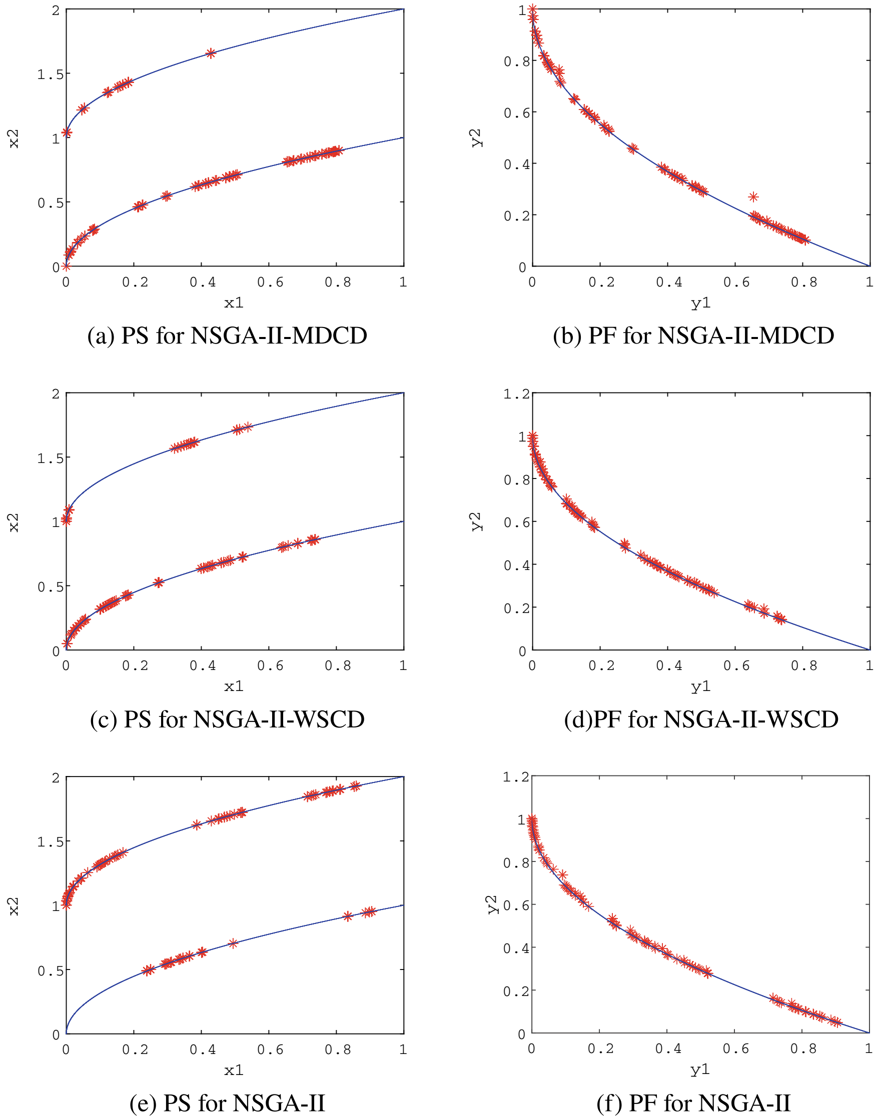
(d) PF for NSGA-II-WSCD

(e) PS for NSGA-II

(f) PF for NSGA-II

Fig. 9.3 Obtained solutions in decision and objective space for SSUF1 problem

from Fig. 9.3, as an instance, the solutions of NSGA-II-MDCD are more evenly distributed in decision space than the solutions of NSGA-II-WSCD and NSGA-II algorithms. In objective space, the algorithm is still obtaining a good approximation of the PF, but some parts of it are less crowded than others in comparison with NSGA-II.



(a) PS for NSGA-II-MDCD

(b) PF for NSGA-II-MDCD

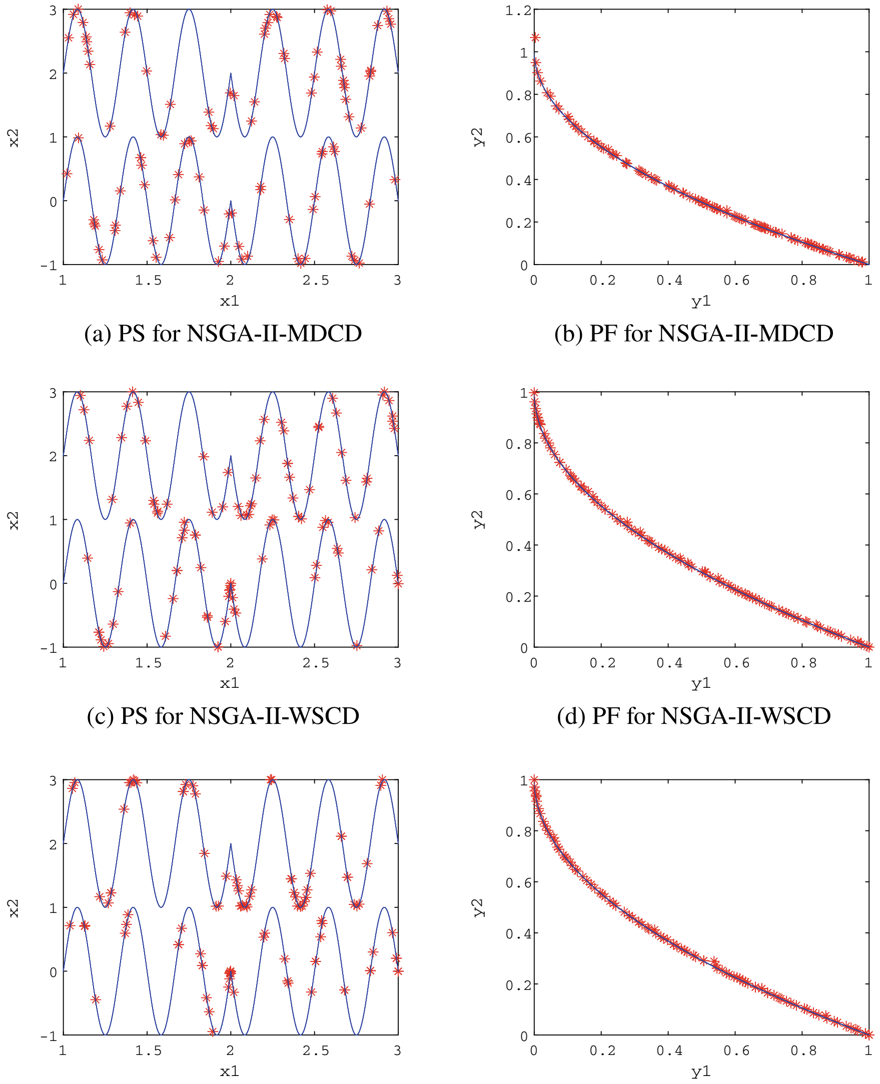
(c) PS for NSGA-II-WSCD

(d)PF for NSGA-II-WSCD

(e) PS for NSGA-II

(f) PF for NSGA-II

Fig. 9.4 Obtained solutions in decision and objective space for SSUF3 problem



(a) PS for NSGA-II-MDCD

(b) PF for NSGA-II-MDCD

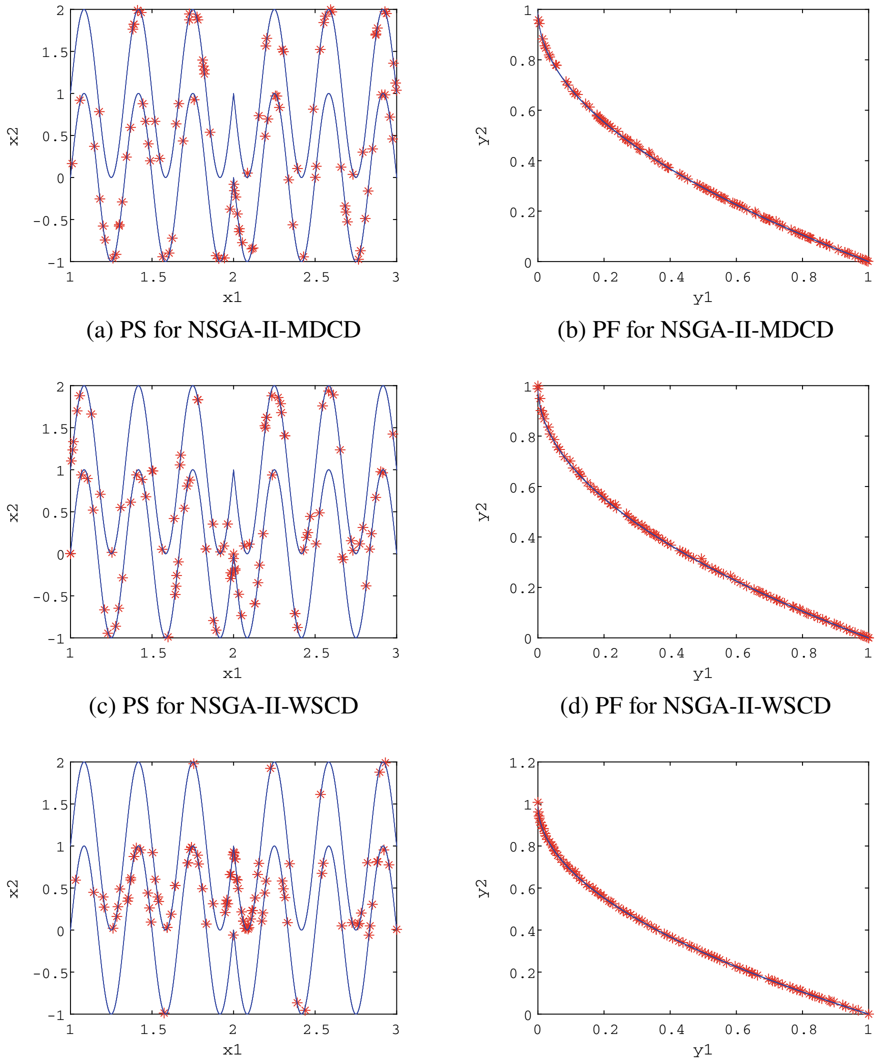
(c) PS for NSGA-II-WSCD

(d) PF for NSGA-II-WSCD

(e) PS for NSGA-II

(f) PF for NSGA-II

Fig. 9.5 Obtained solutions in decision and objective space for MMF5 problem



(a) PS for NSGA-II-MDCD

(b) PF for NSGA-II-MDCD

(c) PS for NSGA-II-WSCD

(d) PF for NSGA-II-WSCD

(e) PS for NSGA-II

(f) PF for NSGA-II

Fig. 9.6 Obtained solutions in decision and objective space for MMF6 problem

9.6 Conclusions

The goal of this study is to develop a method for MMOPs to provide a better approximation of solutions in the decision space. It is important to note that the good diversity of solutions in objective space does not guarantee a good diversity of solutions in decision space. As a result, we propose a technique to focus on increasing the distribution of solutions in decision space. We combine the Manhattan distance metric with crowding distance in decision space to satisfy our goal. Both distance measurement metrics together help to make a better distribution of solutions in decision space. The results of our experiments with 6 test problems show the superiority of the proposed method according to the approximation of PS over the NSGA-II-WSCD and NSGA-II algorithms. Moreover, we investigate the effects of gridding size, where the results show that the grid size does not change the performance of the proposed algorithm.

Further studies are required to investigate and develop new techniques providing the ability of better local search to locate more optimal solutions.

References

1. Liang J, Yue C, Qu B (2016) Multimodal multi-objective optimization: a preliminary study. In: Evolutionary Computation (CEC). IEEE, Vancouver, pp 2454–2461
2. Liu Y, Ishibuchi H, Nojima Y, Masuyama N, Shang K (2018) A double-niched evolutionary algorithm and its behavior on polygon-based problems. In: International conference on swarm intelligence. Springer, Coimbra, pp 262–273
3. Cuate O, Schütze O (2019) Variation rate: an alternative to maintain diversity in decision space for multi-objective evolutionary algorithms. In: International conference on evolutionary multi-criterion optimization. Springer, Berlin, pp 203–215
4. Deb K,Tiwari S (2005) Omni-optimizer: a procedure for single and multi-objective optimizationn. In: International conference on evolutionary multi-criterion optimization. Springer, Guanajuato, pp 47–61
5. Deb K, Pratap A (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
6. Zhou A, Zhang Q, Jin Y (2009) Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. IEEE Trans Evol Comput 13:1167–1189
7. Yue C, Qu B, Liang J (2018) A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Trans Evol Comput 22:805–817
8. Liang J, Xu W, Yue C, Yu K, Song H et al (2019) Multimodal multiobjective optimization with differential evolution. Swarm Evol Comput 44:1028–1059
9. Javadi M, Zille H, Mostaghikm S (2016) Modified crowding distance and mutation for multimodalmulti-objective optimization. ACM, Prague, To appear
10. Yusoff Y, Zhang Q, Ngadiman M, Zain A (2011) Overview of NSGA-II for optimizing machining process parameters. Procedia Eng 15:3978–3983
11. Reyes-Sierra M, Coello CA (2005) A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization: an experimental study. In: International conference on electronics computer technology. IEEE, Edinburgh, pp 65–72
12. Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. IEEE Trans Evol Comput 12:41–63

Chapter 10
An Unsteady Aerodynamic/Aeroacoustic Optimization Framework Using Continuous Adjoint

M. Monfaredi, X. S. Trompoukis, K. T. Tsiakas, and K. C. Giannakoglou

Abstract In this paper, an unsteady aerodynamic/aeroacoustic optimization framework is presented. This is based on the continuous adjoint method to a hybrid acoustic prediction tool, in which the near-field flow solution results from an unsteady CFD simulation while the acoustic propagation to far-field makes use of an acoustic analogy. The CFD simulation is performed using the in-house GPU-enabled URANS equations' solver for which a continuous adjoint solver is available. The noise prediction tool and its adjoint are developed based on the permeable version of the Ffowcs Williams and Hawkings (FW-H) analogy, solved in the frequency domain. Its implementation is verified w.r.t. the analytical solution of the sound field from a monopole source in uniform flow. Then, the accuracy of the hybrid solver is verified by comparing the sound directivity computed by the FW-H analogy with that of a CFD run, for a 2D pitching airfoil in a subsonic inviscid flow. The accuracy of the sensitivities computed using the unsteady adjoint solver is verified w.r.t. those computed by finite differences. Finally, the programmed software is used to optimize the shape of the pitching airfoil, aiming at min. noise with an equality constraint for the lift.

Keywords Aeroacoustic · Ffowcs Williams and Hawkings analogy · Shape optimization · Adjoint method

M. Monfaredi (✉) · X. S. Trompoukis · K. T. Tsiakas · K. C. Giannakoglou
Parallel CFD & Optimization Unit, School of Mechanical Engineering, National Technical University of Athens (NTUA), Athens, Greece
e-mail: morteza.monfaredi@gmail.com

X. S. Trompoukis
e-mail: xeftro@gmail.com

K. T. Tsiakas
e-mail: tsiakost@gmail.com

K. C. Giannakoglou
e-mail: kgianna@central.ntua.gr

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_10

10.1 Introduction

During the last decades, there have been tight regulations regarding noise pollution which underline the importance of an effective noise source mitigation strategy. For example, based on the Flightpath 2050 report of the European Commission [1], it is mandated to reduce the perceived noise by 65% from its level in 2000, by the year 2050. This means designers must investigate innovative methods to further improve the process of designing quieter and more efficient systems. Among the various existing methods performing numerical optimization, adjoint methods [2, 3], are advantageous since their computational cost is independent of the number of design variables.

Although adjoint methods have a strong background in aerodynamic shape optimization [4], they are relatively new in the field of aeroacoustic optimization [5–12]. In [5], a steady continuous adjoint method was presented for the reduction of the noise perceived by the car driver due to its side mirror using a turbulence-based surrogate objective function. With this model, the omission of the adjoint to the turbulence model equations would merely lead to zero sensitivities, since the objective depends only on turbulence. In [6], a discrete adjoint to a hybrid URANS-FW-H solver was developed for inverse shape design and turbulent blunt trailing edge noise reduction. Recently, the same method has been used to perform shape optimization to reduce the far-field noise from a pitching airfoil in an inviscid flow [7], a 2D wing-flap in laminar flow [8], a rod-airfoil in turbulent flow [9] and a jet-flap interaction in turbulent flow [10]. In these works [6–10], discrete adjoint was used with the help of automatic differentiation. Regarding continuous adjoint, the permeable FW-H formula is solved using a finite element method, leading to the necessary adjoint conditions at the interface between the Computational Fluid Dynamics (CFD) and Computational Aeroacoustics (CAA) domains [11]. The continuous adjoint for a hybrid solver for incompressible flow models and the Kirchhoff integral, for automotive applications, can also be found in the literature [12]. To the author's knowledge, the continuous adjoint method to compressible flows based on the FW-H analogy appears, for the first time, in this paper.

For the first verification of the method presented in this paper, the CFD model is restricted to the Euler equations. A hybrid aeroacoustic noise prediction tool is built on the in-house GPU-enabled flow solver [13, 14], by additionally implementing the FW-H analogy. For its verification, numerical results are compared with the analytical solution of a monopole sound source in a flow-stream, and CFD results for a 2D pitching isolated airfoil. Then, the continuous adjoint method is verified and used to perform shape optimization, with an aeroacoustic objective function and an aerodynamic equality constraint.

10.2 Governing Equations

10.2.1 Flow Equations

The 2D unsteady inviscid flow equations of a compressible fluid are discretized using a dual-time stepping method, being second-order accurate in time. Spatial discretization is based on vertex-centered finite volume. Convective fluxes are computed using the upwind Roe scheme, with second-order accuracy in space. The discretization of the governing equations in the pitching airfoil case takes the geometrical conservation law into account. Along the far-field boundary, a non-reflecting condition is applied [15].

10.2.2 Noise Prediction Using the FW-H Analogy

Based on the assumption that sound is perceived as pressure fluctuations, acoustic noise can be computed using CFD simulations. However, a purely CFD-based approach may become very expensive when the acoustic noise at a far-field location is of interest, because the fine CFD mesh should be extended far away, up to the receiver's location. The combination of CFD methods and acoustic analogies rely upon the computationally cheaper wave equation. Such methods are usually referred to as hybrid methods and their origin can be traced back to the Lighthill analogy [16]; this was later extended by Curle [17] to account for the presence of stationary solid surface and Ffowcs Williams and Hawkings to include moving surfaces [18]. In this paper, the permeable version of FW-H analogy is used. The resulting wave equation, a.k.a. the FW-H equation, reads:

$$\left(\frac{\partial^2}{\partial t^2} - c_\infty^2 \frac{\partial^2}{\partial x_i \partial x_i}\right)\left(\frac{H(f)p'}{C_\infty^2}\right) = \frac{\partial}{\partial t}(Q\delta(f)) - \frac{\partial}{\partial x_i}(F_i\delta(f)) + \frac{\partial^2}{\partial x_i \partial x_j}(H(f)T_{ij}) \quad (10.1)$$

and for bodies in motion (such as a pitching airfoil in uniform flow), the Galilean transformation can be used to transform Eq. 10.1 into a relative system associated with the moving body, as follows [19]:

$$\begin{aligned}\left(\frac{\partial^2}{\partial t^2} + \upsilon_{\infty i}\upsilon_{\infty j}\frac{\partial^2}{\partial x_i \partial x_j} + 2\upsilon_{\infty i}\frac{\partial^2}{\partial x_i \partial t} - c_\infty^2\frac{\partial^2}{\partial x_i \partial x_i}\right)\left(\frac{H(f)p'}{C_\infty^2}\right) \\ = \frac{\partial}{\partial t}(Q\delta(f)) - \frac{\partial}{\partial x_i}(F_i\delta(f)) + \frac{\partial^2}{\partial x_i \partial x_j}(H(f)T_{ij})\end{aligned} \quad (10.2)$$

where f is the signed distance from the interface of the CFD and CAA domains, as shown in Fig. 10.1. This interface will be referred to as the FW-H surface. The FW-H surface lays inside the CFD domain though far away from the body in order for this not to be affected by changes in the body shape to be designed. H is the Heaviside function, δ is the Dirac delta function and c_∞ is the free-stream sound speed. $Q(\vec{x}, t) =$

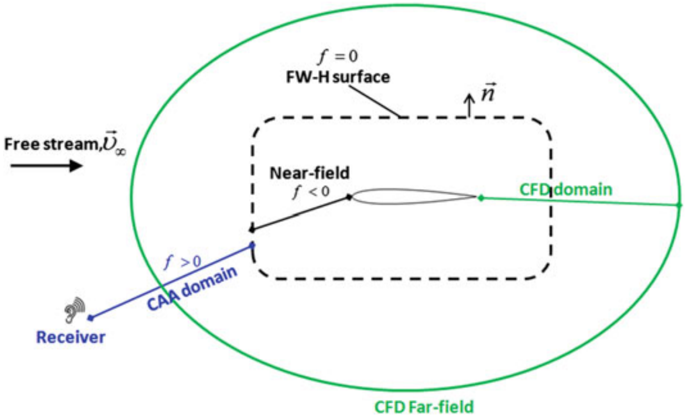


Fig. 10.1 Schematic of permeable FW-H surface. The dashed-line shows the interface between the CFD and CAA domains

$(\rho \upsilon_i - \rho_\infty \upsilon_{\infty i}) n_i$, $F_i(\vec{x}, t) = (\rho(\upsilon_i - 2\upsilon_{\infty i}) + \rho_\infty \upsilon_{\infty i} \upsilon_{\infty j} + p\delta_{ij} - \tau_{ij}) n_j$ and $T_{ij}(\vec{x}, t) = \rho(\upsilon_i - \upsilon_{\infty i})(\upsilon_j - \upsilon_{\infty j}) + (p - c_\infty^2 \rho)\delta_{ij}$ are known as the monopole, dipole and quadrupole source terms, respectively, defined along the FW-H surface. $\rho = \rho_\infty + \rho'$, $p = p_\infty + p'$ and $\upsilon_i = \upsilon_{\infty i} + \upsilon_i'$ are local density, pressure and velocity components, respectively, and τ_{ij} is the viscous stress tensor. Free-stream quantities are indexed by ∞. $\vec{n}$ is the unit normal vector to the FW-H surface pointing towards the CAA domain. δ_{ij} is the Kronecker delta.

For 3D problems, integral solutions to the FW-H equation are available in the time domain. However, in 2D problems, to avoid tail effects, an infinitely long time integration range must be used. To avoid this, Eq. 10.2 is transformed into the frequency domain using the Fourier transformation as follows:

$$\left(\frac{\partial^2}{\partial x_i \partial x_i} + k^2 - 2iM_{\infty i} k \frac{\partial}{\partial x_i} - M_{\infty i} M_{\infty j} \frac{\partial^2}{\partial x_i \partial x_j}\right)\left(H(f)\hat{p}'\right) \\ = -i\omega \hat{Q}\delta(f) + \frac{\partial}{\partial x_i}(\hat{F}_i \delta(f)) - \frac{\partial^2}{\partial x_i \partial x_j}(H(f)\hat{T}_{ij}) \tag{10.3}$$

where the hat symbol ($\hat{}$) denotes frequency domain variables and ω is the frequency. $M_{\infty i} = \upsilon_{\infty i}/c_\infty$ and the wave number is $k = \omega/c_\infty$. Equation 10.3 is solved by convolving it with the appropriate Green function. Then, the pressure fluctuation in the frequency domain, at the receiver's location, results from:

$$H(f)\hat{p}'(\vec{x}_o, \omega) = -\oint_{f=0} i\omega \hat{Q}(\vec{x}_s, \omega)\hat{G}(\vec{x}_o, \vec{x}_s, \omega) ds \\ - \oint_{f=0} \hat{F}_i(\vec{x}_s, \omega)\frac{\partial \hat{G}(\vec{x}_o, \vec{x}_s, \omega)}{\partial x_{si}} ds - \oint_{f>0} \hat{T}_{ij}(\vec{x}_s, \omega)\frac{\partial^2 \hat{G}(\vec{x}_o, \vec{x}_s, \omega)}{\partial x_{si} \partial x_{sj}} dV \tag{10.4}$$

where $\vec{x}_o$ and $\vec{x}_s$ are the receiver and sources' (sources are located on the FW-H surface) positions, respectively. $\hat{G}(\vec{x}_o, \vec{x}_s, \omega)$ is the 2D Green function for subsonic flows in the frequency domain, which is defined as:

$$\hat{G}(\vec{x}_o, \vec{x}_s, \omega) = \frac{i}{4\beta}\exp(iM_\infty k\bar{x}_1/\beta^2)H_0^{(2)}\left(\frac{k}{\beta^2}\sqrt{\bar{x}_1^2 + \beta^2\bar{x}_2^2}\right) \qquad (10.5)$$

$$\bar{x}_1 = (x_{o1} - x_{s1})\cos\theta + (x_{o2} - x_{s2})\sin\theta \qquad (10.6)$$

$$\bar{x}_2 = -(x_{o1} - x_{s1})\sin\theta + (x_{o2} - x_{s2})\cos\theta \qquad (10.7)$$

In the above equations, θ is the free-stream flow angle, such that $\tan\theta = \upsilon_{\infty 2}/\upsilon_{\infty 1}$, M_∞ is the free-stream Mach number and the Prandtl-Glauert factor is $\beta = \sqrt{1 - M_\infty^2}$. $H_0^{(2)}$ stands for Hankel function of the second kind of zero order.

For the low-speed cases this paper is dealing with, the contribution of quadrupole terms can be neglected, avoiding thus the computation of a volume integral. The noise prediction module is combined with the in-house flow solver as follows: first, an unsteady flow solution is performed in the CFD domain and, at the end of each time step, source terms Q and F_i are computed over the FW-H surface. Upon completion of the unsteady CFD simulation, the mean value of each source is subtracted from instantaneous values since the mean value corresponds to zero frequency that does not generate noise. Since it is hard to achieve pure periodic results, a Hanning window is applied to the sources to eliminate discontinuity between the first and last points, followed by a Fourier transform. At the end, pressure fluctuations in the frequency domain are computed using Eq. 10.4.

10.3 Formulation of the Continuous Adjoint Method

In aerodynamic shape optimization, adjoint methods compute the gradient of an objective function w.r.t. the design variables. The objective functions, such as the lift, drag etc. are integral quantities defined along the solid boundaries and contribute to either the adjoint boundary conditions or the adjoint sensitivities. On the other hand, in aeroacoustic problems, the objective function is defined at the remote receiver's location, $\vec{x}_o$, and affects neither the adjoint boundary conditions nor the sensitivities; instead this contributes to the adjoint equations in the form of source terms applied along the FW-H surface. An objective function J, originally written as a time integral of p', can also be expressed in the frequency domain as:

$$J = \int_\omega \left|\hat{p}'(\vec{x}_o, \omega)\right| \mathrm{d}\omega \qquad (10.8)$$

where $\hat{p}'(\vec{x}_o, \omega)$ is the outcome of Eq. 10.4 and $\left|\hat{p}'\right| = \sqrt{\hat{p}'^2_{Re} + \hat{p}'^2_{Im}}$, where subscripts Re and Im refer to the real and imaginary parts of complex variables. Here, the integration range is over the whole frequency domain.

To formulate the continuous adjoint problem, an augmented objective function is defined as $F_{aug} = J + \int_T \int_\Omega \psi_n R_n d\Omega dt$, where $n = 1, 4$ and ψ_n, R_n, Ω and T are the adjoint variable fields, the residuals of the unsteady Euler equations, the CFD domain and the solution period, respectively. By differentiating F_{aug} w.r.t. the design variables b_n and setting the multipliers of the variations in the flow variables to zero, the unsteady adjoint equations are obtained as:

$$-\frac{\partial \psi_m}{\partial t} - A_{nmk}\frac{\partial \psi_n}{\partial x_k} + S_{FW-H_m}\delta(f) = 0 \tag{10.9}$$

where $A_{nmk} = \frac{\delta g_{nk}}{\delta U_m}$. U_m and g_{nk} are the conservative flow variables and inviscid fluxes, respectively. The adjoint boundary condition along the solid walls is $\psi_{m+1} n_{w_m} + (u_m^{grid} n_{w_m})\psi_4 = 0$, where $\vec{n}_w$ is the unit normal to the wall and u_m^{grid} is the grid velocity at each node on the pitching airfoil. b_n are the coordinates of the control points of the shape parameterization method which is based on Bezier polynomials.

In Eq. 10.9, S_{FW-H_m} is a term that includes contributions from the FW-H analogy to the adjoint equations. To find this term, Eq. 10.8 is differentiated w.r.t. b_n, as follows:

$$\frac{\delta J}{\delta b_n} = \int\limits_{\omega} \frac{1}{\left|\hat{p}'\right|}\left(\hat{p}'_{\text{Re}}\frac{\delta \hat{p}'_{\text{Re}}}{\delta b_n}\right) d\omega + \int\limits_{\omega} \frac{1}{\left|\hat{p}'\right|}\left(\hat{p}'_{\text{Im}}\frac{\delta \hat{p}'_{\text{Im}}}{\delta b_n}\right) d\omega \tag{10.10}$$

For the sake of simplicity, starting from Eq. 10.10, $\hat{p}'(\vec{x}_o, \omega)$, $\hat{G}(\vec{x}_o, \vec{x}_s, \omega)$, $\hat{F}_k(\vec{x}_s, \omega)$ and $\hat{Q}(\vec{x}_s, \omega)$ are shorted to $\hat{p}'$, $\hat{G}$, $\hat{F}_k$ and $\hat{Q}$, respectively. The real and imaginary part of the $\hat{p}'$ can be found based on Eq. 10.4. Since the grid does not change at the FW-H surface location during the optimization, the derivatives of the Green function and its spatial derivatives as well as those of the surface element ds, w.r.t. b_n are zero. So, the variation of the real and imaginary part of $\hat{p}'$ w.r.t. b_n read:

$$\begin{aligned}\frac{\delta \hat{p}'_{\text{Re}}}{\delta b_n} =& -\oint\limits_{f=0}\left[\left(\frac{\delta \hat{F}_k}{\delta b_n}\right)_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}} - \left(\frac{\delta \hat{F}_k}{\delta b_n}\right)_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}}\right] ds \\ &+ \oint\limits_{f=0} \omega\left[\left(\frac{\delta \hat{Q}}{\delta b_n}\right)_{\text{Re}} \hat{G}_{\text{Im}} + \left(\frac{\delta \hat{Q}}{\delta b_n}\right)_{\text{Im}} \hat{G}_{\text{Re}}\right] ds\end{aligned} \tag{10.11}$$

and

$$\frac{\delta \hat{p}'_{\text{Im}}}{\delta b_n} =- \oint_{f=0} \Bigg[\left(\frac{\delta \hat{F}_k}{\delta b_n}\right)_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}} + \left(\frac{\delta \hat{F}_k}{\delta b_n}\right)_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}}\Bigg] ds$$
$$- \oint_{f=0} \left[\omega \left(\left(\frac{\delta \hat{Q}}{\delta b_n}\right)_{\text{Re}} \hat{G}_{\text{Re}} - \left(\frac{\delta \hat{Q}}{\delta b_n}\right)_{\text{Im}} \hat{G}_{\text{Im}}\right)\right] ds \tag{10.12}$$

By introducing Eqs. 10.11 and 10.12 in Eq. 10.10, the derivatives of J become:

$$\frac{\delta J}{\delta b_n} = -\int_{\omega} \oint_{f=0} \left[\frac{1}{|\hat{p}'|}\left(\hat{p}'_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}} + \hat{p}'_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}}\right)\left(\frac{\delta \hat{F}_k}{\delta b_n}\right)_{\text{Re}}\right] ds d\omega \tag{10.13}$$
$$-\int_{\omega} \oint_{f=0} \left[\frac{1}{|\hat{p}'|}\left(\hat{p}'_{\text{Im}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Re}} - \hat{p}'_{\text{Re}}\left(\frac{\delta \hat{G}}{\delta x_{s_k}}\right)_{\text{Im}}\right)\left(\frac{\delta \hat{F}_k}{\delta b_n}\right)_{\text{Im}}\right] ds d\omega$$
$$-\int_{\omega} \oint_{f=0} \omega \frac{1}{|\hat{p}'|}\left[\left(-\hat{p}'_{\text{Re}}\hat{G}_{\text{Im}} + \hat{p}'_{\text{Im}}\hat{G}_{\text{Re}}\right)\left(\frac{\delta \hat{Q}}{\delta b_n}\right)_{\text{Re}}\right] ds d\omega$$
$$-\int_{\omega} \oint_{f=0} \omega \frac{1}{|\hat{p}'|}\left[\left(\hat{p}'_{\text{Re}}\hat{G}_{\text{Re}} - \hat{p}'_{\text{Im}}\hat{G}_{\text{Im}}\right)\left(\frac{\delta \hat{Q}}{\delta b_n}\right)_{\text{Im}}\right] ds d\omega$$

In Eq. 10.13, $\frac{\delta \hat{F}_k}{\delta b_n}$ and $\frac{\delta \hat{Q}}{\delta b_n}$ include derivatives of the flow variables w.r.t. the design variables in the frequency domain. However, these variations should be expressed in the time domain for them to contribute to the adjoint flow equations. To do so, the Fourier transformation needs to be included in Eq. 10.13, by considering the subtraction of the time-averaged value of F_k and Q from their instantaneous values, along with a multiplication with the Hanning window $\mathcal{H}(t)$ before transforming them into the frequency domain. Hence, the Fourier transformation for an arbitrary signal $s(t)$ is performed as follows:

$$\hat{s}(\omega) = \frac{1}{T}\int_{T} \mathcal{H}(t)\left[s(t) - \frac{1}{T}\int_{T} s(t)dt\right]e^{-2i\pi\omega t}dt \tag{10.14}$$

Including Eq. 10.14 into Eq. 10.13 and permuting time and frequency integrals, $\frac{\delta J}{\delta b_n}$ reads:

$$\frac{\delta J}{\delta b_n} = -\frac{1}{T}\int_{T} \oint_{f=0} \left[(A_k + B_k)\frac{\delta F_k}{\delta b_n} + (C + D)\frac{\delta Q}{\delta b_n}\right] ds dt \tag{10.15}$$

where

$$A_k = \int_{\omega} \left(\frac{\hat{p}'_{\text{Re}}}{|\hat{p}'|}\left(\frac{\partial \hat{G}}{\partial x_{s_k}}\right)_{\text{Re}} + \frac{\hat{p}'_{\text{Im}}}{|\hat{p}'|}\left(\frac{\partial \hat{G}}{\partial x_{s_k}}\right)_{\text{Im}}\right)(\mathcal{H}(t)\cos(2\pi\omega t) - H_c(\omega))d\omega \tag{10.16}$$

$$B_k = \int_{\omega} \left(\frac{\hat{p}'_{\text{Re}}}{|\hat{p}'|}\left(\frac{\partial \hat{G}}{\partial x_{s_k}}\right)_{\text{Im}} + \frac{\hat{p}'_{\text{Im}}}{|\hat{p}'|}\left(\frac{\partial \hat{G}}{\partial x_{s_k}}\right)_{\text{Re}}\right)(\mathcal{H}(t)\sin(2\pi\omega t) - H_s(\omega))d\omega \tag{10.17}$$

$$C = \int_{\omega} \left(\frac{\hat{p}'_{\text{Im}}}{|\hat{p}'|}\hat{G}_{\text{Re}} - \frac{\hat{p}'_{\text{Re}}}{|\hat{p}'|}\hat{G}_{\text{Im}}\right)(\mathcal{H}(t)\cos(2\pi\omega t) - H_c(\omega))d\omega \tag{10.18}$$

$$D = \int_{\omega} \left(\frac{\hat{p}'_{\mathrm{Im}}}{|\hat{p}'|} \hat{G}_{\mathrm{Im}} - \frac{\hat{p}'_{\mathrm{Re}}}{|\hat{p}'|} \hat{G}_{\mathrm{Re}} \right) (\mathcal{H}(t) \sin(2\pi \omega t) - H_s(\omega)) d\omega \tag{10.19}$$

$$H_c(\omega) = \frac{1}{T} \int_{T} \mathcal{H}(t) \cos(2\pi \omega t) dt \tag{10.20}$$

$$H_s(\omega) = \frac{1}{T} \int_{T} \mathcal{H}(t) \sin(2\pi \omega t) dt \tag{10.21}$$

Equation 10.15 contains a double time/surface integral over the FW-H surface. Therefore, in order to eliminate the derivatives of the flow variables w.r.t. b_n, this equation is taken into account as source terms (S_{FW-H_m} in Eq. 10.9) at the cells lying along the FW-H surface, when solving the adjoint equations. Since the in-house code solves for the conservative variables, F_k and Q must be expressed in terms of these variables before differentiation, yielding:

$$\begin{aligned} \frac{\delta F_k}{\delta b_n} = \delta_{kj} (\gamma - 1) \left[\frac{|\vec{\upsilon}|^2}{2} \frac{\delta U_1}{\delta b_n} - (\upsilon_j \frac{\delta U_{j+1}}{\delta b_n}) + \frac{\delta U_4}{\delta b_n} \right] n_j \\ + (\upsilon_k - 2\upsilon_{\infty k}) \left[\frac{\delta U_{j+1}}{\delta b_n} n_j \right] + (\upsilon_j n_j) \frac{\delta U_{k+1}}{\delta b_n} - (\upsilon_j n_j) \upsilon_k \frac{\delta U_1}{\delta b_n} \end{aligned} \tag{10.22}$$

$$\frac{\delta Q}{\delta b_n} = n_k \frac{\delta U_{k+1}}{\delta b_n} \tag{10.23}$$

where $k = 1, 2$ and $j = 1, 2$ are the Cartesian directions; γ is the heat capacity ratio. Since the FW-H surface remains invariant during the optimization, for the FW-H surface nodes, total and partial derivatives of flow variable are identical or $\frac{\delta}{\delta b_n} = \frac{\partial}{\partial b_n}$. Using Eqs. 10.22 and 10.23 in Eq. 10.15, replacing total with partial derivatives and canceling all derivatives of the flow variables w.r.t. b_n, the S_{FW-H_m} term reads:

$$S_{FW-H} = \begin{bmatrix} \left\{ \frac{(\gamma-1)}{2} |\boldsymbol{\upsilon}|^2 n_k - (\upsilon_i n_i) \upsilon_k \right\} (A_k + B_k) \\ \{(1-\gamma)\upsilon_1 n_k + (\upsilon_k - 2\upsilon_{\infty k}) n_1\} (A_k + B_k) + (\upsilon_i n_i)(A_1 + B_1) + n_1 (C + D) \\ \{(1-\gamma)\upsilon_2 n_k + (\upsilon_k - 2\upsilon_{\infty k}) n_2\} (A_k + B_k) + (\upsilon_i n_i)(A_2 + B_2) + n_2 (C + D) \\ (\gamma - 1) n_k (A_k + B_k) \end{bmatrix} \tag{10.24}$$

and the J sensitivities are computed as follows:

$$\frac{\delta J}{\delta b_n} = - \int_T \int_{\Omega} \psi_i \frac{\partial U_i}{\partial x_k} \frac{\partial}{\partial t} (\frac{\delta x_k}{\delta b_n}) d\Omega dt - \int_T \int_{\Omega} \psi_i \frac{\partial g_{ik}}{\partial x_e} \frac{\partial}{\partial x_k} (\frac{\delta x_e}{\delta b_n}) d\Omega dt - \int_T \int_s \psi_i g_{ik} \frac{\delta n_k}{\delta b_n} ds dt \tag{10.25}$$

where s is the solid wall which in this case is the airfoil surface.

A single cycle of the CFD-CAA optimization framework is shown in Fig. 10.2.

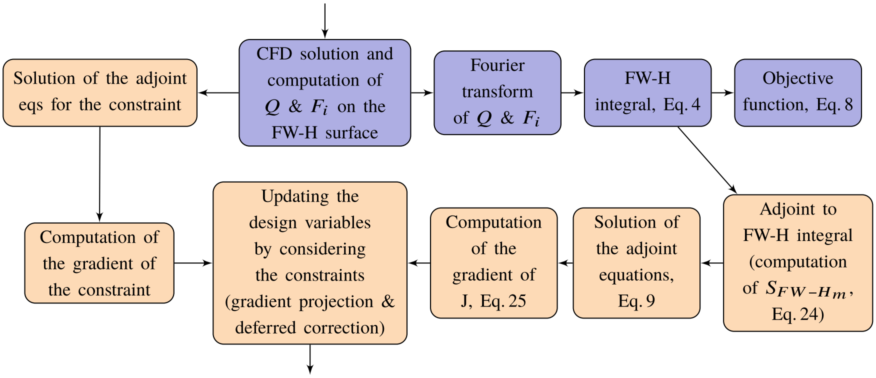

Solution of the adjoint eqs for the constraint
CFD solution and computation of Q & F_i on the FW-H surface
Fourier transform of Q & F_i
FW-H integral, Eq. 4
Objective function, Eq. 8
Computation of the gradient of the constraint
Updating the design variables by considering the constraints (gradient projection & deferred correction)
Computation of the gradient of J, Eq. 25
Solution of the adjoint equations, Eq. 9
Adjoint to FW-H integral (computation of S_{FW-H_m}, Eq. 24)

Fig. 10.2 A single cycle of the CFD-CAA Optimization. Primal and adjoint workflow in blue and orange, respectively

10.3.1 Constraint Imposition Methods

In the constrained case, a gradient projection method with an additional correction term is used to impose an equality constraint on the lift force. Although gradient projection methods are very effective when the constraint function is linear w.r.t. b_n, they lack efficiency otherwise. In case of a non-linear constraint, the optimization is not able to follow the constraint line and gradually deviates from it. To overcome this, the standard gradient projection method is enhanced with a deferred correction.

Let J be the objective function to be minimized subjected to the constraint $L = L_1$. The design variables $\vec{b}$ are updated using a constant step η. Instead of updating each design variable by adding

$$\delta\vec{b}_{projected} = -\eta\left[\vec{\nabla}J - (\vec{\nabla}J \cdot \vec{\nabla}L)\vec{\nabla}L^*\right] \tag{10.26}$$

where $\vec{\nabla} = \frac{\delta}{\delta b_i}$ and $\vec{\nabla}L^* = \frac{\vec{\nabla}L}{|\vec{\nabla}L|}$, an additional correction is applied as follows:

$$\delta\vec{b}_{corrected} = \delta\vec{b}_{projected} - \gamma\vec{\nabla}L^* \tag{10.27}$$

where $\gamma = \frac{\Delta L}{\vec{\nabla}L \cdot \vec{\nabla}L^*}$, and ΔL is the difference between the current and the threshold value of the constraint function.

10.4 Verification of the Hybrid CFD/FW-H Solver

This section is focusing on the verification of the coupled CFD-CAA solver, given that the background CFD tool has adequately been validated in the past [13, 14].

10.4.1 Monopole in Uniform Flow

In the first case, results of the FW-H integral are compared to a well-known analytical solution of the sound field generated by a monopole source in a uniform flow. The stationary monopole source is located at the origin of the coordinate system and there is a uniform flow υ_∞ along the $+x$ direction. The complex velocity potential of the case is [19]:

$$\phi(\vec{x}_o, \vec{x}_s, \omega) = \frac{Ai}{4\beta} \exp i(\omega t + M_\infty k\bar{x}_1/\beta^2)H_0^{(2)}\left(\frac{k}{\beta^2}\sqrt{\bar{x}_1^2 + \beta^2\bar{x}_2^2}\right) \quad (10.28)$$

where $\bar{x}_1$ and $\bar{x}_2$ are the same as in Eq. 10.6 and 10.7. The perturbation field of flow variables and variables needed to compute the F_i and Q in the FW-H integral are obtained from the real parts of $p' = -\rho_0(\frac{\partial\phi}{\partial t} + \upsilon_{\infty 1}\frac{\partial\phi}{\partial x})$, $u' = \nabla\phi$ and $\rho' = p'/{c_0}^2$. In this case, $M_\infty = 0.6$, $A = 0.02\,m^2/s$ and $\omega = 0.162\,rad/s$. Figure 10.3 compares the directivity plot at the radius of $R = 500m$ and Fig. 10.4 shows the time history of p' at a receiver located at $(500m, 0m)$. The results of the FW-H integral exactly match the analytical solution. This is a convincing verification of the implementation of the 2D FW-H formulation, in problems with a uniform mean flow.

10.4.2 Pitching Airfoil in Inviscid Flow

In the second case, a comparison between the hybrid solver and the outcome of a pure CFD simulation is performed. A RAE2822 isolated airfoil is pitching about the quarter-chord point in an inviscid flow, with a 2 de.g. amplitude and period equal to 0.114 sec. The free-stream Mach number is $M_\infty = 0.6$ and the simulation computes 40 time steps per period. A 2D unstructured grid that extends 50 chords away from the airfoil is used, with 51000 nodes overall, among which 202 nodes on the airfoil

Fig. 10.3 Monopole source in uniform flow with $M_\infty = 0.6$. Comparison of the directivity plots at $R = 500$ m

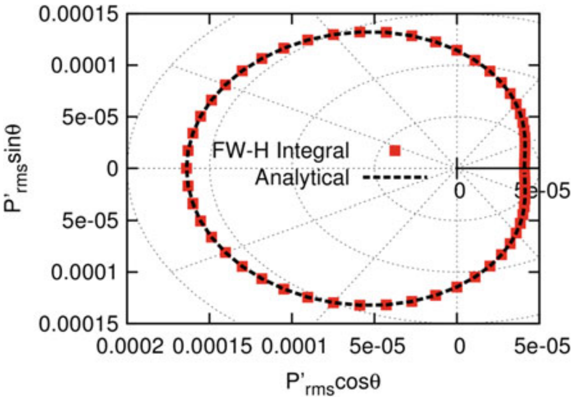


Fig. 10.4 Monopole source in uniform flow with $M_\infty = 0.6$. Comparison of the time history of pressure fluctuation within a period, for a receiver located at (500 m, 0 m)

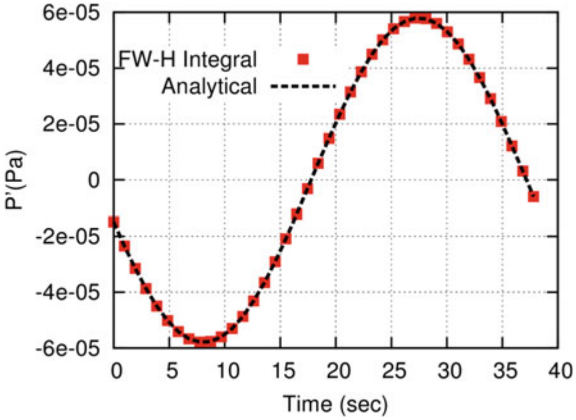
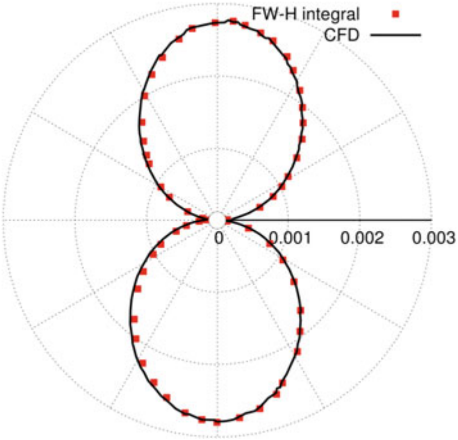


Fig. 10.5 Pitching isolated air-foil. Comparision of the directivity plots (p'_{rms}) at radius R=9C



contour and 151 nodes on the FW-H surface. The FW-H surface is placed at R = 4C from the airfoil mid-chord (0.5C, 0), where C is the airfoil chord length. The directivity pattern at R = 9C is plotted in Fig. 10.5 and shows a very good agreement between results of the unsteady CFD (incl. post-processing of the computed pressure time-series along a circle with R = 9C) and the application of the FW-H integral on the flow time-series computed along the FW-H surface.

10.5 Optimization Results

Before proceeding to the aeroacoustic optimization, the computed gradients using the adjoint solver are verified w.r.t. those obtained by Finite Difference (FD) for the time-averaged lift force. The case and the computational grid are the same as

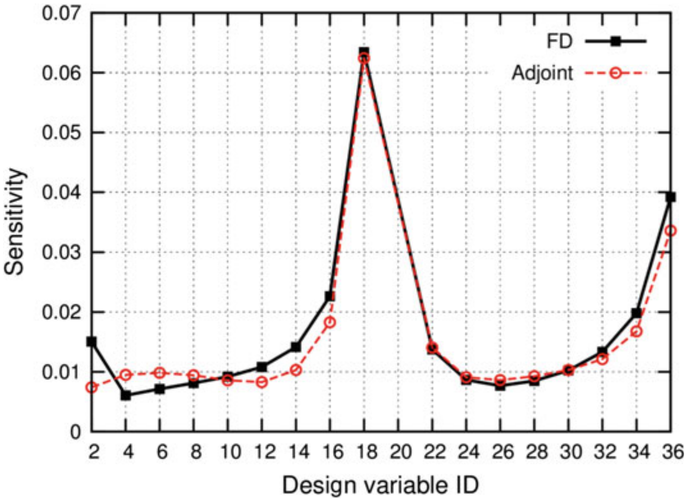


Fig. 10.6 Pitching isolated airfoil. Comparison of the time-averaged lift sensitivities for some control points, using the proposed adjoint method and FD

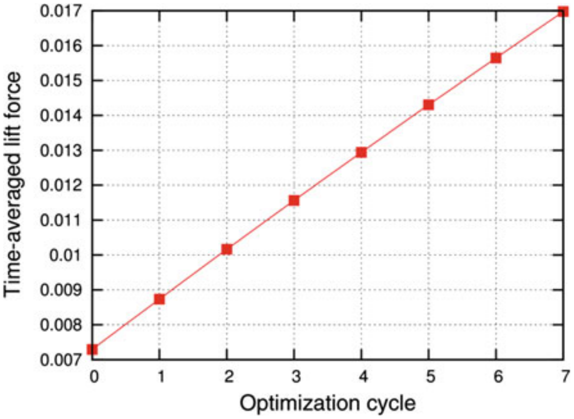


Fig. 10.7 Optimization of a pitching isolated airfoil (target lift). Evolution of the time-averaged lift force during the optimization loop

the pitching RAE2822 isolated airfoil presented in Sect. 10.4.2. The airfoil pressure and suction sides are parameterized using two Bezier curves, with 20 control points each, which are free to move in the y direction. Since the first and last control points are fixed, this case has 36 design variables. Figure 10.6 shows a good agreement between the gradients of the time-averaged lift force obtained by the two methods. Then, the so-computed adjoint sensitivities are used to run a shape optimization loop. Figure 10.7 shows the gradual increase in the lift force from its initial value after 7 optimization cycles, by changing the shape basically at the trailing edge, Fig. 10.8.

Next, the optimization framework is used for aeroacoustic noise reduction. Starting geometry and flow conditions are the same as in the lift maximization problem,

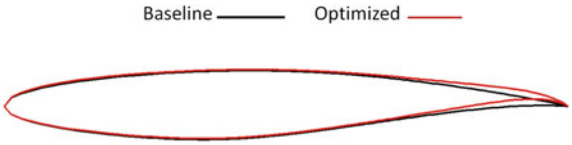


Fig. 10.8 Optimization of a pitching isolated airfoil (target lift). Shapes of the baseline and optimized airfoils

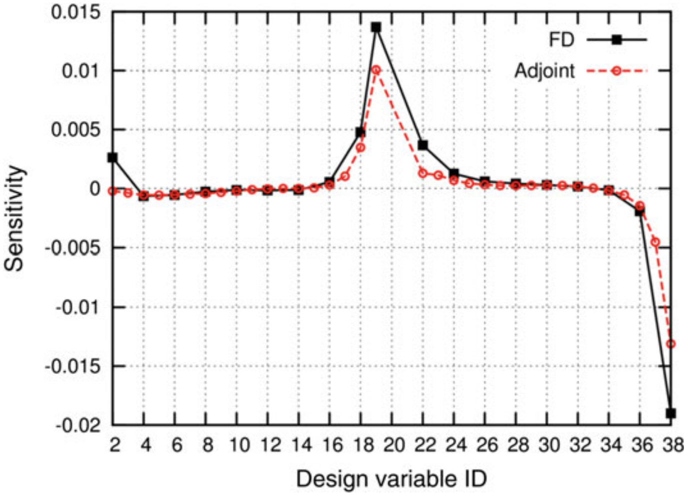


Fig. 10.9 Lift-constrained aeroacoustic optimization of a pitching isolated airfoil. Comparison of the noise (J as in Eq. 10.8) sensitivities for some control points, using the proposed adjoint method and FD

by minimizing the objective function of Eq. 10.8. In this case, a lift constraint is additionally imposed using a gradient projection method based on a deferred correction scheme. The receiver is located at $\vec{x}_o = (0, -20C)$. To verify the computed gradients using the adjoint solver, these are compared with those obtained by FD in Fig. 10.9. It shows a good agreement between the gradients obtained by the two methods. There are discrepancies at the trailing and leading edge areas; however, even for those control points, the gradients obtained by the two methods have the same signs.

Then, the adjoint-based shape optimization takes place. As illustrated in Fig. 10.10, after 18 design cycles, the noise objective function, Eq. 10.8, is reduced by more than 60%. This figure also shows that the proposed constraint imposition method with the deferred correction keeps the time-averaged lift value almost constant, as it changes about 3% at the end. As expected, the reduction in the objective value results in a lower amplitude in pressure fluctuations, as shown in Fig. 10.11. Figure 10.12 compares the directivity plot of the baseline and the optimized airfoils at the radius of R=20C and shows that the reduction in noise is omnidirectional. Figure 10.13 com-

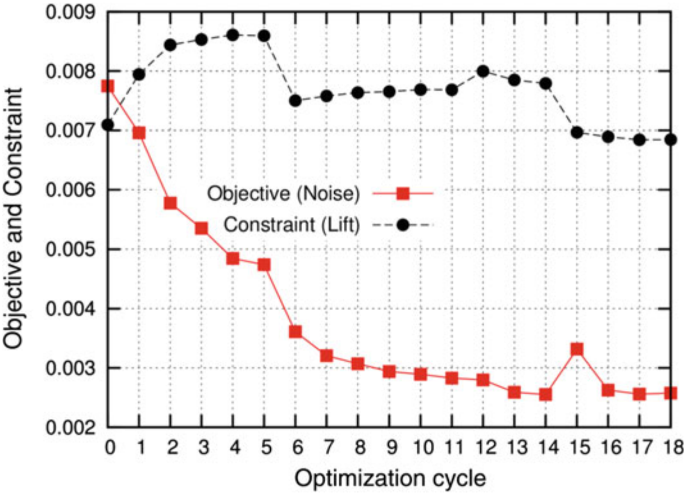


Fig. 10.10 Lift-constrained aeroacoustic optimization of a pitching isolated airfoil. Convergence of the objective and constraint functions

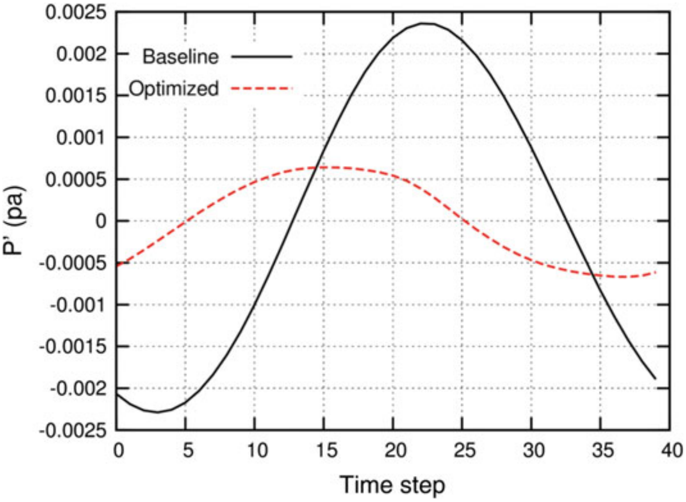


Fig. 10.11 Lift-constrained aeroacoustic optimization of a pitching isolated airfoil. Time history of pressure fluctuation within a period at the receiver's location

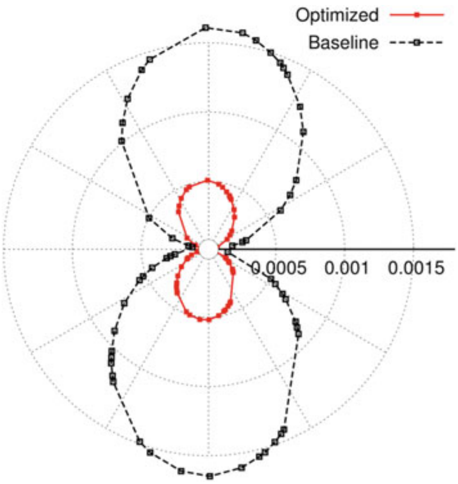


Fig. 10.12 Lift-constrained aeroacoustic optimization of a pitching isolated airfoil. Comparison of the directivity plots of the baseline and optimized airfoils at R=20C

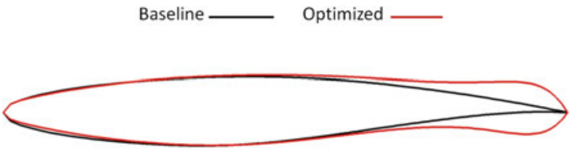


Fig. 10.13 Lift-constrained aeroacoustic optimization of a pitching isolated airfoil. Shapes of the baseline and optimized airfoils

pares the baseline and the optimized airfoil shapes. It shows that the airfoil's shape is changed mainly close to the trailing edge while the rest of it remains almost intact. This practically reconfirms the important role of the airfoil trailing edge shape on noise generation.

10.6 Conclusions

The in-house flow/adjoint solver is extended to include an aeroacoustic noise prediction tool and its adjoint counterpart, based on the permeable version of the FW-H analogy in the frequency domain. The design sensitivities obtained by the continuous adjoint method are verified versus FD, for the noise objective function and the time-averaged lift for a pitching airfoil. Then, aerodynamic and aeroacoustic shape optimization is performed and the results show that the objective values are significantly improved. The aeroacoustic optimization is subjected to an equality constraint on the lift. Results of the aeroacoustic optimization highlighted the importance of the trailing edge shape in airfoil self-noise generation.

Acknowledgments This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska Curie Grant Agreement No 722401.

References

1. Kallas S et al (2011) Flightpath 2050 Europe's vision for aviation. Report of the high level group on aviation research. European commission, Brussels, Belgium, Report No. EUR 98
2. Pironneau O (1974) On optimum design in fluid mechanics. J Fluid Mech 64(1):97–110
3. Jameson A (1988) Aerodynamic design via control theory. J Sci Comput 3(3):233–260
4. Papoutsis-Kiachagias EM, Giannakoglou KC (2016) Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. Arch Comput Methods Eng 23(2):255–299
5. Papoutsis-Kiachagias E, Magoulas N, Mueller J, Othmer C, Giannakoglou K (2015) Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. Comput Fluids 122:223–232
6. Rumpfkeil M, Zingg D (2010) A hybrid algorithm for far-field noise minimization. Comput Fluids 39(9):1516–1528
7. Zhou B, Albring T, Gauger N, Economon T, Palacios F, Alonso J (2015) A discrete adjoint framework for unsteady aerodynamic and aeroacoustic optimization. In: 16th AIAA/ISSMO multidisciplinary analysis and optimization conference, p. 3355
8. Zhou BY, Albring TA, Gauger NR, Economon TD, Alonso JJ (2016) An efficient unsteady aerodynamic and aeroacoustic design framework using discrete adjoint. In: 17th AIAA/ISSMO multidisciplinary analysis and optimization conference, p 3369
9. Zhou B, Albring T, Gauger N, Ilario C, Economon T, Alonso J (2017) Reduction of airframe noise components using a discrete adjoint approach. In: 18th AIAA/ISSMO multidisciplinary analysis and optimization conference, p 3658
10. Zhou BY, Albring T, Gauger NR, Ilario da Silva CR, Economon TD, Alonso JJ (2017) A discrete adjoint approach for jet-flap interaction noise reduction. In: 58th AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference, p 0130
11. Economon T, Palacios F, Alonso J (2012) A coupled-adjoint method for aerodynamic and aeroacoustic optimization. In: 12th AIAA Aviation Technology, Integration, and Operations (ATIO) conference and 14th AIAA/ISSMO multidisciplinary analysis and optimization conference, p 5598
12. Kapellos CS, Papoutsis-Kiachagias EM, Giannakoglou KC, Hartmann M (2019) The unsteady continuous adjoint method for minimizing flow-induced sound radiation. J Comput Phys 392:368–384
13. Kampolis I, Trompoukis X, Asouti V, Giannakoglou K (2010) CFD-based analysis and two-level aerodynamic optimization on graphics processing units. Comput Methods Appl Mech Eng 199(9–12):712–722
14. Asouti VG, Trompoukis XS, Kampolis IC, Giannakoglou KC (2011) Unsteady cfd computations using vertex-centered finite volumes for unstructured grids on graphics processing units. Int J Numer Methods Fluids 67(2):232–246
15. Yoo C, Wang Y, Trouve A, Im H (2005) Characteristic boundary conditions for direct simulations of turbulent counterflow flames. Combust Theory Model 9(4):617–646
16. Lighthill MJ (1954) On sound generated aerodynamically ii. turbulence as a source of sound. Proc R Soc Lond Ser A Math Phys Sci 222(1148):1–32
17. Curle N (1955) The influence of solid boundaries upon aerodynamic sound. Proc R Soc Lond Ser A Math Phys Sci 231(1187):505–514
18. Ffowcs Williams J, Hawkings D (1969) Sound generation by turbulence and surfaces in arbitrary motion. Philoso Trans R Soc Lond Ser A Math Phys Sci 264(1151):321–342
19. Lockard D (2000) An efficient, two-dimensional implementation of the Ffowcs Williams and Hawkings equation. J Sound Vib 229(4):897–911

Chapter 11
Discrete Adjoint Approaches for CHT Applications in OpenFOAM

Markus Towara, Johannes Lotz, and Uwe Naumann

Abstract Conjugate Heat Transfer (CHT) simulations allow the prediction of complex interactions between fluid and solid mediums. Our application is the optimization of heat transfer between heat sinks and a cooling fluid, used to extract the heat from server infrastructure. Adjoint methods allow the optimization of high dimensional parameter settings, using sensitivity information. Compared to classical approaches to sensitivity generation, e.g. finite differences, a significant improvement in run time can be achieved, as the complexity of deriving the sensitivity scales with the output dimension, instead of the input (parameter) dimension. As an initial prove of concept, our discrete adjoint OpenFOAM framework has been extended to facilitate the differentiation of the `chtMultiRegionSimpleFoam` solver. To combat prohibitive memory loads a traditional and a novel checkpointing approach are used. We will present results of the heat transfer of a copper heat sink immersed in water.

Keywords CHT · CFD · Algorithmic Differentiation · OpenFOAM

11.1 Introduction

Conjugate heat transfer (CHT) simulations allow the prediction of complex interactions between solids and fluids. A discussion on the history of CHT methods can e.g. be found in [1]. Previous studies with heat transfer using the continuous or adjoint method include [2–4].

M. Towara (✉) · J. Lotz · U. Naumann
RWTH Aachen, Software and Tools for Computational Engineering, Aachen, Germany
e-mail: towara@stce.rwth-aachen.de

J. Lotz
e-mail: lotz@stce.rwth-aachen.de

U. Naumann
e-mail: naumann@stce.rwth-aachen.de

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_11

The paper builds on our previous works [5–7] to introduce Algorithmic Differentiation (AD) into OpenFOAM [8]. AD, specifically employing operating overloading techniques [9], allows to differentiate complex (here `C++`) codes w.r.t. arbitrary input variables with great flexibility and accuracy (for a variety of applications of AD see e.g. [10]).

The outline of this paper is as follows. In Sect. 11.2 we briefly introduce the CHT problem formulation, as utilized by OpenFOAM. Further, in Sect. 11.3 we introduce the basic approaches of AD. In Sects. 11.4 and 11.5 we then focus on the implementation of checkpointing techniques for the CHT problem and how they differ from our existing implementations for singe domain solvers. Methods for identifying issues in checkpointing implementations are discussed. In Sect. 11.6 further details, required for obtaining accurate shape sensitivities, are discussed. Sect. 11.7 will introduce a CHT case of a copper heat sink immersed in water, and presents sensitivity results. In Sect. 11.8 we present an alternative checkpointing approach, which does not rely on the manual identification of states and instead utilizes the primal copy constructors.

11.2 CHT Foundations

The CHT problem is characterized by the discretization and solution of multiple PDEs on different domains. In the fluid domain, the incompressible Navier-Stokes equations, including the momentum (11.1), mass (11.2), and energy conservation (11.3) equations are solved. In OpenFOAMs `multiRegionSimpleFoam` solver this is achieved by discretizing the equations using the finite volume method (FVM) and applying the SIMPLE algorithm, implicitly coupling the pressure to the velocities [11].

On the solid domain, the energy equation simplifies to the less complex Poisson equation (11.4), that can be solved to predict the temperature distribution within the solid. The governing equations are outlined below, for details and how they are implemented and discretized within OpenFOAM see [12].
Fluid domain:

$$(\mathbf{u} \otimes \boldsymbol{\nabla})\,\mathbf{u} = \nu \boldsymbol{\nabla}^2 \mathbf{u} - \frac{1}{\rho} \boldsymbol{\nabla} p + \mathbf{b}\,, \tag{11.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0\,, \tag{11.2}$$

$$\boldsymbol{\nabla} \cdot \left(\rho c_p \mathbf{u} T\right) = \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) + \dot{q}_F\,. \tag{11.3}$$

Solid domain:

$$k \boldsymbol{\nabla}^2 T = -\dot{q}_S\,. \tag{11.4}$$

Here $\mathbf{u}$ denotes velocity, p pressure, ρ fluid density, ν kinematic viscosity, T temperature, c_P specific heat capacity, k heat conductivity, $\dot{q}_F$ external heat fluxes into the fluid domain, and $\dot{q}_S$ external heat fluxes into the solid domain.

Assuming a negative temperature gradient between solid and fluid, the fluid convects heat energy away from the solid surface, thus effectively cooling the boundary and interior of the solid domain. The solution of the Navier-Stokes and Poisson equation are only loosely coupled, that is both equations are discretized and solved for independently and are only coupled by the shared temperature boundary conditions. This helps with stability and reduces complexity of individual simulation steps, but it can lower the overall convergence rate. In our experience, the under-relaxation factor for the temperature in the solid domain can be chosen close or equal to one, greatly improving convergence of the solid temperature field.

The interface between solid and fluid regions can either be a conforming mesh, where both regions share the same patch with identical boundary faces (with flipped normals) or a non-conforming mesh with incompatible boundary faces. In this case the values can be interpolated from the fluid to solid patch and vice-versa. Both cases can be differentiated by AD without modifications, however the interpolation adds a non-trivial amount of computational work.

11.3 Algorithmic Differentiation

We consider the optimization problem $J(\mathbf{x})$ for $J : I\!R^n \rightarrow I\!R$, where each function evaluation $J(\mathbf{x})$ comprises the solution of the discrete Navier-Stokes equations and the coupled heat equations, forming a very large system of parameterized nonlinear equations. First-order AD assumes J to be at least once continuously differentiable at all points of interest. For a given implementation of the primal objective $y = J(\mathbf{x})$, a corresponding (first-order) adjoint code computes

$$\bar{\mathbf{x}} = \bar{J}(\mathbf{x}, \bar{y}) \equiv \nabla J^T \cdot \bar{y},$$

where $\bar{\mathbf{x}} \in I\!R^n$ and $\bar{y} \in I\!R$ are the adjoints of $\mathbf{x}$ and y respectively. Using the adjoint mode of AD, the gradient can be obtained at a computational cost of $O(1) \cdot Cost(J)$, where $Cost(J)$ denotes the computational cost of a single evaluation of J. The actual run time factor depends on various parameters, including the mode of differentiation (continuous vs. discrete adjoint), the expertise of the adjoint code developer, and the quality of the AD software tool, if one is used. For reference, the computational cost to compute the same gradient using finite differences or the tangent mode of AD is $O(n) \cdot Cost(J)$. For our CHT applications we use the adjoint model, as typically a very large number of inputs are mapped onto a single output. Conceptually, AD is based on the fact that the given implementation of the primal objective as a computer program can be decomposed at run time into a *single assignment code*.

$$\begin{aligned}&\text{for } j = n, \dots, n+p\\ &v_j = \varphi_j(v_i)_{i \prec j},\end{aligned}$$

where $i \prec j$ denotes a direct dependence of the variable v_j on v_i. The result of each *elemental function* φ_j is assigned to a unique auxiliary variable v_j. The n *independent inputs* $x_i = v_i$, for $i = 0, \dots, n-1$, are mapped onto the *dependent output* $y = v_{n+p}$. The values of p *intermediate variables* v_k are computed for $k = n, \dots, n+p-1$.

The primal code is augmented with instructions for storing data which is required for the reversal of the data flow and for the computation of the local partial derivatives $\frac{\partial \varphi_j}{\partial v_i}$, for $j = n, \dots, n+p$ and $i \prec j$. A data structure commonly referred to as *tape* is used for this purpose. This *(augmented) forward section* of the adjoint code is succeeded by the *reverse section* propagating adjoints $\bar{v}_i$ for all v_i in reverse order, that is, for $i = n+p-1, \dots, 0$:

$$\begin{aligned}&\left.\begin{aligned}&\text{for } j = n, \dots, n+p+m-1\\ &\quad v_j = \varphi_j(v_i)_{i \prec j}\end{aligned}\right\}\text{forward section}\\ &\left.\begin{aligned}&\text{for } i = n+p-1, \dots, 0\\ &\quad \bar{v}_i = \sum_{j: i \prec j} \frac{\partial \varphi_j}{\partial v_i} \cdot \bar{v}_j\end{aligned}\right\}\text{reverse section}\end{aligned} \tag{11.5}$$

Note that the v_j computed in the forward section are potentially required as arguments of local partial derivatives within the reverse section. They are read in reverse with respect to the original order of their evaluation. The additional persistent memory requirement of the adjoint code becomes $O(n+p)$. The efficient reversal of the data flow is among the main challenges in adjoint AD. It is responsible for black-box adjoint AD typically not being applicable to large-scale numerical simulations. The available persistent memory may simply not be large enough [13]. For our implementation we use the tape based AD tool dco/c++ [14], which implements an operator overloading approach of AD, as opposed to source code transformation.

11.4 Checkpointing Considerations

Checkpointing is an important technique to reduce the memory demands of the adjoint mode of AD by trading memory against run time [9]. Only parts of the program are adjoined at a time, a previous state is then restored from a checkpoint and a different part of the program is taped and adjoined. Let $\mathbf{x}^i$ be the state at an iteration step i. E.g. for the incompressible laminar Navier-Stokes equations the state is the combination of velocity, pressure and face flux fields $\mathbf{x} = (\mathbf{U}, \mathbf{p}, \boldsymbol{\phi})$. The general procedure to adjoin a single iteration step f^i, transforming state $\mathbf{x}^i$ into $\mathbf{x}^{i+1}$ can be formalized as follows. We assume at least one checkpoint at $\mathbf{x}^0$ is available. We further assume that the adjoints $\mathbf{x}^{i+1}$ are already known from previous applications of the procedure.

- Restore state $\mathbf{x}^j$ where $j \leq i$ and $\min\limits_{c_j \in C} (i - j)$;
- If $j < i$ passively recalculate $\mathbf{x}^i$;
- Register state $\mathbf{x}^i$ as inputs. If no other statements are executed in this step, this has the added benefit, that the adjoints $\bar{\mathbf{x}}^i$ will be located in memory contiguously, once they are calculated;
- Calculate and tape iteration step i: $\mathbf{x}^{i+1} = f^i(\mathbf{x}^i)$;
- Register state $\mathbf{x}^{i+1}$ as outputs. If no other statements are executed in this step, this again has the benefit, that the adjoints $\bar{\mathbf{x}}^{i+1}$ can be written to memory contiguously;
- Restore previously calculated adjoints $\bar{\mathbf{x}}^{i+1}$ into the tape;
- Interpret tape, calculating $\bar{\mathbf{x}}^i = \left(\frac{\partial f(\mathbf{x}^\mathbf{i})}{\partial \mathbf{x}^\mathbf{i}}\right)^T \cdot \bar{\mathbf{x}}^{i+1}$ and $\bar{\boldsymbol{\alpha}} = \bar{\boldsymbol{\alpha}} + \left(\frac{\partial f(\mathbf{x}^\mathbf{i})}{\partial \boldsymbol{\alpha}}\right)^T \cdot \bar{\mathbf{x}}^{i+1}$;
- Extract calculated adjoints $\bar{\mathbf{x}}^i$ from tape;
- Reset tape.

This procedure can be repeated until all iteration steps have been adjoined and all desired adjoints $\bar{\boldsymbol{\alpha}}$ have been accumulated.

Compared to the checkpointing procedure already outlined in [5], the complexity is increased for CHT applications in OpenFOAM by the following: Firstly the mesh is decomposed into multiple regions, corresponding to solid and fluid phases. Secondly, the CHT implementation and case setup uses boundary conditions not previously studied in the context of our discrete adjoint implementation. Two of these offending boundary conditions are outlined below. The `fixedFluxPressure` condition for `p_rgh` inherits from the `fixedGradient` boundary condition. Thus the boundary field on patches declared with the `fixedFluxPressure` are of type `fixedGradientFvPatchField`. The `fixedGradientFvPatchField` class declares a private data member `Field<Type> gradient_`, storing the surface normal (pressure) gradient. This is easily overlooked, as the gradient is private to the specific implementation of the boundary condition and is not part of the general `fvPatchField` boundary condition it inherits from. The `fixedFluxPressure` boundary condition iteratively updates the gradient, making the gradient part of the state. Thus, it needs to be checkpointed. The same principle applies to the `mixedFvPatchField` class, that is utilized by the `inletOutlet` boundary condition. The `inletOutlet` condition locally switches between the fixed value and fixed gradient boundary condition, depending on flow direction. It is commonly used to prohibit backflow. Similarly, the `mixedFvPatchField` class stores a private `scalarField volumeFraction_`, which in the context of `inletOutlet` switches between a fixed gradient and fixed value. If this field is not checkpointed, wrong primals are calculated during the repeated passive evaluations.

Table 11.1 lists the quantities that were identified as being part of the state and need to be checkpointed for the `chtMultiRegionSimpleFoam` solver using the `kOmegaSST` turbulence model. This is basically a complete list of OpenFOAMs `IOobject` registry with some additional quantities.

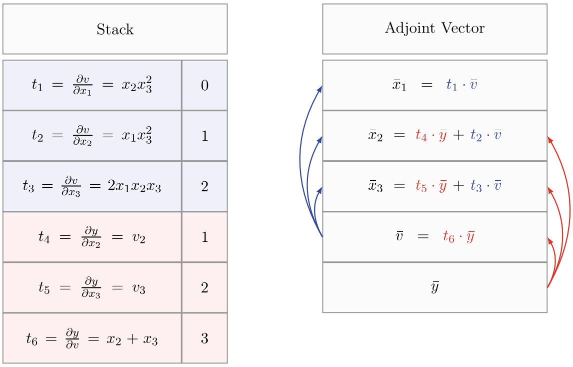
Our checkpointing interface needs the possibility to advance the iteration state one step at a time (from a given state). Previously this was achieved by holding references to all fields (locally) created in the OpenFOAM solvers `main()` routine in a separate class structure. This involves a lot of code duplication and additional work to adopt the checkpointing procedure to different solvers. Therefore, we recently switched to an implementation where the iteration step is captured in an C++11 lambda expression, which allows to explicitly or implicitly capture the variables local to the main routine. By wrapping the created lambda function into a `std::function<T>` structure it can be passed to the checkpointing interface. Thus, the simulation state can be advanced whenever necessary by calling the created function. As checkpointing schemes we support Revolve [15] and a simple equidistant scheme.

As stated earlier, the interpolation between different meshes can add a significant overhead to the required tape memory. For a static (non-moving) mesh the interpolation coefficients are constant. However, the interpolation is currently recorded in the tape during each iteration step. The calculation of the adjoints of the interpolation can potentially be handled more efficiently using automatic or manual local pre-accumulation [9].

11.5 Verifying the Checkpointing Implementation

A robust checkpointing implementation is important, as it also builds the foundation for our more advanced reverse accumulation [16] and piggybacking [9] solvers. For the verification of the correctness of the checkpointing implementation and easier identification of issues, we implemented three different debug modes for our AD tool `dco/c++`. Besides allowing to find issues in the current cases, these modes can also help to prevent future problems. They can identify assignments which not yet actively influence the numerics, but might become relevant for different activity paths. The modes are described below and illustrated with brief examples.

Fig. 11.1 Conceptual tape layout of the stack and adjoint vector for the program $v = x_1 \cdot x_2 \cdot x_3;\ y = (x_2 + x_3) \cdot v$ [7]



Stack
$t_1 = \frac{\partial v}{\partial x_1} = x_2 x_3^2$ 0
$t_2 = \frac{\partial v}{\partial x_2} = x_1 x_3^2$ 1
$t_3 = \frac{\partial v}{\partial x_3} = 2x_1 x_2 x_3$ 2
$t_4 = \frac{\partial y}{\partial x_2} = v_2$ 1
$t_5 = \frac{\partial y}{\partial x_3} = v_3$ 2
$t_6 = \frac{\partial y}{\partial v} = x_2 + x_3$ 3
Adjoint Vector
$\bar{x}_1 = t_1 \cdot \bar{v}$
$\bar{x}_2 = t_4 \cdot \bar{y} + t_2 \cdot \bar{v}$
$\bar{x}_3 = t_5 \cdot \bar{y} + t_3 \cdot \bar{v}$
$\bar{v} = t_6 \cdot \bar{y}$
$\bar{y}$

Figure 11.1 shows the conceptual tape layout for a simple example code with two assignments. For each assignment an entry in the stack is created, storing the partial derivatives w.r.t. the variables on the right hand side of the assignment, as well as pointers to the location in the adjoint vectors which need to be incremented by the product between the partial derivative and the incoming adjoints during the tape interpretation. Note how edges in the tape always point upwards, propagating adjoint information backwards through the tape. For a more complete discussion of the tape implementation see [7, 17].

The debug modes can e.g. be applied to the iteration loop, to check if any part of the current step depends on any variables outside of the state. If this is the case, there is dependence on data which might not be correct once the chronological order of iterations is broken for the recomputation of states. The challenge in essence is to capture the full state necessary to accurately recompute future states, without storing unneeded intermediate values. For a complex code, as OpenFOAM, this is not a trivial task, exemplified by the amount of fields listed in Table 11.1.

Overwrite barrier: After each assignment into a floating point variable previously known to the tape, its associated tape index increases in order to handle the name aliasing of the variable. After an *overwrite* barrier is introduced, variables that were defined before the barrier are not allowed to be overwritten. An exception is raised if such a variable is modified. Whenever a variable with global scope is modified within the iteration, it has to become part of the checkpointed state, else it will not be restored to its correct value when a previous checkpoint is loaded. By placing the barrier in front of the iteration it can be used to determine if variables not part of the current iteration or state are overwritten.

Forward barrier: A window in the adjoint vector is declared, to which no partial edges are allowed to point. This means that the primal variables corresponding to these tape positions are not allowed to occur on the right hand side of an assignment. To allow the adjoint accumulation of global parameters, the corresponding tape entries can be moved to before the window. These parameters must not be overwritten during the iteration phase. The barrier is enforced during the (augmented) *forward* evaluation of the code.

Reverse barrier: As forward barrier, but takes the actual dependencies of the outputs into account. This is done by enforcing the barrier during the *reverse* interpretation phase. This helps to avoid false positives, where the desired cost functional does not actually depend on the quantity on the left hand side of the assignment. Thus, no adjoints will ever be propagated along the offending edges, producing a false positive in forward barrier mode.

The forward and reverse barriers are especially useful to debug issues with the propagation of adjoints, when the correct recomputation of primals has already been established. On the other hand, the overwrite barrier can be used when the primals of the recomputed state do not match the expected values. In order to not negatively influence the efficiency of the AD tool, the debugging capability has been implemented in a separate adjoint data type. The introduction of AD types into the OpenFOAM code base has already been discussed in [5] and in detail in [7]. Note, that

Table 11.1 Quantities that need to be checkpointed for the solid and fluid phases. All quantities are either `volScalarFields`, `volVectorFields` or `surfaceScalarFields`, with the exception of `cumulativeContErr`. The checkpoint for `cumulativeContErr` is not strictly required, as it is not connected to the parameters, however it will trip the debugging safeguards of the AD tool

Region	Type	Name
global	scalar	cumulativeContErr
fluid	volScalarField	gh
fluid	volScalarField	thermo:mu
fluid	volScalarField	alphat
fluid	volScalarField	thermo:psi
fluid	volScalarField	nut
fluid	volScalarField	yWall
fluid	volScalarField	p
fluid	volScalarField	T
fluid	volScalarField	e
fluid	volScalarField	rho
fluid	volScalarField	k
fluid	volScalarField	omega
fluid	volScalarField	p_rgh
fluid	volScalarField	thermo:rho
fluid	volScalarField	thermo:alpha
fluid	volVectorField	U
fluid	surfaceScalarField	phi
fluid	surfaceScalarField	ghf
solid	volScalarField	thermo:mu
solid	volScalarField	betavSolid
solid	volScalarField	thermo:psi
solid	volScalarField	thermo:rho
solid	volScalarField	p
solid	volScalarField	T
solid	volScalarField	thermo:alpha
solid	volScalarField	h

to enforce the barriers no actual calculation and propagation of adjoints has to take place, only dependency information is needed. Thus, this functionality is removed for the debugging type, significantly lowering the memory footprint of this type.

In addition to the mentioned barriers, another check is implemented in dco/c++, which prevents edges pointing to positions further in the tape. During normal operation such edges should never exist and in the context of checkpointing are an indication that states from a previous iteration have not correctly been identified and checkpointed.

11.6 Additional Considerations for Shape Optimization

Conceptually, the application of checkpointing remains unchanged from the case of topology optimization [5]. Compared to topology optimization, the active path through the pre-processor stage is much more complex. During mesh construction the parameters, that is the location of the individual points of the mesh (contained in the OpenFOAM primitive mesh), are used at various locations in the code to construct the CFD mesh representation. This mesh construction phase is only executed once and can not be restored from a checkpoint easily, therefore it is permanently included in the tape. Following the pre-processing phase, the tape is switched off and the usual checkpointed iteration phase begins. After all iteration steps have been adjoined, the remaining tape of the pre-processor is adjoined, yielding the adjoints of the parameters.

A naive implementation yields results that are not consistent with black-box adjoints, indicating that some dependencies are missed. Those missing dependencies have been first identified as the non-orthogonal correction vectors by manually comparing the tapes of black-box and checkpointed adjoint [7]. With the newly introduced debugging facilities the issues can be easily identified using the forward or reverse barrier technique. The reason the dependencies are missed is the presence of on demand functions in OpenFOAM. Several data fields in the mesh object are stored in dynamic memory, and are only constructed once they are first requested by their access routine.

The following access functions in the `fvMesh` class create their fields on demand:

- `C()`: Constructs the cell center vector;
- `Cf()`: Constructs the face center vector;
- `V()`: Constructs the cell volume vector;
- `Sf()`: Constructs the face area vectors;
- `magSf()`: Constructs the magnitude of face area vectors;
- `deltaCoeffs()`: Constructs delta coefficients;
- `nonOrthDeltaCoeffs()`: Constructs the non orthogonal delta coefficients;
- `nonOrthCorrectionVectors()`: Constructs the non orthogonal correction vectors.

Most of these functions are first accessed during the pre-processor phase, and thus the construction of the fields is captured by the tape. However, the non-orthogonal correction vectors are first constructed when discretizing the gradient operator in the momentum equations, using the corrected surface-normal gradient scheme. The first occurrence of this discretization is within the first SIMPLE iteration, at which point the tape has already been switched off by the checkpointing procedure, to advance the state in passive mode to the first active section. When the `nonOrthCorrectionVectors()` access function is subsequently called while the tape is active, only a reference to the field created earlier is returned. Therefore the dependence of the correction vectors on the parameters is lost.

To fix this problem, we explicitly call all on demand generator functions of the `fvMesh` instance, after the pre-processing is finished but before the tape is switched off. This might be redundant for some functions, if the field has already been initialized. However, as in that case only a reference is returned, which is subsequently ignored, the run time and memory cost of those additional calls is negligible. The actual constructors generating the data are private to the `fvMesh` class, and would require modifications inside the OpenFOAM code base in order to be accessible from our solvers. Therefore we simply trigger dummy calls to the accessor routines, which have the side effect of creating the required data fields. The changes required in order to obtain a consistent checkpointed shape adjoint are presented in Listing 11.1.

Listing 11.1 Forcing the early on demand construction of the `fvMesh` fields by calling their access routines.

```
void init_mesh(Foam::fvMesh& mesh){
  mesh.Sf();     mesh.magSf();
  mesh.C();      mesh.deltaCoeffs();
  mesh.Cf();     mesh.nonOrthDeltaCoeffs();
  mesh.V();      mesh.nonOrthCorrectionVectors();
}

int main(int argc, char *argv[])
{
  #include "createTime.H"
  #include "createMeshes.H"
  #include "createFields.H"

  for(fvMesh& solidMesh : solidRegions)
    init_mesh(solidMesh);
  forAll(fvMesh& fluidMesh : fluidRegions)
    init_mesh(fluidMesh);

  ADmode::global_tape->switch_to_passive();
  [...] // Continue w. checkpointed CHT algorithm
}
```

An example for the calculation of shape adjoints using checkpointing is presented in the following section. The same issues arise and fixes apply when using the checkpointing interface to implement reverse accumulation or piggybacking.

11.7 CHT Sensitivity Results

Figures 11.2, 11.3 and 11.4 show preliminary results for the calculation of heat transfer between a heat sink with seven fins with a draft angle of approximately 1.7°. Both domains are meshed with hex cells by `blockmesh` with conforming interfaces. The solid domain contains 129 360, the fluid domain 321 552 cells. The bottom patch of the solid domain (with material properties of copper) is held at a constant temperature of 375 K. The fluid (with material properties of water) enters the domain with a constant velocity of 0.05 m/s and temperature of 300 K. All exterior walls are assumed to be adiabatic. The heat transfer between solid and fluid domain is modeled with OpenFOAMs `turbulentTemperatureCoupledBaffleMixed` model.

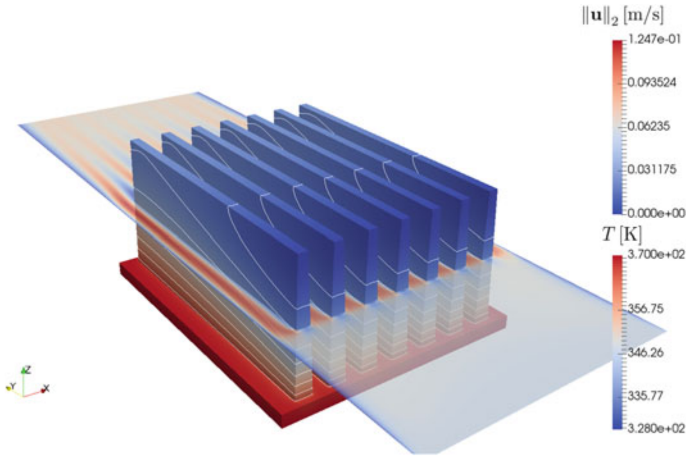


Fig. 11.2 Temperature distribution T on the solid surface with temperature isolines (white). Velocity magnitude distribution in the fluid domain on a z-normal slice

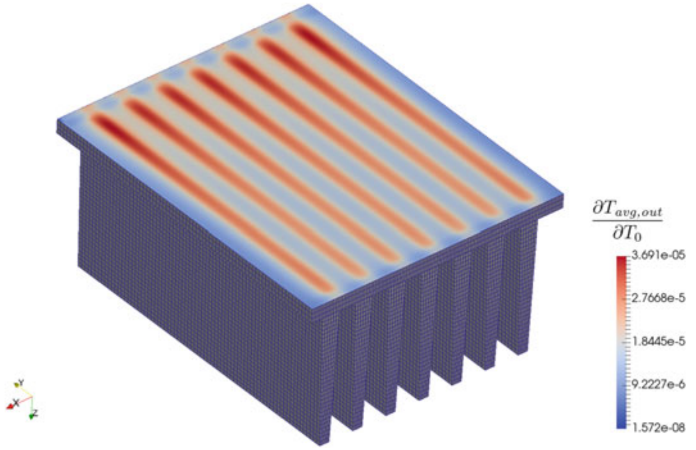


Fig. 11.3 Sensitivity of the average outlet temperature w.r.t. the temperature on the heated bottom wall of the solid domain

The flow fields and the temperature on the solid is initialized by running 400 (passive) iteration steps of `chtMultiRegionSimpleFoam`. At this point the simulation has mostly converged. We then run 20 additional iteration steps of our `adjointChtMultiRegionCheckpointingSimpleFoam` solver to obtain sensitivities.

Taping one iteration step of the shape sensitivity problem (using efficient symbolic differentiation of linear solvers) takes roughly 52 GB of tape memory, while one step of the temperature sensitivity problem takes only 23 GB. The higher demand of the shape sensitivity is due to the additional complexity caused by the differentiation of the OpenFOAM mesh representation.

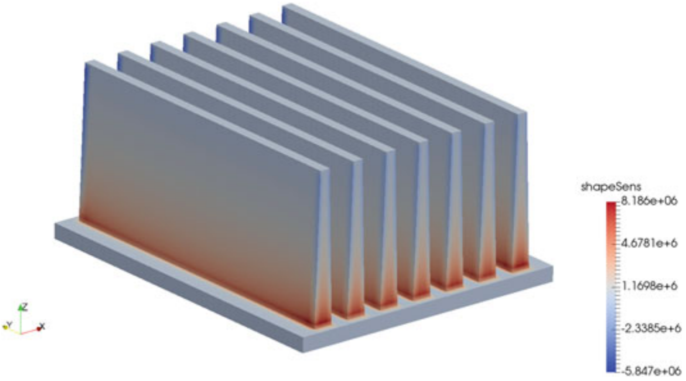


Fig. 11.4 Surface sensitivity on the solid to fluid interface

Figure 11.2 shows the temperature distribution on the surface of the solid, as well as a slice through the fluid domain, showing the velocity distribution. As the cost function we choose the average temperature on the outlet patch, as calculated by `gAverage(T)`. The dependence of the average temperature on the temperature distribution at the heated wall is depicted in Fig. 11.3. Red regions therefore indicate where the heat energy is best transported from the bottom plate to the fluid.

For the same case, Fig. 11.4 shows the shape sensitivities of the average temperature at the outlet w.r.t. movement of the surface mesh points in surface normal direction. Red regions indicate where the cross section of the fins should shrink, making them narrower, blue regions (with negative sign) indicate where the cross section should be expanded.

11.8 Primal Copy Constructor Checkpointing

As outlined previously, manually identifying the full state required to create accurate checkpoints is a tedious and error prone process. Assuming that the primal copy constructors within a code, OpenFOAM in our case, are implemented correctly, an alternative checkpointing procedure can be implemented, relying solely on the existing copy constructors and the treatment of elemental (floating point) value copies by the AD tool.

Normally, an efficient AD tool will identify assignments that can be optimized out (partial derivatives of one, assignments involving passive right hand sides,...) [18]. To obtain clearly separated windows in the adjoint vector containing only the adjoint inputs and outputs of a specific iteration we temporarily disable these kinds of optimizations. Now, whenever a floating point variable of active type is overwritten, (to avoid memory aliasing), it gets assigned a new (increasing) tape index, corresponding to an entry in the adjoint vector.

Obviously we can recreate a primal state by overwriting all checkpoint objects with copies from an earlier point in the iteration history. However, if we overwrite all objects we want to checkpoint by themselves, the copy constructors of the individual data members will also create a contiguous window in the adjoint vector, where all desired adjoints will be located after interpretation. We identify such a window by its first index and the number of contained elements.

The modified checkpointing interface has to provide the following functionality:

- Store and restore primal values by copying to/from a temporary object;
- Create a window in the adjoint vector corresponding to the inputs of the state by fully overwriting object with a copy of itself;
- Create a window in the adjoint vector corresponding to the outputs of the state by fully copying object to a temporary object;
- Extract adjoint values from the input window of the adjoint vector and store them in a contiguous vector;
- Restore adjoint values from contiguous vector into the output window of the adjoint vector.

All functionality is implemented purely by using primal operations and by accessing individual elements of the adjoint vector (without necessarily knowing which object they belong to). A memory overhead is introduced by storing a temporary copy of the object. This copy is required as to not trigger self assignment optimizations within the copy constructors. Further, the n primal checkpoints are stored as a copies of the full object, containing active types, instead of just storing the (passive) floating point values of the object.

Listing 11.2 outlines the implementation of the templatized `CheckpointObject` class, which can hold primal copies of arbitrary objects that implement a copy constructor. Due to an implementation detail of flow fields in OpenFOAM the `CheckpointObject` class has to be overloaded to use the custom `operator==` operator instead of `operator=` to copy flow fields including its boundary values.

Listing 11.2 CheckpointObjectGeometricField

```
template<typename T>
struct CheckpointObject : public CheckpointObjectBase
{
  T& objRef; // reference to the object to be checkpointed
  std::vector<T> objCheck; // primal checkpoints
  T objCopy; // temporary copy

  CheckpointObject(T& obj, const int n)
    : objRef(obj), objCopy(obj), objCheck(n,obj) {}
  void replaceCheckpoint(const unsigned i){
      objCheck[i] = objRef;
  }
  void restoreCheckpoint(const unsigned i){
      objRef = objCheck[i];
  }
  void copyToTemporary(){
    objCopy = objRef;
  }
  void copyFromTemporary(){
    objRef = objCopy;
  }
};
```

Listing 11.3 shows how to utilize the `CheckpointObject` class to mimic the behaviour of the tape operations `register_input()` and `register_output()` (Fig. 11.5).

Listing 11.3 CheckpointObjectGeometricField

```
struct CheckpointContainer{
  std::vector<CheckpointObjectBase*> checkpoints_;
  label inputStateStartIndex_;
  label outputStateStartIndex_;
  std::vector<double> adjoints_;

  void registerInputs(){
    for(const auto& c : checkpoints_)
      c->copyToTemporary();
    ADmode::global_tape->switch_to_active();
    inputStateStartIndex_ = ADmode::global_tape->get_position().index() + 1;
    ADmode::global_tape->varied_analysis() = false;
    for(const auto& c : checkpoints_)
      c->copyFromTemporary();
    ADmode::global_tape->varied_analysis() = true;
    label p2 = ADmode::global_tape->get_position().index();
    n_ = p2-p1;
  }
  void registerOutputs(){
    label p1 = ADmode::global_tape->get_position().index();
    outputStateStartIndex_ = p1+1;
    ADmode::global_tape->varied_analysis() = false;
    for(const auto& c : checkpoints_)
      c->copyToTemporary();
    ADmode::global_tape->varied_analysis() = true;
    label p2 = ADmode::global_tape->get_position().index();
    n_ = p2-p1;
  }
  void storeAdjoints(){
    adjoints_.resize(n_);
    for(label i=0; i<n_; i++){
      adjoints_[i] = ADmode::global_tape->_adjoint(i+inputStateStartIndex_);
  }
  void restoreAdjoints(){
    for(label i=0; i<n_; i++)
      ADmode::global_tape->_adjoint(i+outputStateStartIndex_) = adjoints_[i];
  }
};
```

In Table 11.2 we present the run time and memory consumption for the *angled duct* case described in [7]. This is a case without heat transfer, however the presented checkpointing procedure can be readily converted to our CHT solvers. The case consists of 32 500 cells (refinement level 50). Timed are 50 iterations with five checkpoints, which are used to perform the reversal using Revolve. Timings and peak memory (memory consumption is sampled and thus not completely deterministic) are averaged over five executions. For the coarse case run time is not influenced discernibly, for the finer case the run time is even slightly improved. At this point we don't have a convincing explanation for the improved run time behavior, as more data should be copied than before. Possibly the old checkpointing implementation could be improved for efficiency. In both cases the memory demand is increased by around 5% for the copy checkpointing. For challenging applications, where the

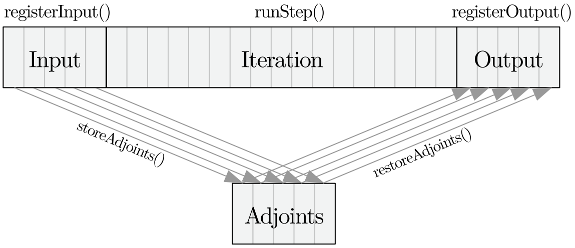


Fig. 11.5 Adjoint vector layout created by the calls to `registerInputs()`, `runStep()` and `registerOutputs`. The adjoints of the input variables are stored in a temporary array and can be restored into the outputs of a previous iteration

Table 11.2 Run time and memory consumption for the *angled duct* test case. Coarse case with 32 500 cells and finer case with 119 808 cells

Solver	Run time (s)	Peak memory (MB)
Regular checkpointing (lvl50)	149.2	1001
Primal copy checkpointing (lvl 50)	146.6	1047
Regular checkpointing (lvl 96)	457.8	3414
Primal copy checkpointing (lvl 96)	441.8	3564

classical checkpointing approach is not straightforward to implement, we deem the run time penalty to be acceptable for the simplified implementation.

11.9 Summary and Outlook

We demonstrated the applicability of a discrete adjoint framework implemented in OpenFOAM to complex CHT cases. Emphasis was placed on the correct checkpointing treatment of the states in the solid and fluid domains. While, utilizing the presented debugging tools, the manual treatment of the conflicting boundary conditions is possible, it is desirable to obtain a more robust implementation that relies on the already existing primal copy constructors of OpenFOAM. Such an approach was briefly presented and will be incorporated into future adjoint solvers. As part of our ongoing research, we plan to apply our discrete adjoint CHT framework to a variety of different heat sink geometries.

Acknowledgments This work is part of ZIM project *Entwicklung optimierter Kühlgeometrien mittels adjungierter Simulationsmethoden für die Direkt-Heißwasserkühlung von Rechenzentren* (Development of optimized cooling geometries for hot watercooling of data centers using adjoint simulation methods). It is supported by the Federal Ministry for Economical Affairs and Enegy (BMWi), on the basis of a decision by the German Bundestag.

References

1. Dorfman AS (2009) Conjugate problems in convective heat transfer. CRC Press
2. Zeinalpour M, Mazaheri K, Kiani K (2016) A coupled adjoint formulation for non-cooled and internally cooled turbine blade optimization. Appl Therm Eng 105:327–335
3. Kontoleontos EA, Papoutsis-Kiachagias EM, Zymaris AS, Papadimitriou DI, Giannakoglou KC (2013) Adjoint-based constrained topology optimization for viscous flows, including heat transfer. Eng Opt 45(8):941–961
4. Burghardt O, Gauger NR, Economon TD (2019) Coupled adjoints for conjugate heat transfer in variable density incompressible flows. In: AIAA Aviat. 2019 Forum, p 3668
5. Towara M, Naumann U (2013) A discrete adjoint model for OpenFOAM. Procedia Comp Sci 18(0):429–438; Int Conf Comp Sci
6. Towara M, Schanen M, Naumann U (2015) MPI-parallel discrete adjoint OpenFOAM. Procedia Comp Sci 51:19–28; Int Conf Comp Sci
7. Towara M (2019) Discrete Adjoint Optimization with OpenFOAM. Dissertation, RWTH Aachen University
8. OpenFOAM Ltd, OpenFOAM—The Open Source Computational Fluid Dynamics (CFD) Toolbox. http://openfoam.com/
9. Griewank A, Walther A (2008) Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM
10. Christianson B, Forth SA, Griewank A (2018) editors: advances in algorithmic differentiation. Opt Meth Soft 33(4–6):671–671. https://doi.org/10.1080/10556788.2018.1486553
11. Patankar SV, Spalding D (1972) A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. Int J Heat Mass Transfer 15(10):1787–1806
12. Moukalled F, Mangani L, Darwish M, et al (2016) the finite volume method in computational fluid dynamics. Springer, Berlin
13. Naumann U (2009) DAG reversal is NP-complete. J Discret Algorithms 7:402–410
14. Leppkes K, Lotz J, Naumann U (2016) Derivative Code by Overloading in C++ (dco / C++): introduction and summary of features. Technical report AIB-2016-08, RWTH Aachen University (2016)
15. Griewank A, Walther A (2000) Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Trans Math Soft 26(1)
16. Christianson B (1994) Reverse accumulation and attractive fixed points. Opt Methods Soft 3(4):311–326
17. Lotz J (2016) Hybrid approaches to adjoint code generation with dco/c++. Dissertation, RWTH Aachen University
18. Hascoët L, Naumann U, Pascual V (2005) "To be Recorded" analysis in reverse-mode automatic differentiation. Future Gener Comput Syst 21(8):1401–1417

Chapter 12
Robustness Measures for Multi-objective Robust Design

Lisa Kusch and Nicolas R. Gauger

Abstract A significant step to engineering design is to take into account uncertainties and to develop optimal designs that are robust with respect to perturbations. Furthermore, when multiple optimization objectives are involved it is important to define suitable descriptions for robustness. We introduce robustness measures for robust design with multiple objectives that are suitable for considering the effect of uncertainties in objective space. A direct formulation and a two-phase formulation based on expected losses in objective space are presented for finding robust optimal solutions. We apply both formulations to the robust design of an airfoil. Fluid mechanical quantities are optimized under the consideration of aleatory uncertainties. The uncertainties are propagated with the help of the non-intrusive polynomial chaos approach. The resulting multi-objective optimization problem is solved with a constraint-based approach, that combines adjoint-based optimization methods and evolutionary methods evaluated on surrogate models.

Keywords Multi-objective optimization · Robust design · Aerodynamic shape optimization

12.1 Introduction

Multi-objective optimization and robust design are two well-established fields of research. Especially, in engineering applications it is important to optimize for different conflicting criteria like for example cost and quality aspects. Here, the aim is to find a set of solutions that fulfill the concept of Pareto optimality. A feasible design x is Pareto optimal if it is non-dominated, i.e. there does not exist any feasible design

L. Kusch (✉) · N. R. Gauger
Chair for Scientific Computing, Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany
e-mail: lisa.kusch@scicomp.uni-kl.de

N. R. Gauger
e-mail: nicolas.gauger@scicomp.uni-kl.de

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_12

$\overline{x}$ such that $f_i(\overline{x}) \leq f_i(x)$ for every objective function f_i with $i \in \{1, \ldots, k\}$ and $f_j(\overline{x}) < f_j(x)$ for at least one $j \in \{1, \ldots, k\}$. Here, k denotes the number of objective functions. The image of the Pareto optimal set in objective space is denoted as the Pareto optimal front. We distinguish between direct Pareto approaches and scalarization approaches for multi-objective optimization. Direct Pareto approaches for multi-objective optimization try to find several Pareto optimal points at once. This can for example be realized using evolutionary approaches (e.g. NSGA-II [1]). A drawback of evolutionary approaches is that there is no proof of convergence and, accordingly, there are no clear stopping criteria. Another disadvantage is the high number of function evaluations reached to find an agreeable set of solutions, which is a significant problem if the computational costs for a single evaluation are already high. In scalarization approaches, for example constraint-based methods, the problem is transformed into several single-objective optimization problems, that can be solved efficiently using hybrid methods combining gradient-based optimization methods and global search methods.

Another significant step towards realistic multi-objective design is to take into account uncertainties for finding robust optimal solutions. Robust optimal solutions are solutions, that are optimal and robust with respect to perturbations. Most of the robustness measures for multi-objective optimization are inspired by single-objective robustness definitions based on statistical quantities. We distinguish between expectation-based and variance-based measures. Also, the quantities can either be formulated as objectives or set as additional constraints. Two expectation-based measures were for example proposed by Deb and Gupta [2] and adapted for aerodynamic shape optimization [3]. Furthermore, there exist methods specifically tailored for multi-objective optimization problems. The application to evolutionary multi-objective optimization enables the use of a probability of dominance or an expected fitness function [4], or a dominance relation based on worst-case analyses [5]. For a local sensitivity analysis a local sensitivity region [6] can be used in objective space.

In Sect. 12.2 expected losses are introduced as a new measure for robustness when considering multiple objectives and two different approaches to robust optimal design are presented that both result in a multi-objective optimization problem. The constraint method for solving the multi-objective optimization problems is presented in Sect. 4.3. The proposed strategy is applied for finding robust optimal solutions in aerodynamic shape optimization with aleatory uncertainties in Sect. 12.4 followed by a conclusion and an outlook in Sect. 12.7.

12.2 Robust Design

In single-objective optimization problems a solution is considered to be robust if it is not very sensitive to uncertainties. In multi-objective optimization problems the main difference to single-objective robust design is that one has to measure a combined effect of sensitivities for all objective functions. Additionally, the aim is to find a set of robust solutions instead of only a single robust solution. From this problem arises the question of how to define robust Pareto optimal designs. A new robustness measure and two corresponding formulations of robust optimal design problems are shown in the following.

12.2.1 Robustness Measures

In the following we introduce a measure for robustness, that can be used in a scalarization approach and account for effects in objective space. The general idea is to measure the expected distance of an outcome from the deterministic Pareto optimal front. Using the Pareto optimal front we can state if an outcome of random samples is better or worse (corresponding to gains and losses in objective space).

Figure 12.1 shows the contours for a fixed probability, that we will refer to as probability region, for two designs in objective space. Both designs have a similar deterministic value (black dot). Also, the variances and the expected value (circle) are similar. Nevertheless, one would prefer the left design over the right one due to the shape of the probability region. Note that for a minimization problem the points found in the lower left region are definitely better (gains) and the points in the upper right region definitely worse (losses) in comparison to the deterministic outcome. The design on the right has a large region of losses. When comparing both design it can be noticed that most outcomes of the left design dominate the outcomes of the right design. Additionally, the fact that the gains outweigh the losses for the right design shows that a robustness measure should be defined using a loss function based on the distance to the deterministic value or to the Pareto optimal front.

We propose two approaches to describe robustness with the help of losses in objective space. In both approaches the expected losses are constrained by a prescribed upper bound. Note that the existence of a solution then depends on the choice of the upper bound. Before introducing both approaches we present the expectation-based approach, that is most commonly used. The general multi-objective optimization without considering any uncertainties, that is often referred to as the deterministic optimization problem, can be formulated as

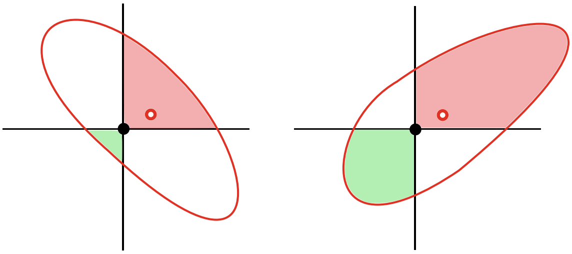


Fig. 12.1 Two different probability regions in objective space

$$\begin{aligned} \min_{y,u} \quad & \mathbf{F}(y,u) \\ \text{s.t. } & c(y,u) = 0. \end{aligned} \tag{12.1}$$

Here, the variables $y \in \mathbb{R}^d$ and $u \in \mathbb{R}^n$ are the state and design variables that fulfill the state equation $c(y, u) = 0$. The minimization of the objective function vector $\mathbf{F} : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^k$ with objective functions F_i for $i = 1, \ldots, k$ has to be understood component-wise. For notational convenience we omit any additional constraints or design variables bounds.

12.2.2 Expectation-Based Approach

For the expectation-based approach the expected value is used as a statistical quantity and a solution is denoted as robust Pareto optimal, if it is a solution to the problem

$$\begin{aligned} \min_{y,u} \ & \mathrm{Exp}(\mathbf{F}(y,u,z(\omega))) \\ \text{s.t. } & c(y,u,z(\omega)) = 0. \end{aligned} \tag{12.2}$$

In the above equation $z(\omega)$ is a realization of the vector of random input variables for uncertainties $\omega \in \Omega$. The expectation operator is to be understood component-wise. When additional constraints are present it is common to demand for constraint satisfaction for every realization z to obtain reliable designs.

12.2.3 Two-Phase Approach

In the two-phase approach we assume that a given set of Pareto optimal points has been determined for the deterministic optimization problem (12.1) in a first phase. Additionally we assume to have an approximation of the Pareto optimal front in objective space, e.g. by means of splines in the two-dimensional case or by the help of other sophisticated interpolation methods for higher dimension. Note that the approximation can become non-trivial for disconnected Pareto optimal fronts, although a distance to the front can still be defined. We denote the representation of the Pareto optimal front as ϕ_0.

The expected losses can be expressed by means of a signed distance function δ, that can be defined by using a level-set method with zero level set ϕ_0. The corresponding optimization problem to be solved in the second step is

$$\begin{aligned} \min_{y,u} \quad & \mathbf{F}(y,u,\bar{z}) \\ \text{s.t.} \quad & c(y,u,\bar{z}) = 0, \\ & \mathrm{Exp}(\max(0, \delta(\mathbf{F}(y,u,z(\omega)), \boldsymbol{\phi_0})) \leq \delta_{max}. \end{aligned} \tag{12.3}$$

The evaluation at $\bar{z}$ denotes the deterministic case where z is not a random variable but the value prescribed when not considering any uncertainties. The expected losses are constrained by an upper bound $\delta_{\max}$. The $\max(0, .)$-function ensures that only losses are considered. For reasons of clarity we will omit to include the dependency on y and u in the following definitions.

12.2.4 Direct Approach

In the direct approach the deterministic Pareto optimal front is not needed. Instead, the local distance of the samples to the current deterministic value for $\bar{z}$ is used to describe losses. When not considering only losses, this approach is similar to the constrained expectation-based approach [2]. Different assumptions for the local estimation of losses can be made. When considering expected possible losses we may formulate the optimization constraint as

$$\sum_{i=1}^{k} \text{Exp}(\max(0, F_i(z(\omega)) - F_i(\bar{z}))) \leq \mu_1. \tag{12.4}$$

Another assumption is to approximate the losses based on a local linear approximation of the Pareto optimal in the current deterministic outcome. The local front can then be represented as the zero level set of $\phi = \sum_{i=1}^{k} F_i(z(\omega)) - F_i(\bar{z})$. The corresponding optimization constraint is

$$\text{Exp}(\max(0, \sum_{i=1}^{k} F_i(z(\omega)) - F_i(\bar{z}))) \leq \mu_2. \tag{12.5}$$

Other expressions may be based on the expected definite losses or a better local approximation of the front (for example a convex representation for convex multi-objective problems).

Note that for gradient-based optimization the problem has to be transformed to make the constraint functions continuously differentiable. This can be done by either reformulating the problem with the help of additional variables or by approximating the maximum function, which is done in the context of the present work.

Figure 12.2 depicts the signed distance functions for the different approaches. In the two-phase approach the signed distance function is built using for example linear splines for approximating the Pareto optimal front. The expected possible losses and the linear approximation are always obtained locally for the respective deterministic outcome.

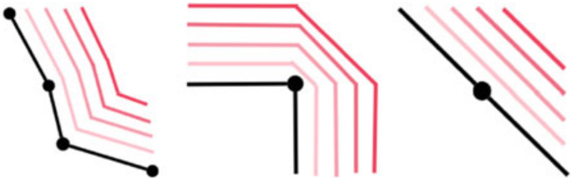


Fig. 12.2 Signed distance function for two-phase approach (left), expected possible losses (middle), linear approximation (right)

12.2.5 Uncertainty Quantification

There exist different methods to propagate uncertainties ω in the model. We make use of a non-intrusive polynomial chaos approach, which is also referred to as pseudo-spectral approach. In this approach the stochastic objective function is expanded in terms of polynomials Φ_i that are orthogonal with respect to the probability density function of the input random variables $z(\omega)$ (cp. [7]), such that

$$f(y, u, z(\omega)) = \sum_{i=0}^{\infty} \hat{f}_i(y, u)\Phi_i(z(\omega)), \tag{12.6}$$

with $\hat{f}_i(y, u) = \gamma_i^{-1}\text{Exp}(f(y, u, z(\omega))\Phi_i(z(\omega))$ and $\text{Exp}(\Phi_i\Phi_j) = \gamma_i\delta_{ij}$.

When applied to find statistical quantities the infinite expansion is truncated with m being the order of truncation. The Fourier coefficients are approximated by first using stochastic collocation with quadrature points and then employing a quadrature rule that is suitable for the used polynomials. The expected value for a normally distributed z with probability density function ρ_z, for example, reduces to

$$\begin{aligned}\text{Exp}(f(y, u, z(\omega))) &\approx \text{Exp}\left(\sum_{i=0}^{m} f_i(y, u)H_i(z(\omega))\right) = 0!f_0(y, u) \\ &= \int_{-\infty}^{\infty} f(y, u, z)H_0(z)\rho_z(z)dz \approx \sum_{k=1}^{n} f(y, u, x_k)w_k.\end{aligned} \tag{12.7}$$

Here, it was made use of the orthogonality of the Hermite polynomials $H(z)$ and that the resulting integral is approximated with a quadrature formula with weights w_k and points x_k for $k = 1, \ldots, n$. The non-intrusive polynomial chaos approach can be used for a moderate number of uncertainties. The computational effort of the quadrature can be reduced by using sparse grid quadrature rules.

12.3 Multi-objective Optimization

The formulation of robust Pareto optimal solutions results in a multi-objective optimization problem. We solve it by using a constraint-based approach. The constrained single-objective optimization problems are solved using a hybrid optimization method.

12.3.1 Constraint-Based Approach

The concept of the ε-constraint method [8] is to optimize one objective function f_{s_j} while imposing inequality constraints on the remaining competing objective functions. For the robust multi-objective optimization the constraint function is a statistical quantity. The constraints $f_i^{(j)}$ as well as the objective function f_{s_j}, that is to be optimized, are varied in the steps of the algorithm to find different Pareto optimal solutions that are evenly distributed. The resulting minimization problem for the j-th step of the algorithm applied to the general multi-objective PDE-constrained optimization problem (12.1) is

$$\begin{aligned} \min_{y,u} \quad & f_{s_j}(y,u) \\ \text{s.t.} \quad & c(y,u) = 0, \\ & f_i(y,u) \leq f_i^{(j)} \quad \forall\, i \in \{1,\ldots,k\} : \ i \neq s_j . \end{aligned} \tag{12.8}$$

The inequality constraints for the different steps are distributed equidistantly. The outlines of the front can be found by minimizing the objective functions individually without imposing additional constraints. It can be shown that all unique solutions to the resulting single-objective optimization problem (12.8) are globally Pareto optimal for any upper bound $f_i^{(j)}$ [9].

12.3.2 Global Optimization Method

The correct choice of the algorithm for solving the single-objective optimization problems (12.8) that result from the ε-constraint method is very important. In Kusch et al. [10] a hybrid algorithm is proposed for the single-objective optimization problems to enhance the chance of finding a global optimum and thus Pareto optimal points. The hybrid method combines the advantages of evolutionary and gradient-based methods. In a first step a genetic algorithm is applied on a Kriging surrogate model to avoid computationally expensive calculations. We make use of the software RoDeO [11], that is adjusted to handle the given optimization constraints. The initial data acquisition is done using Latin Hypercube sampling. The Kriging model is trained in each optimization step using adaptive sampling based on the expected improvement method. Furthermore, several designs in the direction of steepest descent are included in the training set. In the second step of the hybrid algorithm a gradient-based optimization method is applied for the full model. The design found in the first step is used as a starting point for gradient-based optimization. The gradients are obtained using a discrete adjoint method based on algorithmic differentiation. The use of accurate derivative from algorithmic differentiation is especially useful for solving constrained optimization problems.

12.4 Aerodynamic Shape Optimization

We apply the proposed method to an aerodynamic shape optimization problem for a 2D airfoil with a NACA0012 as initial design. The objective is to minimize the drag coefficient c_d and maximize the lift coefficient c_l. Additional inequality constraints are prescribed for the thickness t of the airfoil and the resulting moment c_m. The flow is transonic and inviscid with a Mach number of 0.8 and an angle of attack of 1.25. The scalar-valued uncertainties in the flight conditions are modelled by using random variables with an assumed probability density function. We assume an uncertain Mach Number, that is normally distributed such that $Ma \sim N(0.8, 0.01)$. The associated orthogonal polynomials for non-intrusive polynomial chaos are Hermite polynomials. The airfoil is parametrized with the help of 38 Hicks-Henne functions. The underlying steady Euler equations are solved with the open-source software SU2 [12] using a Jameson-Schmidt-Turkel scheme. Gradients for the optimization in SU2 are provided by algorithmic differentiation [13].

12.5 Results for the Two-Phase Approach

The two-phase approach was used with a prescribed constraint on the distance $\delta_{max} = 0.15$ in normalized objective space. The expected losses are calculated with the help of non-intrusive polynomial chaos using Gauss-Hermite quadrature with four quadrature points x_i and weights w_i for $i = 1, ..., n$. The resulting multi-objective optimization problem

$$\begin{aligned} \min_{y,u} \quad & (c_d(y, u, \bar{z}), -c_l(y, u, \bar{z}))^\top \\ \text{s.t.} \quad & c(y, u, \bar{z}) = 0, \\ & c_m(y, u, \bar{z}) \geq 0, \\ & t(u) \geq 0.12, \\ & \sum_{i=1}^{n} w_i (\max(0, \delta(\mathbf{F}(y, u, x_i), \phi_{\mathbf{0}})) \leq \delta_{max} \end{aligned} \tag{12.9}$$

is solved for eight Pareto optimal points using the ε-constraint method.

Figure 12.3 shows the optimization result in objective function space. The dots indicate the robust optimal designs evaluated for the Mach number $\bar{z} = 0.8$, that is used in the deterministic optimization. The crosses indicate the expected value. For reasons of comparison the deterministic values and the expected values of the multi-objective optimization without considering uncertainties (compare Eq. 12.1) are shown by the grey-coloured dots and crosses. The corresponding designs are plotted on the right of the figure. The upper design corresponds to the maximum lift coefficient and the lower design to the minimum drag coefficient. It can be observed that the designs are very similar, while the expected values for the robust design approach are significantly improved.

In Figure 12.4 random samples are shown for a chosen design to depict the probability region. The grey-coloured region is the probability region for a comparable design that was obtained using an expectation-based approach [3]. Here, the aim was to minimize the expected value of the drag coefficient and maximize the expected value of the lift coefficient. The probability regions differ significantly as the result obtained by the expectation-based approach leads to higher losses in objective space. In particular, the probability region based on the expected losses is close to the deterministic Pareto optimal front.

12.6 Results for the Direct Approach

The direct approach can be used if a construction of the Pareto optimal front is computationally too expensive. Exemplary, it was applied to the given test case by constraining the expected possible losses presented in Eq. (12.4) with $\mu_1 = 0.15$. The results in objective space are shown in Fig. 12.5.

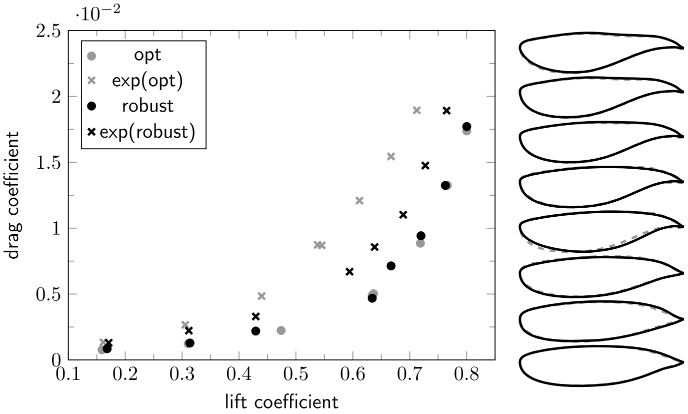


Fig. 12.3 Pareto optimal front for two-phase approach (robust) and deterministic Pareto optimal front (opt)

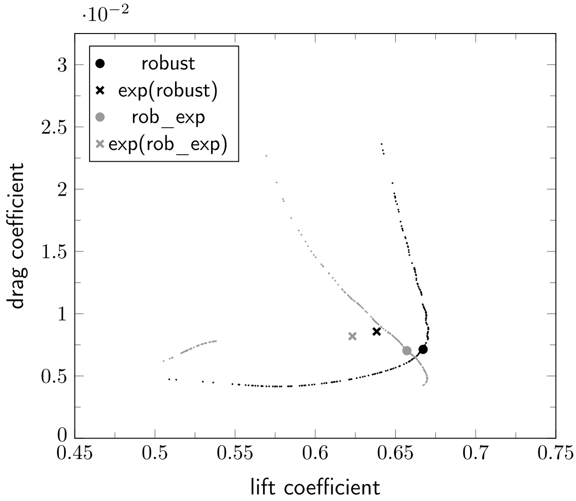


Fig. 12.4 Sampled probability region for expected losses (robust) and the expectation-based approach (rob_exp)

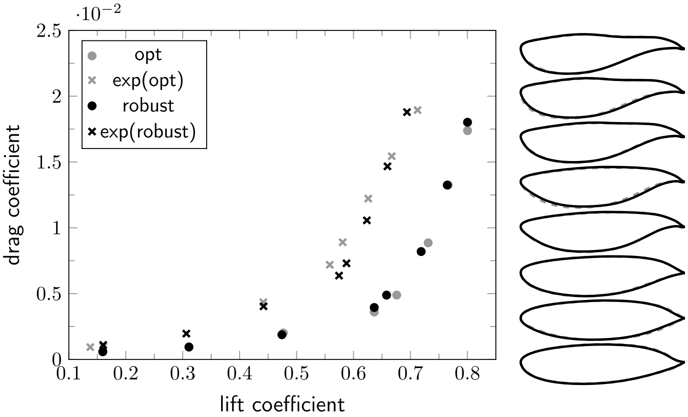
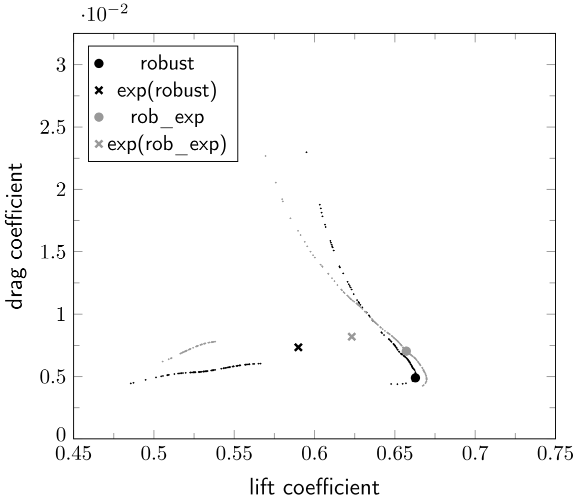


Fig. 12.5 Pareto front for direct approach (expected possible losses)

The constraining of expected possible losses leads to designs with better expected values. In Fig. 12.6 the probability region for a specific design is compared to the probability region of the design obtained using the expectation-based approach. Again, the expectation-based approach leads to higher losses in objective space. The samples of the probability region of the expectation-based approach are dominated by the samples for the design obtained using the direct approach.

When comparing the results of the direct approach with the results of the two-phase approach, it can be seen that the two-phase approach results in designs with

Fig. 12.6 Sampled probability region for direct approach (robust) and the expectation-based approach (rob_exp)



a better probability region. Nevertheless, the direct approach is a good compromise when the additional construction of the deterministic Pareto optimal front is too expensive. It can be expected that the results of the direct approach can be improved using a different approximation of the local Pareto optimal front as presented in Sect. 12.2.4.

12.7 Summary and Outlook

We have presented a new measure for robustness when considering multiple objectives. Two approaches to include expected losses in a robust design formulation are given. A constraint-based multi-objective optimization approach making use of a hybrid method is suggested for solving the robust design problem. The approach is applied for the robust design of an airfoil. The results show that the proposed method successfully finds robust designs with less losses in objective space compared to expectation-based approaches. The direct approach is computationally more efficient and the two-phase approach leads to a higher reduction of expected losses, such that both approaches are of interest for robust optimal design with multiple objectives.

Future research shall include the application to different test cases. The presented test case also offers the opportunity for introducing other types of uncertainties. In previous studies interesting results where obtained for the introduction of uncertainties in the geometry for example due to icing or manufacturing inaccuracies. Furthermore, the aim is to include objective functions from different disciplines. One possibility in the field of aerodynamics is to take into account the structural behaviour.

Acknowledgements We would like to thank Lionel Mathelin from LIMSI-CNRS for the joint development of the two-phase approach for multi-objective robust design.

References

1. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolu Comput 6(2):182–197. https://doi.org/10.1109/4235.996017
2. Deb K, Gupta H (2005) Searching for robust pareto-optimal solutions in multi-objective optimization. In: Coello Coello CA, Hernández Aguirre A, Zitzler E (eds) Proceedings of 3rd international conference on evolutionary multi-objective optimization, lecture notes in computer science, vol 3410. Springer, Berlin, pp 150–164. https://doi.org/10.1007/978-3-540-31880-4_11
3. Kusch L, Gauger NR (2019) Robust airfoil design in the context of multi-objective optimization. In: Minisci E, Vasile M, Periaux J, Gauger N, Giannakoglou K, Quagliarella D (eds) Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences, computational methods in applied sciences, vol 48. Springer, Berlin, pp 391–403. https://doi.org/10.1007/978-3-319-89988-6_23
4. Teich J (1993) Pareto-front exploration with uncertain objectives. In: Zitzler E, Thiele L, Deb K, Coello Coello CA, Corne DW (eds) First international conference on evolutionary multi-criterion optimization, lecture notes in computer science. Springer, Berlin, pp 314–328. https://doi.org/10.1007/3-540-44719-9_22
5. Li M, Silva R, Guimaraes F, Lowther D (2015) A new robust dominance criterion for multiobjective optimization. IEEE Trans Mag 51(3):1–4. https://doi.org/10.1109/TMAG.2014.2372692
6. Gunawan S, Azarm S (2005) A feasibility robust optimization method using sensitivity region concept. J Mech Des 127(5):858–865. https://doi.org/10.1115/1.1903000
7. Xiu D, Karniadakis GM (2002) The wiener-askey polynomial chaos for stochastic differential equations. SIAM J Sci Comput 24(2):619–644
8. Marglin SA (1967) Public investment criteria. MIT Press, Cambridge, MA
9. Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic Publishers, Boston
10. Kusch L, Gauger NR, Spiller M (2014) Efficient calculation of Pareto-optimal points for shape optimization. In: Full paper compilation: evolutionary and deterministic methods for design, optimization and control with applications to industrial and societal problems—EUROGEN 2013, ISBN 978-84-617-2141-2, Universidad de Las Palmas de Gran Canaria, Spain
11. Özkaya E, Gauger NR (2019) Global aerodynamic design optimization via primal-dual aggregation method. arXiv:1811.00433v1 (submitted to STAB Proceedings)
12. Economon TD, Palacios F, Copeland SR, Lukaczyk TW, Alonso JJ (2016) SU2: an open-source suite for multiphysics simulation and design. AIAA J 54(3):828–846. https://doi.org/10.2514/1.J053813
13. Albring T, Zhou BY, Gauger NR, Sagebaum M (2015) An aerodynamic design framework based on algorithmic differentiation. ERCOFTAC Bull 102:10–16

Chapter 13
Uncertainty Assessment of an Optimized ERCOFTAC Pump

R. De Donno, A. Fracassi, A. Ghidoni, and P. M. Congedo

Abstract Centrifugal pumps, being used nowadays for many applications, must be suited for a wide range of pressure ratios and flow rates. To overcome difficulties arising from the design and performance prediction of this class of turbomachinery, many researchers proposed the coupling of CFD codes and optimization algorithms for a fast and effective design procedure. However, uncertainties are present in most engineering applications such as turbomachines, and their influence on turbomachinery performance should be considered. In this work we apply some advanced optimization techniques to the blade optimization of an ERCOFTAC-like pump, and we assess the robustness of the optimal profiles through an uncertainty propagation study. The main sources of uncertainty are related to the operating conditions, primarily the rotational speed of the pump shaft that affects also the flow rate.

Keywords Shape optimization · Uncertainty quantification · Surrogate model · ERCOFTAC pump

R. De Donno
Industrie Saleri Italo S.p.A, via Ruca 406, 25065 Lumezzane, BS, Italy
e-mail: Remo.DeDonno@saleri.it

A. Fracassi (✉) · A. Ghidoni
Department of Mechanical and Industrial Engineering (DIMI),
University of Brescia, via Branze 38, 25123 Brescia, Italy
e-mail: a.fracassi004@unibs.it

A. Ghidoni
e-mail: antonio.ghidoni@unibs.it

P. M. Congedo
DEFI Team (INRIA Saclay Île-de-France and Ecole Polytechnique),
CMAP, 1 rue d'Estienne d'Orves, 91120 Palaiseau, France
e-mail: pietro.congedo@inria.fr

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_13

13.1 Introduction

Centrifugal pumps are used for many applications with different requirements of pressure ratio and flow rate. Their design and performance prediction are not an easy task, being influenced by many free geometric parameters. Experimental approaches based on the modification of prototypes and/or previous models and numerical approaches based on CAD/CFD tools have been applied to their design and analysis. However, the former approach is expensive and time-consuming, while the latter makes available many data which are not easily related to the pump performance, making difficult the improvement of the pump. In the last decade, to overcome these problems, the coupling of CFD codes and optimization algorithms has started to be applyed for the design of turbomachinery [1–5]. This approach has been also successfully applied to the pump design [6–8].

An optimization tool allows to find an optimal design that maximizes some performances in a deterministic sense. However, the performance of turbomachinery can be highly affected by the presence of uncertainty in every engineering application. The aim of this study is to assess how the uncertainty affects the pump performances of the initial and the optimized design, to verify the robustness of the latter. In fact, the new design could have a better efficiency at nominal condition, but a drop of performances in the uncertainty range. The purpose of this work is to investigate the use of an uncertainty propagation approach to assess a deterministic optimization of a complete centrifugal pump (impeller and diffuser) under uncertain operating conditions. Moreover, a critical assessment of two surrogate based optimization strategies is also presented.

In this work an optimal blade profile for the ERCOFTAC pump is obtained with a deterministic optimization through two surrogate based strategies. Then the epistemic uncertainties related to the experimental tests are considered for an uncertainty assessment employing a Polynomial Chaos Expansion. In particular, the main source of uncertainty is constituted by the rotational speed of the pump shaft and by the losses in the cooling system, both affecting also the flow rate.

In the following, Sect. 13.2 is devoted to the description of the geometric parametrization algorithm, Sect. 13.3 describes the CFD solver, while Sects. 13.4 and 13.5 describe the optimization algorithm and the framework for the uncertainty quantification assessment, respectively. Section 13.6 presents the results.

13.2 Geometric Parameterization

The parametrization algorithm allows to represent the blade geometry of the impeller and vaned diffuser as a combination of the camber-line and thickness distribution. 17 design variables are used to parametrize the complete geometry. The inlet and outlet diameters of both impeller and diffuser are fixed. The algorithm can

Table 13.1 Main dimensions of the ERCOFTAC pump

Impeller	
Inlet blade diameter	$D_1 = 240\,\text{mm}$
Outlet diameter	$D_2 = 420\,\text{mm}$
Number of blades	$z_i = 7$
Diffuser	
Inlet vane diameter	$D_3 = 444\,\text{mm}$
Outlet vane diameter	$D_4 = 664\,\text{mm}$
Number of vanes	$z_d = 12$

reproduce the ERCOFTAC blades, but with smooth profiles. The main dimensions of the ERCOFATC pump are summarized in Table 13.1.

13.2.1 Camber-Line

The camber-lines of the impeller and the diffuser are described through a Bézier curve. To define the most suitable number of control points, i.e. the order of the Bézier curve, 3rd, 4th and 5th order curves are considered. The purpose is to use the minimum number of input variables to represent the camber-line of the ERCOFTAC pump blades and of profile commonly used to manufacture pump blades, i.e. the NACA 6-series, the double circular arc (DCA) and the C4 airfoil. These curves are built leaving two degrees of freedom for each control point inside the curve. The approximation error has been measured by evaluating the distance (root mean square distance normalized with the chord length) between the real and parametrized profiles, and is reported in Table 13.2 for each reference camber-line considered.

Table 13.2 Approximation error [%] for different reference camber-lines and order of the Bezier curves

	3rd	4th	5th
Impeller ERCOFTAC	0.518	0.156	0.065
Vaned diffuser ERCOFTAC	0.210	0.0558	0.0457
NACA 63	0.853	0.266	0.105
NACA 64	0.440	0.0946	0.0640
NACA 65	0.127	0.0131	0.000993
DCA	0.182	0.0474	0.0104
C4	0.0115	0.00108	1.71E-05

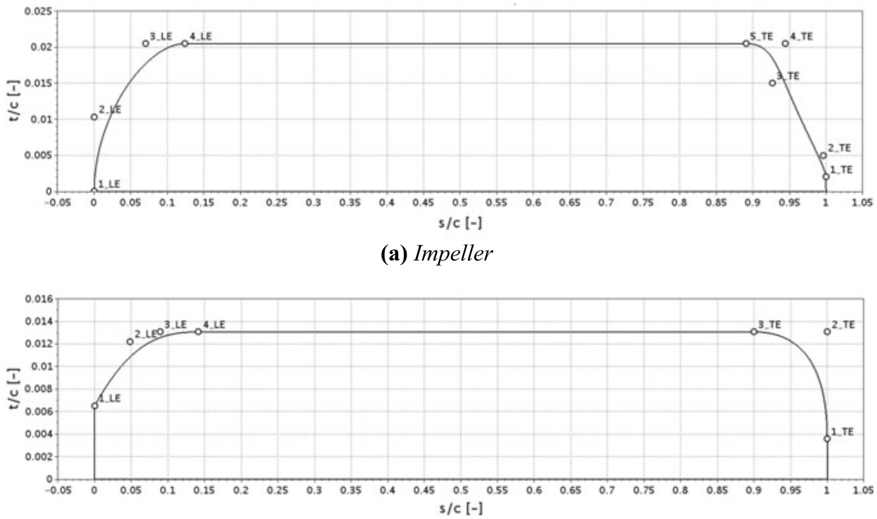
A fourth order Bézier curve has been chosen for the parameterization, characterized by a number of degrees of freedom equal to a third order curve. In fact, during the optimization the distance of the two internal control points from the leading and trailing edge is fixed and equal to the corresponding distance for the ERCOFTAC geometry, while inlet and outlet angles of the blades can change. This choice allows for a better approximation of the ERCOFTAC camber-lines and for a higher geometrical flexibility than a standard third order curve, even if sharing the same number of variables. The approximation error is comparable to a third order curve.

In addition to inlet and outlet angles also the stagger angle is considered a variable.

13.2.2 Thickness Function

The thickness function is parametrized in a different way for the impeller and the diffuser to better fit the thickness function of the ERCOFTAC blades. In both cases the leading and the trailing edge are described through a Bézier curve and they are joined with a constant thickness line (see Fig. 13.1).

The control points are set according to the following formulas, where s denotes the coordinate along the chord and t the coordinate perpendicular to s:



(a) *Impeller*

(b) *Vaned diffuser*

Fig. 13.1 Control points for the thickness function parameterization

- **Impeller leading edge**

$$\begin{aligned}
s_{1\_LE} &= 0 & t_{1\_LE} &= 0 \\
s_{2\_LE} &= 0 & t_{2\_LE} &= 3k_{t,i}\frac{dr^2_{LE,i}}{2} + \hat{y}_{t,i} \\
s_{3\_LE} &= x_{t\_LE,i} - dr_{LE,i} & t_{3\_LE} &= \hat{y}_{t,i} \\
s_{4\_LE} &= x_{t\_LE,i} & t_{4\_LE} &= \hat{y}_{t,i}
\end{aligned}$$

- **Impeller trailing edge**

$$\begin{aligned}
s_{1\_TE} &= c_i & t_{1\_TE} &= \hat{y}_{TE,i} \\
s_{2\_TE} &= c_i + \frac{\hat{y}_{2\_TE,i} - \hat{y}_{TE,i}}{\tan(\alpha_{TE,i})} & t_{2\_TE} &= \hat{y}_{2\_TE,i} \\
s_{3\_TE} &= x_{3\_TE,i} & t_{3\_TE} &= \hat{y}_{3\_TE,i} \\
s_{4\_TE} &= x_{t\_TE,i} + dr_{LE,i} & t_{4\_TE} &= \hat{y}_{t,i} \\
s_{5\_TE} &= x_{t\_TE,i} & t_{5\_TE} &= \hat{y}_{t,i}
\end{aligned}$$

- **Vaned diffuser leading edge**

$$\begin{aligned}
s_{1\_LE} &= 0 & t_{1\_LE} &= y_{1\_LE,d} \\
s_{2\_LE} &= \hat{x}_{2\_LE,d} & t_{2\_LE} &= \frac{\hat{x}_{2\_LE,d}}{\tan(\alpha_{LE,d})} + y_{1\_LE,d} \\
s_{3\_LE} &= x_{t\_LE,d} - dr_{LE,d} & t_{3\_LE} &= \hat{y}_{t,d} \\
s_{4\_LE} &= x_{t\_LE,d} & t_{4\_LE} &= \hat{y}_{t,d}
\end{aligned}$$

- **Vaned diffuser trailing edge**

$$\begin{aligned}
s_{1\_TE} &= c_d & t_{1\_TE} &= \hat{y}_{1\_TE,d} \\
s_{2\_TE} &= c_d & t_{2\_TE} &= \hat{y}_{t,d} \\
s_{3\_TE} &= x_{t\_TE,d} & t_{3\_TE} &= \hat{y}_{t,d}
\end{aligned}$$

The definition of the input variables is reported in Table 13.3, where the letter c is the chord length, the subscripts i and d refer to the impeller and the diffuser, respectively. The hat symbol defines a fixed value. In particular, $\hat{y}_t$ is the maximum thickness of the blade, which is set equal to the thickness of ERCOFTAC blade to compare different profiles and to avoid the computation of too thin blades.

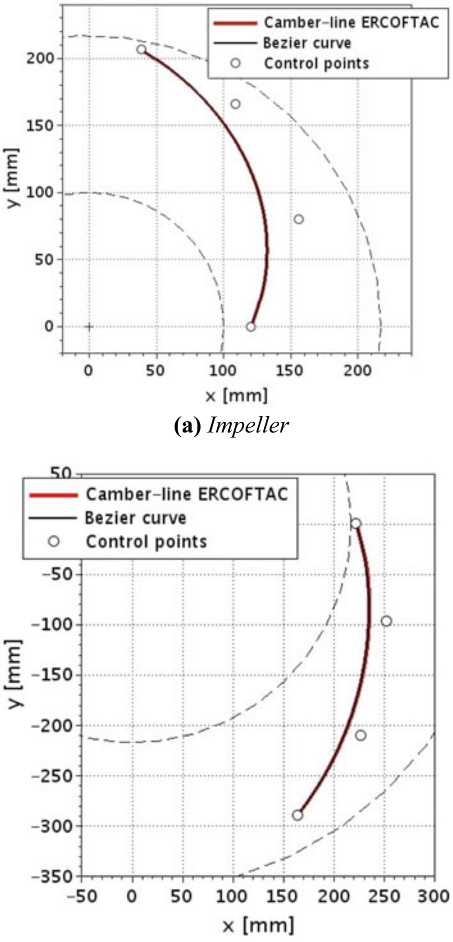
Table 13.3 List of the design variables, description, baseline value, minimum and maximum value during the optimization

Variable	Description	Baseline	Min value	Max value
γ_{imp}	Stagger angle of the impeller (°)	−111.5	−120	−90
β_1	Inlet angle of the impeller (°)	−66.0	−76	−56
β_2	Outlet angle of the impeller (°)	−70.9	−75	−60
γ_{dif}	Stagger angle of the diffuser (°)	101.4	90	105
α_3	Inlet angle of the diffuser (°)	72.8	65	85
α_4	Outlet angle of the diffuser (°)	67.8	60	75
x_{t_LEi}	Position along the chord of the point of maximum thickness at leading edge of the impeller (-)	0.1330	0.1	0.3
$dr_{LE,i}$	Difference between x_{t_LEi} and the radius at leading edge of the impeller (-)	0.06323	0.010	0.064
$k_{t,i}$	Curvature at the point of maximum thickness at leading edge of the impeller (-)	−3.2591	−3.26	−1.00
$x_{t\_TE,i}$	Position along the chord of the point of maximum thickness at trailing edge of the impeller (-)	0.9146	0.80	0.95
$x_{3\_TE,i}$	Position along the chord of the third control point at trailing edge of the impeller (-)	0.9323	0.92	1.00
$\alpha_{TE,i}$	Slope at trailing edge of the impeller	−0.9556	−1.2	−0.5
$x_{t\_LE,d}$	Position along the chord of the point of maximum thickness at leading edge of the diffuser (-)	0.1424	0.1	0.3
$dr_{LE,d}$	Difference between x_{t_LEd} and the radius at leading edge of the diffuser (-)	0.01008	0.005	0.100
$y_{1\_LE,d}$	Thickness at leading edge of the diffuser (-)	4.0129	2.0	5.0
$\alpha_{LE,d}$	Slope at leading edge of the diffuser (-)	0.004266	0.003	0.075
$x_{t\_TE,d}$	Position along the chord of the point of maximum thickness at trailing edge of the diffuser (-)	0.999	0.700	0.999

13.2.3 Range of the Input Variables

The ERCOFTAC geometry is considered as the baseline configuration for the optimization process, and the corresponding values of the input variables are found optimizing the position of the control points to minimize the approximation error (root mean square distance between corresponding points of the real and parametrized geometry).

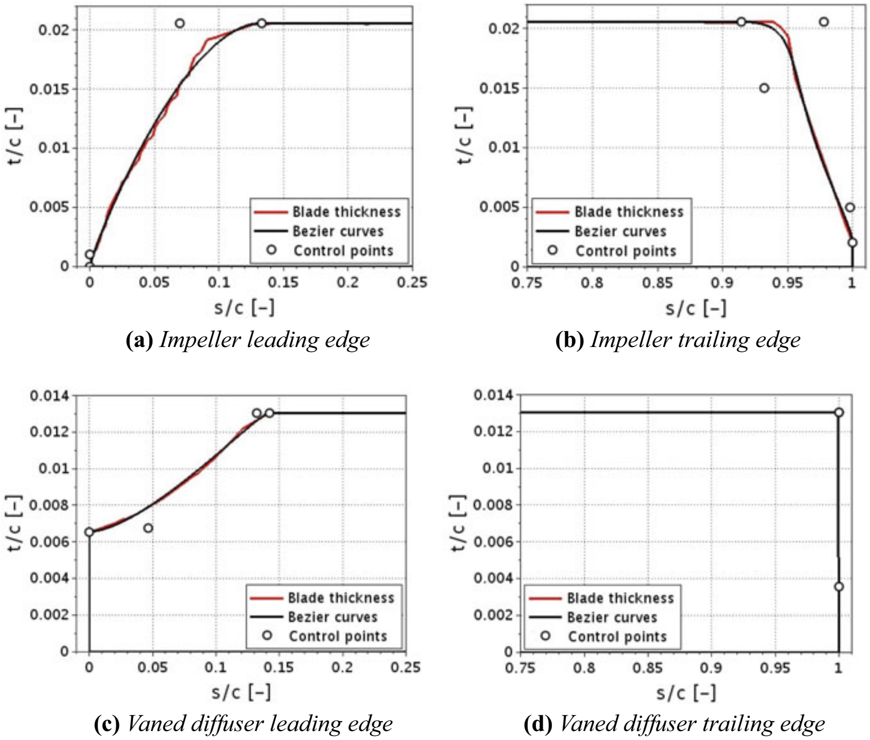
The range of the input variables for the optimization (see Table 13.3) is built ensuring three constraints: (i) input values defining the baseline geometry are included, (ii) not feasible geometries cannot be generated in the Design of Experiment (DoE), and (iii) blade angles for inlet and outlet, which are supposed to be close to the optimum, are included. In particular, the impeller inlet angle is calculated through velocity diagrams, while a common range for impeller outlet angles has been taken from literature. The range of diffuser inlet angles is computed from the impeller outlet angles, applying velocity diagrams, while the range for diffuser outlet angles is computed starting from the volute outlet velocity, estimated with the Stepanoff theory [10], and assuming the flow in the volute satisfies the free-vortex theory.



(a) *Impeller*

(b) *Vaned diffuser*

Fig. 13.2 Control points of the parameterized ERCOFTAC camber-lines



(a) *Impeller leading edge*

(b) *Impeller trailing edge*

(c) *Vaned diffuser leading edge*

(d) *Vaned diffuser trailing edge*

Fig. 13.3 Control points of the parameterized ERCOFTAC thickness function

The comparison between real and parametrized ERCOFTAC geometry is shown in Figs. 13.2 and 13.3.

13.3 CFD

The 2D hybrid meshes of the geometries created during the optimization process are generated with an in-house mesh generator [11]. Only one blade passage is considered for the impeller and diffuser. The size of the elements adjacent to the solid walls is equal to a non-dimensional distance $y^+ \approx 1$, to compute the boundary layer accurately up to the wall.

The open-source CFD toolbox OpenFOAM [12] is used to compute the flow field in the pump. The incompressible Reynolds Averaged Navier-Stokes (RANS) equations coupled with k-ω SST turbulence model [13] are solved. The choice of the turbulence model is dictated by the SST capability to predict correctly flow-fields characterized by adverse pressure gradient and/or detachment, i.e. the expected flow-field of a pump.

Table 13.4 Operating conditions of the ERCOFTAC pump

Operating conditions	
Rotational speed	$n = 2000$ rpm
Flow rate coefficient	$\phi = \frac{4Q}{\pi D_2^2 U_2} = 0.048$
Reynolds number	$Re = 6.5\ 10^5$
Inlet air reference conditions	
Temperature	$T = 298$ K
Air density	$\rho = 1.2$ kg/m^3

Table 13.5 Comparison of different approaches to simulate impeller/diffuser interface for the prediction of η and ψ of the 3D ERCOFTAC pump

CFD approach	ψ (-)	η (%)
unsteady	0.748	87.3
steady-state + Frozen rotor	0.730	84.4
steady-state + Mixing plane	0.764	87.0

On the basis of previous studies [14, 15], which demonstrate the capability of 2D simulations to predict fairly well the ERCOFTAC pump flow-field, 2D simulations have been chosen also for this work to reduce the computational effort.

The operating conditions are summarized in Table 13.4. At the domain inlet the velocity V_1 (computed from ϕ), the turbulence intensity $Tu_1 = 5\%$ and specific dissipation rate ω_1 are prescribed, while at the outflow the mean static pressure is set. Adiabatic wall boundary conditions are applied to all blades.

A steady-state formulation with the Multiple Reference Frame (MRF) approach is used; the impeller and diffuser are fixed with respect to each other, but the momentum equation for the impeller domain is computed in the rotating reference frame. A mixing-plane interface is applied between the impeller and the diffuser. This approach avoid the convection of non-physical wakes created by the impeller blades through the pump, typical problem of the frozen rotor interface. The use of the mixing plane interface allows the simulation of a single blade passage for both impeller and vaned diffuser, reducing significantly the computational cost. Mixing plane and frozen rotor interfaces have been compared in terms of predicted total pressure coefficient ψ and efficiency η with an unsteady simulation for the 3D ERCOFTAC pump. Table 13.5 summarizes the results, showing a good agreement between the mixing plane and unsteady simulations, both in terms of η and ψ.

The second-order upwind discretization scheme is applied to the divergence of the velocity, while the first-order upwind scheme is applied to the turbulent quantities. The Laplacian terms are evaluated using a linear second-order bounded central scheme, while a central differencing method approximates the gradient term.

13.3.1 Performance of the Real and Parametrized ERCOFTAC Pump

The effect of the geometric parametrization in the prediction of the ERCOFTAC pump performance is first investigated, comparing the predicted η and ψ with the real and parametrized geometry. A mesh convergence study has been proposed for both geometries, using four grids with the number of elements ranging from 25000 to 55000. Finer meshes have been obtained refining uniformly the coarser mesh. The grid convergence study (see Fig. 13.4) shows some differences in the predicted results, which can be ascribed to the smooth representation of some geometric details given by the parametrization algorithm. As suggested by the convergence study, the grid with 37000 elements ensures a good compromise between computing time and accuracy of the results, and, therefore, it is chosen for the optimization. For this grid, the difference in the predicted η and ψ for the real and parametrized geometry is summarized in Table 13.6.

An in-depth comparison between the two geometries shows as the main differences are gathered near the leading and trailing edge, where the parameterization smooths the edges of the original geometry and improve the performances of the pump.

To reproduce the exact original geometry, the presence of straight edges in the profile should be enforced. This would lead to a reduction in the performance of the final optimized geometry. However, being the objective of this work the maximization of the pump performance, the parameterization has been not changed and the parametrized geometry is chosen as baseline to be optimized, instead of the original one.

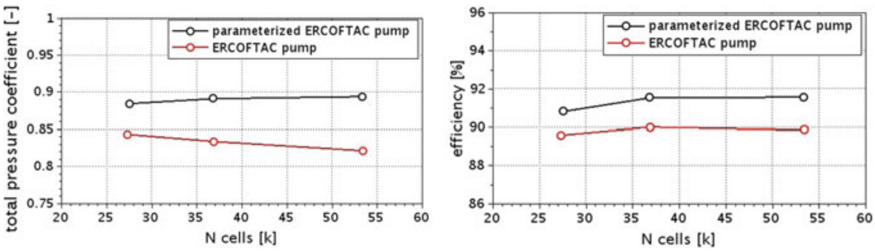


Fig. 13.4 Grid convergence study for the real and parametrized ERCOFTAC pump geometry

Table 13.6 Predicted η and ψ for the real and parametrized ERCOFTAC pump geometry, mesh with about 37000 elements

Geometry	ψ (-)	η (%)
Real	0.833	90.0
Parametrized	0.891	91.5

13.4 Optimization Strategy

In a centrifugal pump, the hydraulic efficiency η and the total pressure rise coefficient ψ have a fundamental role, and, therefore, are chosen as optimization objective and constraint, respectively. In particular, the optimization algorithm maximizes η, while keeping ψ constrained to the considered operating conditions. The efficiency η is defined as the ratio between useful hydraulic power and the provided power, while the total pressure rise coefficient is defined as $\psi = \frac{2(p_{out}-p_{in})}{\rho U_2}$, where the subscript out refers to the pump outlet, in to the pump inlet, and 2 to the impeller outlet. Usually, the pressure head is constrained with a tolerance about $\pm 5\%$, to keep fixed the working condition for the baseline and optimized geometry. However, a numerical investigation has shown that the maximum efficiency is always reached for the upper limit of the constraint, meaning that under uncertainty the constraint could not be guaranteed. Therefore, in the present work only a -5% constraint has been considered for ψ.

Global optimization algorithms require a high number of evaluations in the search of optimum, specially with a high number of input variables. To reduce the computational cost a Surrogate Based Optimization (SBO) [16] is employed.

An initial design of experiments (DoE) with 340 designs distributed over the whole domain is generated through the Latin Hypercube Sampling (LHS) method. Each design is analysed exploiting a CFD simulation, and it is excluded if the CFD calculation does not converge or the solution may not be reliable. A surrogate model is applied to the DoE. The Kriging (KRG) model [17] is suitable for high non-linear objective functions [18] and it is often adopted for turbomachinery optimizations.

Two different SBO strategies are considered:

- A Single Objective Genetic Algorithm (SOGA) is used to solve the optimization problem on the function η and ψ approximated through the surrogate.
- A Global Efficient Optimization (EGO) algorithm [19] is used to search the optimum maximizing the Expected Improvement Function (EIF).

The algorithms available in the software Dakota [20] are used. The EGO algorithm allows to combine exploitation and exploration, so that both zones with good solutions and zones with lack of information are tested. This approach could be advantageous to find the global optimum, if compared to the SBO with SOGA that focuses the research in the zone of good prediction, where the huge number of design variables could lead to accuracy problems in the definition of the response surface.

The optimum found with one of the previous strategies, is verified with a CFD simulation and the design is added to the DoE. Then, the surrogates are updated, iterating until convergence.

13.5 Uncertainty Quantification Assessment

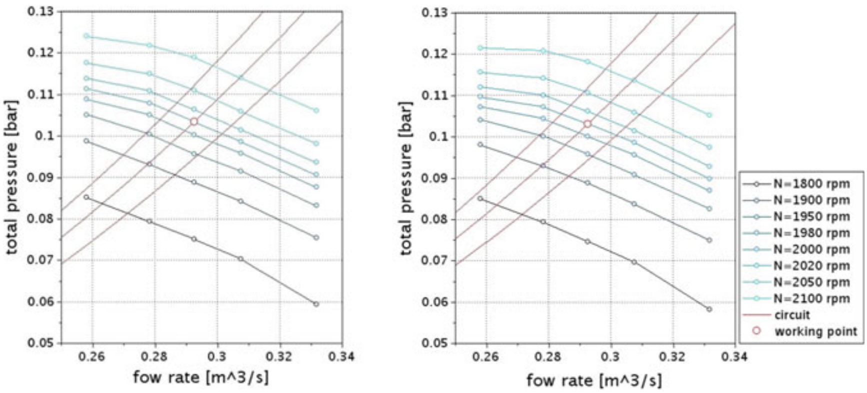
Even if an optimal design is reached in a deterministic environment, the real performance of the optimized pump could be different, due to the presence of uncertainties, which can be found in most engineering applications.

The purpose of this work is to assess the robustness of the deterministic optimum for a centrifugal pump under uncertain boundary conditions. The main sources of uncertainty are the rotational speed and the hydraulic system resistance. In fact, after a preliminary design, the pump is manufactured and tested in a test rig to verify the total pressure head and the efficiency. The pump rotates at the operating rotational speed, and is plugged in a hydraulic system, whose resistance is chosen to provide the operating mass flow rate. The uncertainties of the rotational speed and system resistance have been determined analysing experimental data:

- ±5 rpm for the rotational speed;
- ±8% for the system resistance.

An uniform Probability Density Function (PDF) is defined for each uncertainty and the optimum found by the SBO with SOGA is considered as the deterministic optimum.

To comply with the formulation within the CFD solver, the boundary conditions must be expressed in terms of rotational speed and flow rate. Given the pump curve for a fixed rotational speed and the system curve for a fixed hydraulic resistance, the flow rate is derived by intersecting the two curves. The pump curves for the baseline and the optimum are computed using CFD (see Fig. 13.5), while the system curve is estimated as $\Delta p_{Tot} = a \cdot Q^2$, i.e. a parabola passing through the working point, where the parameter a is proportional to the system resistance. This formulation allows to



(a) *Total pressure head curves of the baseline and system curves*

(b) *Total pressure head curves of the optimum and system curves*

Fig. 13.5 Pump curves of the baseline (left) and the optimum (right) and system curves

apply uncertainty to the system resistance varying the parameter a. Using Fig. 13.5, the minimum and the maximum value of the flow rate given by the uncertainties are obtained. To simplify the problem, a uniform PDF for the flow rate between the previous values is considered, even if this is equivalent to consider a wider uncertainty on the resistance. This simplification is accepted because (i) it ensures higher safety, and (ii) the error of the uncertainty is small.

To assess the robustness of the optimal design and of the baseline, a Polynomial Chaos Expansion (PCE) [21], which is a well-known technique for propagating uncertainties at low computational cost, is employed. It is based on a multidimensional orthogonal polynomial approximation in terms of standardized random variables. A one-to-one correspondence exists between the choice of stochastic variable and the polynomials. For instance, if a normal/uniform variable is considered, the corresponding polynomials are Hermite/Legendre polynomials. The random output R is given by a finite-dimensional series expansion:

$$R = \sum_{i=0}^{P} \alpha_i \Psi_i(\xi), \tag{13.1}$$

where Ψ_i are the multidimensional orthogonal polynomials. They are derived from the family of hyper-geometric orthogonal polynomials or Askey scheme [22]. The α_i are deterministic coefficients of the expansion, computed through a multidimensional integration. A tensor product of Gaussian quadrature rule of fifth order is employed to obtain the expansion coefficients, for a total of 25 evaluations.

Statistics as mean and standard deviation can be computed analytically from the expansion. To evaluate the PDF of the output a Monte Carlo sampling can be performed directly on the polynomial approximation, which is a surrogate model of the function of interest with respect to the input parameters.

13.6 Results

13.6.1 Deterministic Optimization

The initial DOE is constituted by 340 designs, but only 293 are considered, being feasible. SBO with a SOGA algorithm and a EGO have been applied to this DoE to optimize the pump efficiency.

During the optimization process, the convergence can be affected by the presence of not reliable design. This issue is addressed differently for the two strategies. In the SOGA optimization a dummy output is returned, characterized by $\eta = 70\%$ and $\psi = 0.7$. The efficiency value must be lower than the optimum; this value must be chosen carefully because too small values can deteriorate the accuracy of the surrogate model. The value of the pressure coefficient is selected just outside the

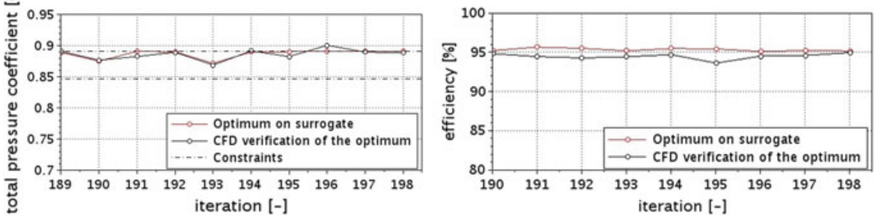


Fig. 13.6 Convergence history of the last set of iterations of the SBO with the SOGA algorithm

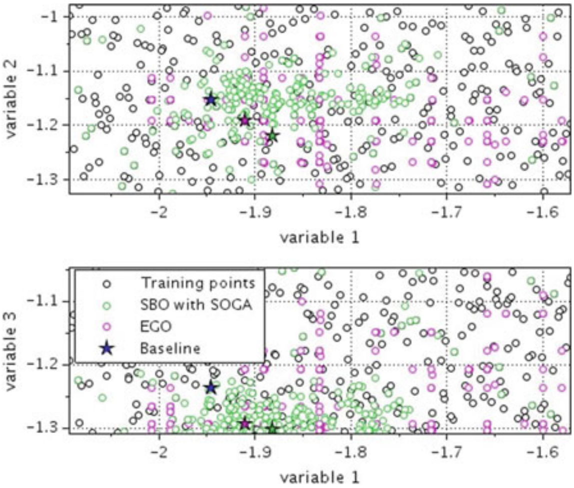
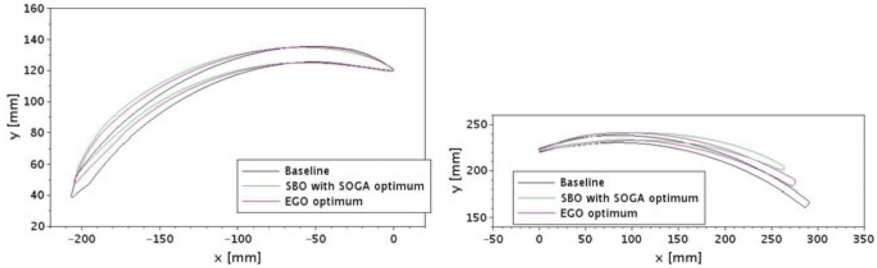


Fig. 13.7 Position of the evaluations during the optimization processes. Variable 1 is the stagger angle, variable 2 is the inlet angle, variable 3 is the outlet angle, of the impeller blades. The stars show the position of the baseline and the optimum for the two strategies

constraint. After a set of iterations, the designs that are not reliable are removed, and the process is reinitialized. For the EGO approach a dummy output can not be used, since it would create instability issues. Therefore, if an unreliable design is found, a reliable one is evaluated in the neighborhood and replaced.

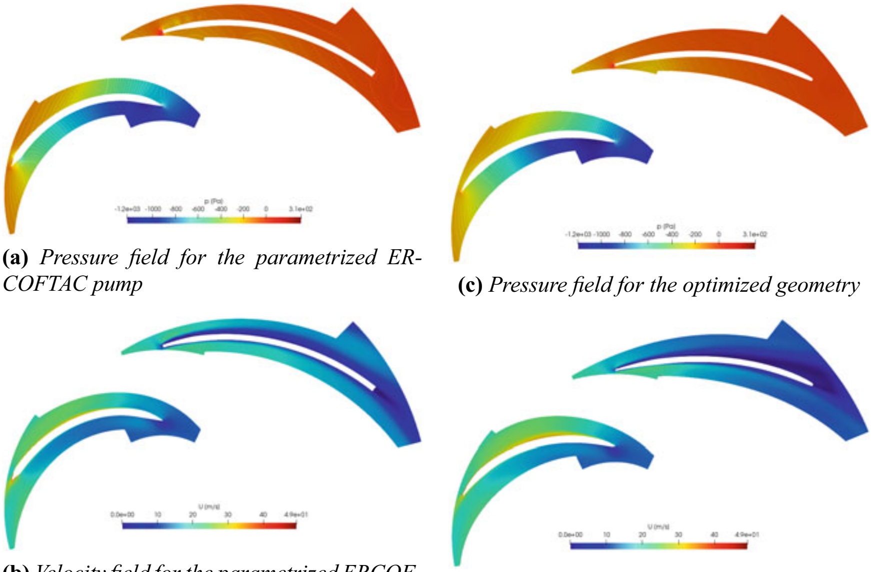
The SOGA optimization reaches the convergence in about 200 iterations and five reinitializations, with a maximum efficiency $\eta_{S,max} = 94.9\%$ and a $\psi_S = 0.889$. In Fig. 13.6 the convergence of the last set of iterations is shown. The EGO converges in about 220 iterations, reaching the maximum efficiency $\eta_{E,max} = 94.6\%$ with $\psi_E = 0.890$. In Fig. 13.7 the evaluated design for the two strategies are represented in the input space. It can be noted as the EGO algorithm scouts a larger space than the SOGA.

In Fig. 13.8 a comparison between the baseline and optimized geometries is shown. In particular the SOGA optimum presents a more tapered impeller blade, a lower chord length of the diffuser and a more rounded trailing edge for the diffuser.



(a) *Impeller* **(b)** *Vaned diffuser*

Fig. 13.8 Comparison between the baseline and the optimized geometry



(a) *Pressure field for the parametrized ERCOFTAC pump*

(c) *Pressure field for the optimized geometry*

(b) *Velocity field for the parametrized ERCOFTAC pump*

(d) *Velocity field for the optimized geometry*

Fig. 13.9 Comparison of pressure and velocity fields of the baseline and the optimized geometry

These features reduce losses, especially at the trailing edge of the impeller, where the

Table 13.7 Values of the design variables for the two optima

Variable	SBO with SOGA optimum	EGO optimum
γ_{imp}	−107.8	−109.4
β_1	−69.9	−68.2
β_2	−74.7	−74.2
γ_{dif}	94.3	97.5
α_3	75.1	70.6
α_4	64.5	64.2
x_{t_LEi}	0.2589	0.1556
$dr_{LE,i}$	0.02725	0.06100
$k_{t,i}$	−3.0502	−3.1344
$x_{t\_TE,i}$	0.8140	0.8417
$x_{3\_TE,i}$	0.9362	0.9244
$\alpha_{TE,i}$	−0.6415	−0.7722
$x_{t\_LE,d}$	0.2156	0.1778
$dr_{LE,d}$	0.01899	0.03139
$y_{1\_LE,d}$	2.0050	2.0556
$\alpha_{LE,d}$	0.07476	0.007000
$x_{t\_TE,d}$	0.7809	0.9244

speed is lowered, and at the leading edge of the diffuser, where the flow detachment of the baseline geometry is not present (see Fig. 13.9). In Table 13.7 the design variables of the two optima are compared.

13.6.2 Uncertainty Quantification Assessment

A PCE is computed for the baseline and the optimum to assess the robustness of the performances. The mean μ and the standard deviation σ of the total pressure coefficient and the efficiency are calculated analytically and they are shown in Table 13.8. Notice that mean values are basically equal to the performances of the related design. Moreover, the standard deviation of the efficiency is for both the design sufficiently small to confirm the deterministic improvement of the optimum, even under uncertain operating conditions. The standard deviation of the total pressure coefficient is equal to the 1.8% of the mean and can not be ignored when constraining the working point.

The PDF, which are estimated through a Monte Carlo sampling on the polynomial expansion and a Kernel Density Estimation (KDE), are also considered (see Fig. 13.10). The remarks based on the previous statistics are confirmed: (i) the efficiency of the optimal design is always higher than the baseline efficiency, and (ii) the two design have the same pressure head PDF. The change in total pressure coef-

Table 13.8 Mean and standard deviation of the performances PDF of the baseline and the optimum

Geometry	μ_ψ (-)	σ_ψ (-)	μ_η (%)	σ_η (%)
Baseline	0.889	0.016	91.4	0.2
Optimum	0.888	0.015	94.8	0.2

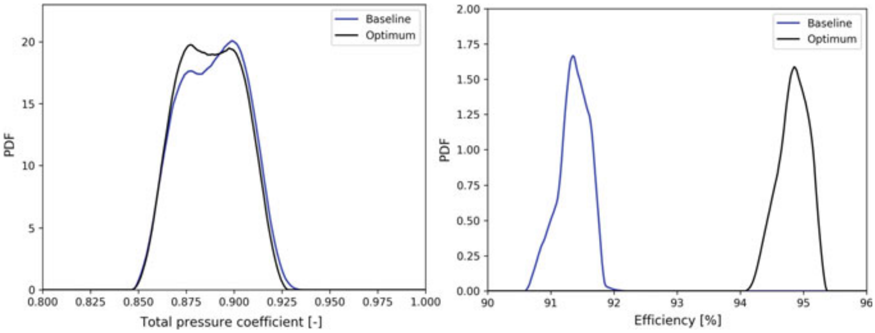


Fig. 13.10 Comparison of the PDF of the total pressure coefficient (left) and the efficiency (right), related to the baseline and the optimum

ficient, even if not negligible, does not break the constraint. In addition, the shape of the PDF for the optimum and the baseline are similar, suggesting that the robustness of the design performances are not effected by the optimization.

13.7 Conclusions

A framework to assess the robustness of the optimum under uncertainty is presented.

First a deterministic optimization is carried out through two different surrogate-based optimization strategies: SBO with SOGA and EGO. The SBO with SOGA reaches an optimal design with an efficiency $\eta = 94.9\%$, corresponding to an improvement about 3.4% with respects to the baseline. The EGO scouts a wider area of the input space, but it does not improve the SOGA result, returning an optimum with an efficiency $\eta = 94.6\%$. The two efficiencies are comparable even though the two design show some differences. This confirms the reability of the SOGA optimum, that is considered in the further study.

Starting from this result, a robustness analysis of the design under uncertainty is performed. In fact, uncertainty occurs in every field. In this work, the focus is on the epistemic uncertainty related to the experimental tests, used to verify the performances of the pump. In particular, the main sources of uncertainty of the test rig are the rotational speed of the pump and the hydraulic resistance of the system.

A polynomial chaos expansion is employed to assess the influence of the operation conditions uncertainties on the efficiency and the total pressure coefficient. The two objective functions PDF of the baseline and the optimum are compared. From this analysis the optimum is robust in terms of efficiency and comparable to the baseline in terms of total pressure head. This validate the result of the deterministic optimization even if uncertainties on the operation conditions are present.

Acknowledgments Research carried out with the support of resources of Big & Open Data Innovation Laboratory (BODaI-Lab), University of Brescia, granted by Fondazione Cariplo and Regione Lombardia

References

1. Van Den Braembussche RA (2006) Optimization of radial impeller geometry. In: Design and analysis of high speed pumps, number RTO-EN-AVT-143. RTO of NATO
2. Pasquale D, Ghidoni A, Rebay S (2013) Shape optimization of an organic rankine cycle radial turbine nozzle. J Eng Gas Turbines Power 135(4):042308-042308-13
3. Guo Z, Song L, Zhou Z, Li J, Feng Z (2015) Multi-objective aerodynamic optimization design and data mining of a high pressure ratio centrifugal impeller. J Eng Gas Turbines Power 137(9):092602-092602-14 09
4. Verstraete T, Alsalihi Z, Van den Braembussche RA (2010) Multidisciplinary optimization of a radial compressor for microgas turbine applications. J Turbomach 132(3):031004-031004-7 03
5. Olivero M, Pasquale D, Ghidoni A, Rebay S (2014) Three-dimensional turbulent optimization of vaned diffusers for centrifugal compressors based on metamodel-assisted genetic algorithms. Optim Eng 15(4):973–992
6. Pei J, Wang W, Yuan S (2016) Multi-point optimization on meridional shape of a centrifugal pump impeller for performance improvement. J Mech Sci Technol 30(11), 4949–4960
7. Wang W, Pei J, Yuan S, Zhang J, Yuan J, Xu C (2016) Application of different surrogate models on the optimization of centrifugal pump. J Mech Sci Technol 30(567–574):02
8. De Donno R, Ghidoni A, Noventa G, Rebay S (2019) Shape optimization of the ercoftac centrifugal pump impeller using open-source software. Optim Eng
9. Beyer H-G, Sendhoff B (2007) Robust optimization âĂŞ comprehensive survey. Comput Methods Appl Mech Eng 196(33):3190–3218
10. Stepanoff A (1993) Centrifugal and axial flow pumps. Krieger Publishing Company
11. Ghidoni A, Pelizzari E, Rebay S, Selmin V (2006) 3d anisotropic unstructured grid generation. Int J Numer Methods Fluids 51(9–10):1097–1115
12. Foam-extend, 4.1 edition. https://sourceforge.net/projects/foam-extend/
13. Menter F (1993) Zonal two equation k-w turbulence models for aerodynamic flows. In: Fluid dynamics and co-located conferences. American Institute of Aeronautics and Astronautics AIAA
14. Petit O, Page M, Beaudoin M, Nilsson H (2009) The ercoftac centrifugal pump openfoam case-study. 01
15. De Donno R, Rebay S, Ghidoni A (2019) Surrogate-based shape optimization of the ercoftac centrifugal pump impeller. Comput Methods Appl Sci 49:227–246
16. Eldred M, Dunlavy D, Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models. In: 11th AIAA/ISSMO multidisciplinary analysis and optimization conference
17. Giunta A, Swiler L, Brown S, Eldred M, Richards M, Cyr E (2006) The surfpack software library for surrogate modeling of sparse irregularly spaced multidimensional data. In: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. AIAA Paper 2006-7049, Portsmouth, VA

18. Li Z, Zheng X (2017) Review of design optimization methods for turbomachinery aerodynamics. Progress Aerosp Sci 93:1–23
19. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Global Optim 13(4):455-492
20. Dakota, 6.8 edition. https://dakota.sandia.gov/
21. Wiener N (1938) The homogeneous chaos. Am J Math 60(4):897–936
22. Askey R, Wilson J (1985) Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials. In: Memoirs of the American Mathematical Society. American Mathematical Society, Providence, RI

Chapter 14
Gradient-Based Aerodynamic Robust Optimization Using the Adjoint Method and Gaussian Processes

Christian Sabater and Stefan Görtz

Abstract The use of robust design in aerodynamic shape optimization is increasing in popularity in order to come up with configurations less sensitive to operational conditions. However, the addition of uncertainties increases the computational cost as both design and stochastic spaces must be explored. The objective of this work is the development of an efficient framework for gradient-based robust design by using an adjoint formulation and a non-intrusive surrogate-based uncertainty quantification method. At each optimization iteration, the statistic of both the quantity of interest and its gradients are efficiently obtained through Gaussian Processes models. The framework is applied to the aerodynamic shape optimization of a 2D airfoil. With the presented approach it is possible to reduce both the mean and standard deviation of the drag compared to the deterministic optimum configuration. The robust solution is obtained at a reduced run time that is independent of the number of design parameters.

Keywords Robust design · Optimization under uncertainty · Adjoint method · Gaussian Processes · Computational fluid dynamics

14.1 Introduction

The use of Robust Optimization in aerodynamic shape optimization is increasing in popularity in order to come up with designs less sensitive against operational and geometrical uncertainties [1–4]. In opposition to deterministic optimization, where the Quantity of Interest, QoI, is a single value to be optimized, in robust optimization the QoI is a random variable. An statistic of this random variable such as the mean,

C. Sabater (✉) · S. Görtz
Institute of Aerodynamics and Flow Technology, German Aerospace Center, DLR,
Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: christian.sabatercampomanes@dlr.de

S. Görtz
e-mail: stefan.goertz@dlr.de

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_14

combination of mean and standard deviation or quantile is usually the objective function.

When dealing with robust optimization involving expensive black box simulations, two problems are commonly present. On the one hand, the complexity of the optimization increases exponentially with the number of design parameters [5]. On the other hand, at each iteration of the optimization, a complete propagation of the uncertainty is required in order to come up with an accurate estimation of the statistic to be minimized [3]. A possible solution to the first problem is the use of adjoint methods [6]. Then, the gradients of the cost function with respect to all the design parameters can be efficiently obtained at a computational cost equivalent to the primal solution. To deal with the problem of uncertainty quantification, the use of surrogate methods such as Gaussian Processes can prove to be efficient to represent the stochastic space [7].

The objective of this paper is the development of a gradient-based robust design framework using the adjoint method and surrogate models and its application to the aerodynamic shape optimization of 2D airfoils.

14.2 Problem Definition

The problem at hand is the minimization of the drag coefficient C_D (the QoI) of the RAE 2822 airfoil against operational uncertainties.

14.2.1 Deterministic Optimization

For reference, a traditional deterministic optimization is computed. In this case, the aim is to find the optimum parameters $\bar{X}$ leading to the airfoil shape that minimizes the drag coefficient at given operational conditions A.

$$\bar{X}^* = \arg\min \left\{ C_D(\bar{X}, A) \right\} \tag{14.1}$$

In this case, the optimization is done at constant lift coefficient, $C_L = 0.79$ and constant Mach number, $M = 0.734$. The lift coefficient constraint is enforced implicitly by iteratively varying the angle of attack during the drag evaluation in the RANS solver.

14.2.2 Robust Optimization

When uncertainties are present, the drag coefficient becomes a random variable. In this case, we choose to minimize a linear combination of mean, μ_{C_D} and standard deviation σ_{CD} of the drag coefficient.

$$\bar{X}^* = \arg\min \left\{ w_\mu \, \mu_{C_D}(\hat{X}, \hat{\xi}) + w_\sigma \, \sigma_{C_D}(\hat{X}, \hat{\xi}) \right\} \tag{14.2}$$

The value of the weights, w_μ and w_σ, are changed in order to come up with different configurations with more focus either on the mean, on its variability or on both. From a different combination of weights, a Pareto front can be obtained with the possible solutions of interest.

14.2.3 Parametrization

The airfoil parametrization follows Hicks-Henne deformation functions [8] that modify the camber of the airfoil. By modifying the airfoil camber, the thickness distribution is kept constant to deal with structural considerations. The vertical displacement, z_i of the camber affected by the design variable X^i can be defined as:

$$z_i = X^i \, \sin\left(\pi x^m\right)^3 \quad \text{where} \quad m = \frac{\log(0.45)}{\frac{i+1}{N_X+4}} \tag{14.3}$$

A total of $N_X = 15$ design parameters are selected. The influence of each bump function in the camber is shown in Fig. 14.1.

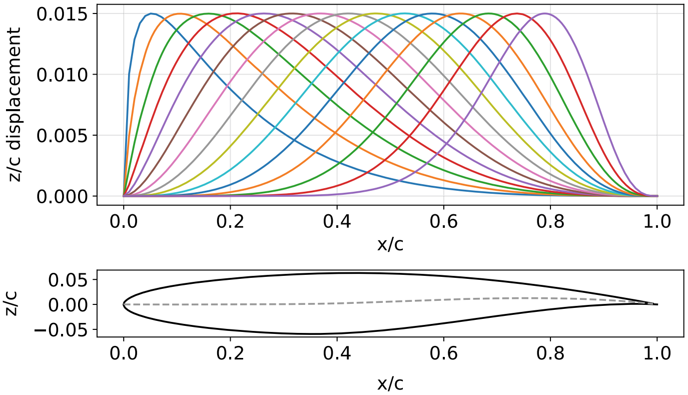


Fig. 14.1 Top: Fifteen Hicks-Henne Bump function used for the parametrization. Bottom: RAE2822 shape and camber line

14.2.4 Uncertainties

In the robust formulation, the Mach and lift coefficient are uncertain as they are expected to slightly change during day to day aircraft operations. They are modeled as symmetric beta distributions. The mean value is centered on the nominal conditions, $\mu_M = 0.734$, $\mu_{C_L} = 0.789$, while the standard deviation is set to $\sigma_M = 0.0045$, $\sigma_{C_L} = 0.0045$. The shape parameters are the same, $\alpha_1 = \alpha_2 = 5$, in order to be symmetric, resembling truncated normal distributions. The truncation allows for a better construction of the surrogate for uncertainty quantification, and for a better representation of the physical problem. The location β_1 and scale β_2 parameters are set to have the required mean μ and standard deviation σ.

$$\text{Beta}(x) = \frac{\gamma\left(\alpha_1 + \alpha_2\right)\left(\frac{x-\beta_1}{\beta_2}\right)^{(\alpha_1-1)}\left(1 - \frac{x-\beta_1}{\beta_2}\right)^{(\alpha_2-1)}}{\gamma(\alpha_1)\gamma(\alpha_2)} \tag{14.4}$$

14.3 Methodology

14.3.1 Numerical Solver

To obtain the aerodynamic performance of the airfoil the high-fidelity DLR flow solver TAU [9] is executed on an HPC cluster system using DLR's FlowSimulator Data Manager (FSDM) environment. The Reynolds Average Navier Stokes (RANS) equations are solved using the Spalart-Allmaras turbulence model. The solution is converged when the density residual is lower than 1e-8. As shown in Fig. 14.2, the unstructured mesh of the baseline configuration, the RAE2822 airfoil, has 29,000 grid nodes, and is quasi two-dimensional. This test case has been successfully used in the past in similar aerodynamic shape optimization problems [3, 10]. A mesh deformation tool developed by DLR using linear elasticity theory [11] is used to change the geometry at any given design vector.

14.3.2 Adjoint Method

The adjoint formulation [6] allows to efficiently solve the total derivative of the QoI with respect to the design parameters X. This is especially useful for high dimensional problems and few cost functions, in which the gradients can be then used for gradient-based optimization [12].

Given the minimization problem of the QoI (in this case the drag coefficient) dependent on the design parameters X, the flow variables W and the mesh variables Z, under the constraint that the flow residual R is converged,

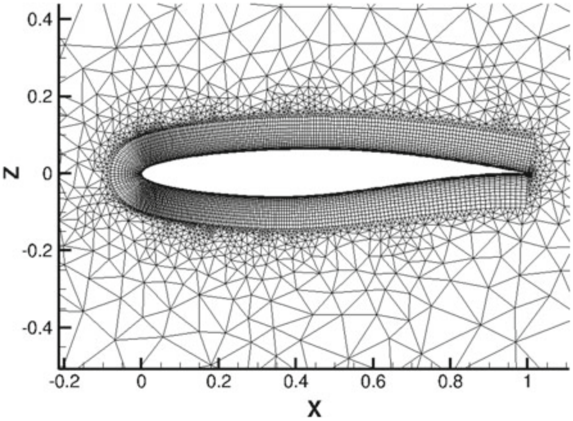


Fig. 14.2 Zoom-in of CFD Mesh of the RAE 2822

$$\min\{\mathrm{QoI}(X, W, Z)\} \quad \text{s.t.} \quad R(X, W, Z) = 0 \tag{14.5}$$

the adjoint equation can be obtained by applying the chain rule to the lagrangian equation:

$$\frac{d\mathrm{QoI}}{dX} = \frac{\partial \mathrm{QoI}}{\partial Z}\frac{\partial Z}{\partial X} + \Lambda^T \frac{\partial R}{\partial Z}\frac{\partial Z}{\partial X} \tag{14.6}$$

where the first term is the variation of the QoI w.r.t. the shape parameter keeping the flow variables, constant. The second term is the variation of the RANS residual w.r.t. the shape parameter by keeping the flow variables constant. The adjoint variables Λ can be obtained from

$$\left(\frac{\partial R}{\partial W}\right)^T \Lambda = -\frac{\partial \mathrm{QoI}}{\partial W} \tag{14.7}$$

In TAU, the discrete adjoint equations are solved [13]. After obtaining Λ, it is possible to evaluate the gradient of the QoI, usually the drag coefficient C_D, w.r.t. the design parameters. When dealing with optimization at constant lift, the gradients w.r.t. the drag must be corrected [14] :

$$\left.\frac{dC_D}{dX}\right|_{C_L=C_{L_0}} = \frac{\partial C_D}{\partial X} - \frac{\partial C_D}{\partial \alpha}\frac{\partial \alpha}{\partial C_L}\frac{\partial C_L}{\partial X} \tag{14.8}$$

The adjoint method has been validated wrt. finite differences for the baseline configuration. Figure 14.3 shows the gradient of the drag coefficient at constant lift with respect to each of the 15 design parameters for both the adjoint and forward finite differences. Despite the small differences, mainly due to the use of forward instead

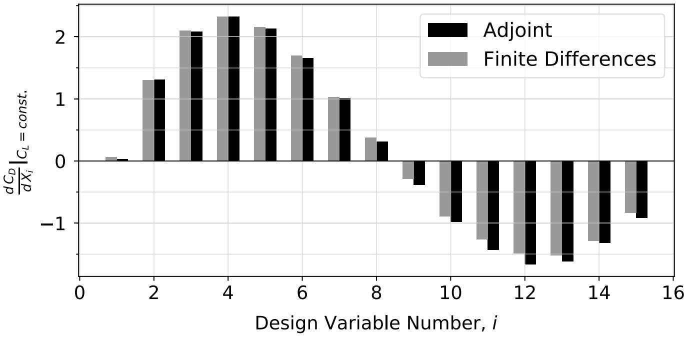


Fig. 14.3 Comparison of the gradients of the drag obtained with finite differences and the adjoint

of central finite differences, the adjoint formulation is able to accurately obtain the desired gradients, reducing the run time by 83%.

14.3.3 Surrogate Based Uncertainty Quantification

The main problem of uncertainty quantification is the large number of function evaluations required to propagate the uncertainty of the input parameters (in this case operational conditions) to the QoI (drag coefficient) at any given design, $\bar{X}_j$ [1]. To directly perform Monte Carlo Simulations is prohibitive when using CFD solvers. A typical approach is then the use of surrogates of the stochastic space for example, through Polynomial Chaos Expansion or Gaussian Processes.

Gaussian Processes models, GPs (also known as Kriging) have been traditionally used in aerodynamic shape optimization as surrogate models for global optimization [15]. However, these have been recently used as non-intrusive approach to perform uncertainty quantification due to its good capability to globally represent the stochastic space [10, 16].

The main idea of uncertainty quantification in Gaussian Processes is as follows: at a given configuration, $\bar{X}_j$, an initial design of experiments (DoE) sampling in the stochastic space $\bar{\xi}$ (in this case random operating conditions), is evaluated in the full order model. Based on this sampling, the GP is built. Then, a large number (N_K) of Quasi Monte Carlo samples can be cheaply evaluated in the surrogate to obtain the statistic, such as the mean or standard deviation of the drag, following Eq. 14.9,

$$\mu_{\text{QoI}}(\bar{X}_j) = \frac{1}{N_K}\sum_{k=1}^{N_K}\hat{\text{QoI}}(\bar{X}_j, \bar{\xi}_k) \qquad \sigma_{\text{QoI}}(\bar{X}_j) = \sqrt{\frac{1}{N_K}\sum_{k=1}^{N_K}\left[\hat{\text{QoI}}(\bar{X}_j, \bar{\xi}_k) - \mu_{\text{QoI}}\right]^2} \tag{14.9}$$

where $\hat{\text{QoI}}\left(\bar{X}_j, \bar{\xi}_k\right)$ is obtained by prediction of the surrogate built in the stochastic space $\bar{\xi}$.

14.3.3.1 Statistics of the Gradients

If the deterministic gradients of the QoI with respect to the design parameters at a given point $\bar{X}_j$ are also available, $\frac{d\text{QoI}}{dX^i}\Big|_{\bar{X}_j,\bar{\xi}_k}$, the gradients of the statistics can also be obtained. In this case a surrogate model needs to be built per each design parameter X^i. For example, the gradient of the mean value of the QoI with respect to a given design parameter X^i at any given design point $\bar{X}_j$, $\frac{d\mu_{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j}$, can be obtained by deriving Eq. 14.9 with respect to X^i:

$$\frac{d\mu_{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j} = \frac{1}{N_K}\sum_{k=1}^{N_K}\frac{d\hat{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j,\bar{\xi}_k} \tag{14.10}$$

In this case the deterministic gradients $\frac{d\hat{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j,\bar{\xi}_k}$ are obtained from direct integration on the given surrogate according to the design parameter X^i.

The gradient of the standard deviation of the QoI respect to each design parameter has also an analytical expression:

$$\frac{d\sigma_{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j} = \frac{1}{N_K\,\sigma_{\text{QoI}}\left(\bar{X}_j\right)}\sum_{k=1}^{N_K}\left(\hat{\text{QoI}}\left(\bar{X}_j,\bar{\xi}_k\right) - \mu_{\text{QoI}}\left(\bar{X}_j\right)\right)\left(\frac{d\hat{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j,\bar{\xi}_k} - \frac{d\mu_{\text{QoI}}}{dX^i}\Big|_{\bar{X}_j,\bar{\xi}_k}\right)^2 \tag{14.11}$$

Then, the stochastic space needs to be characterized for both the QoI (C_D), that is obtained by the primal solution of the CFD solver, and for each of the different N_x gradients of the QoI with respect to the design parameters, that are efficiently obtained by the adjoint method. As shown in Fig. 14.4, $N_X + 1$ different surrogates are constructed, one to obtain the statistics of the primal solution and N_X to obtain the statistics of each of the gradients.

14.3.3.2 Proposed Approach

To construct the surrogate, samples follow a DoE strategy based on Sobol Sequences [17]. Sobol Sequences are a low discrepancy, quasi-random sequence that use a base of two to successively create uniform partitions of the unit interval [17]. The sampling is normalized to the distribution of the input uncertainties, ξ. As a result, more samples will be placed along the mean than in the tails of the input distributions. Locations

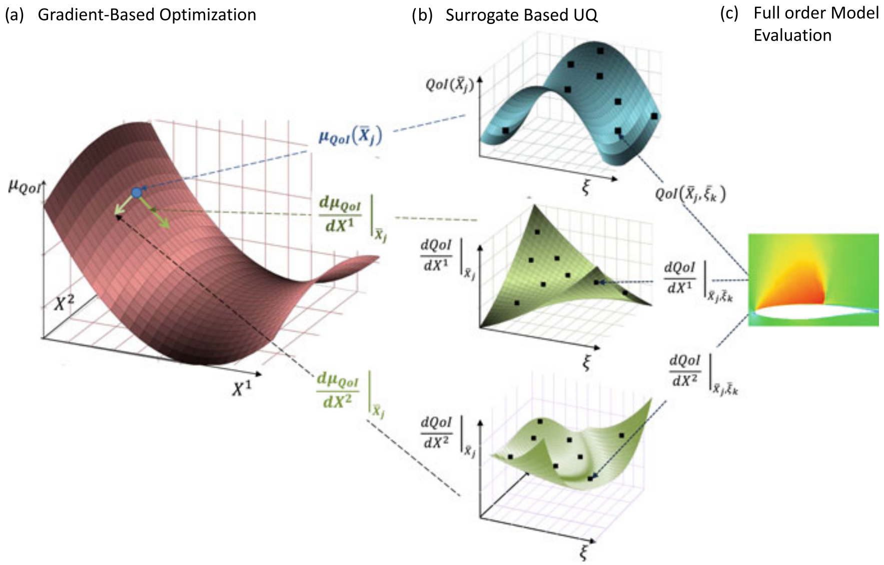

(a) Gradient-Based Optimization
(b) Surrogate Based UQ
(c) Full order Model Evaluation
μ_{QoI}
X^2
X^1
$\mu_{QoI}(\bar{X}_j)$
$\frac{d\mu_{QoI}}{dX^1}\Big|_{\bar{X}_j}$
$\frac{d\mu_{QoI}}{dX^2}\Big|_{\bar{X}_j}$
$QoI(\bar{X}_j)$
ξ
$QoI(\bar{X}_j, \bar{\xi}_k)$
$\frac{dQoI}{dX^1}\Big|_{\bar{X}_j}$
$\frac{dQoI}{dX^1}\Big|_{\bar{X}_j,\bar{\xi}_k}$
ξ
$\frac{dQoI}{dX^2}\Big|_{\bar{X}_j,\bar{\xi}_k}$
$\frac{dQoI}{dX^2}\Big|_{\bar{X}_j}$
ξ

Fig. 14.4 Robust Design Framework using the Adjoint and Gaussian Process: **a** Gradient Based Optimization of the mean of the QoI; **b** Uncertainty Quantification through Gaussian Processes of the QoI and each of its gradients; **c** Evaluation of each deterministic solution in full order model

that will be recalled more often when integrating the surrogate with Monte Carlo will be more accurate than those that have less probability of being evaluated.

The Gaussian Process model consists of Universal Kriging with a Gaussian Kernel (exponent fixed to 2). They hyperparameters of the correlation model are tuned according to the maximization of the model likelihood through Differential Evolution. The Surrogate-Modelling for Aero-Data Toolbox (SMARTy) developed by DLR is used for the initial Design of Experiments sampling and for the creation of the Kriging surrogate [18].

To increase the accuracy of the statistics, after the DoE, an active infill criteria that deals with sampling evenly in the stochastic space [19] is used. Gaussian Processes provide the estimation of the surrogate error at any given point in the stochastic space, $\hat{s}(\bar{\xi})$ [15]. Then, new samples are added in the location $\bar{\xi}_k^*$ where the product of the probability distribution function of the input parameters, PDF_X times the error estimation of the error is maximized. The optimum location is found in the surrogate through Differential Evolution.

$$\bar{\xi}_k^* = \arg\min_{\xi} \left\{-\mathrm{PDF}_X(\bar{\xi})\ \hat{s}(\bar{\xi})\right\} \qquad (14.12)$$

Additional samples are added until convergence on the statistics of the QoI. This is achieved by assessing the error of the statistic that is integrated in the surrogate, $\hat{s}_\mu$ through the Monte Carlo evaluation in both the upper bound, $\widehat{\mathrm{QoI}}(\bar{\xi}) + \hat{s}(\bar{\xi})$, and lower bound, $\hat{\text{QoI}}(\bar{\xi}) - \hat{s}(\bar{\xi})$, prediction given by the surrogate. From here, the upper μ^U_{QoI} and lower μ^L_{QoI} estimation of the statistic are respectively obtained. The difference between upper and lower bound (variability in the determination of the statistics) is associated to the statistical error.

$$\hat{s}_\mu = \frac{\mu^U_{\text{QoI}} - \mu^L_{\text{QoI}}}{2} \tag{14.13}$$

14.3.4 Optimization Framework

As shown in Figure 14.4, the optimization framework combines the gradients obtained by the adjoint formulation with the uncertainty quantification using GPs.

A Sequential Least Squares Programming (SQP) gradient based optimizer is used. At any given design point, $\bar{X}_j$, the optimizer requires both the statistic and its gradients w.r.t. the design parameters $\bar{X}$. Then, at each iteration, the uncertainty quantification is performed in the stochastic space with the help of the surrogate in order to obtain the statistics of the QoI (mean and standard deviation of the drag coefficient in this case). A Gaussian Process is also built for each individual dimension in order to obtain the gradients of the statistics of the QoI w.r.t. the design parameters following Eqs. 14.10 and 14.11. For example, if the focus is in the mean value, both μ_{QoI} and $\frac{d\mu_{\text{QoI}}}{dX}$ are efficiently obtained at each iteration.

This approach differs from the one in which a global surrogate such as Gradient Enhanced Kriging [18] is built, whose values and derivates are computed by the primal and the adjoint. In that case, both the global surrogate accuracy and construction time would be very sensitive to the number of dimensions, N_X. When dealing with more complex problem with hundreds of dimensions, only the training time of the global surrogate would make the approach unfeasible. The strength of the proposed method is that it decouples the dimensionality in the design space from the surrogate accuracy, as this one is built only in the stochastic space with a reduced number of samples. In addition, as each surrogate of the gradients is built independently for each dimension, the training time only increases linearly with the number of design parameters.

14.4 Results

14.4.1 Deterministic Optimization

Figure 14.5 shows the convergence history of the gradient-based deterministic optimization using the adjoint. The optimization starts with the initial RAE2822 configuration. A total of 19 Iterations are required. The optimum configuration decreases

Fig. 14.5 Convergence history of deterministic gradient based optimization

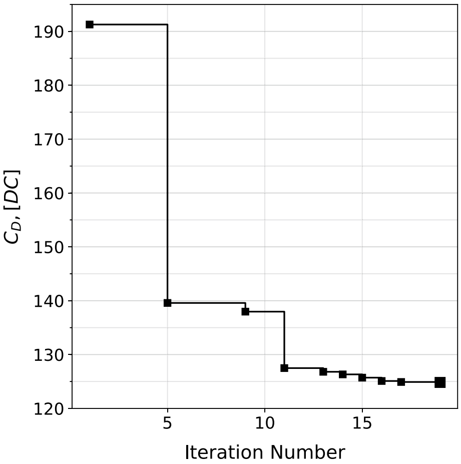
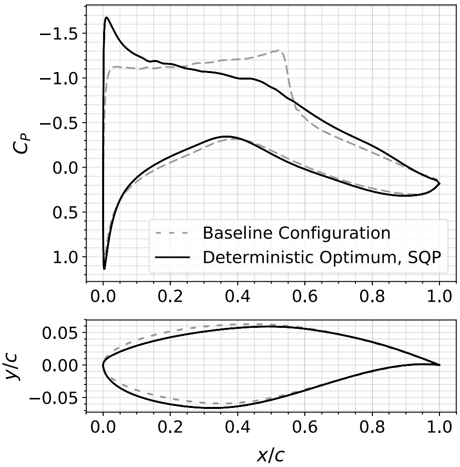


Fig. 14.6 Pressure coefficient for baseline and optimum configurations



the drag coefficient by 34.9%, from to 191.3 drag counts to 124.58 drag counts. According to the pressure coefficient distribution of Fig. 14.6, the optimum airfoil removes the strong normal shock wave of the original configuration, reducing wave drag.

14.4.2 Uncertainty Quantification

To study the accuracy of the proposed uncertainty quantification on GPs, the deterministic optimum configuration is perturbed under uncertainty, following the stochastic operating conditions. To obtain the reference statistics (mean and standard devi-

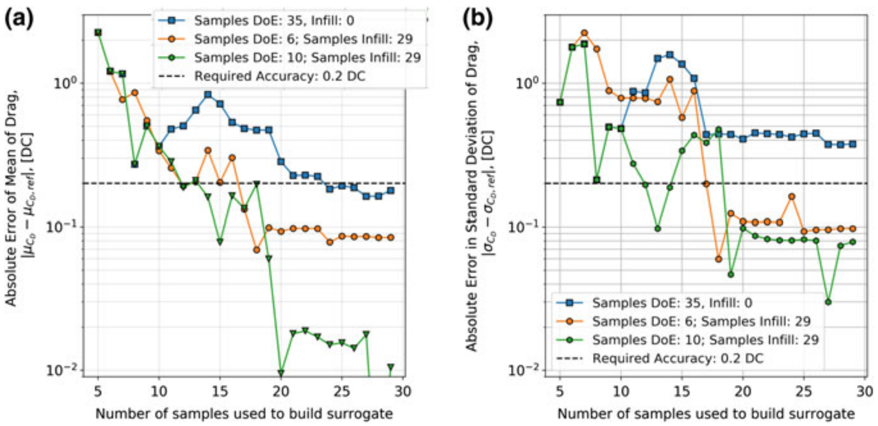


Fig. 14.7 Convergence history of the error in the statistics according to the number of samples used to build the surrogate. Error on: **a** mean; **b** standard deviation

ation) of this configuration, 10,000 Quasi Monte Carlo Samples are evaluated in the CFD model. Based on that, it is possible to obtain the accuracy of the statistics provided by the surrogate for a given number of training samples required to construct them.

Figure 14.7 a shows the convergence in the absolute error between the reference mean and the one obtained with the surrogate built from a given number of samples, for different infill strategies. In general, as the surrogate is built with more and more samples, the mean value is obtained more accurately. However, when only a DoE approach is followed, the accuracy of the surrogate is reduced. For a given computational budget, the use of the infill is preferred. In addition, it is better to start the infill after a good global exploration by using 10 DoE samples. Finally, an error smaller than 0.2 drag counts is desired in order to have a stable convergence during the optimization and provide useful results. According to this, a minimum of 12 to 15 samples are required when the active infill is valid, while if using only a DoE strategy, the required number of samples increases to 24.

The same conclusions can be obtained for the convergence error of the standard deviation in Fig. 14.7 b. When dealing with higher order moments such as the standard deviation the accuracy requirements are more demanding. In this case, the use of the infill criteria is necessary to come up with a good accuracy of the standard deviation.

14.4.3 Robust Optimization

The robust optimization is repeated six times with different weights for the mean and standard deviation following the framework introduced in Sect. 14.3. Each optimum configuration is obtained at a reduced computational cost, requiring from 17 to 24

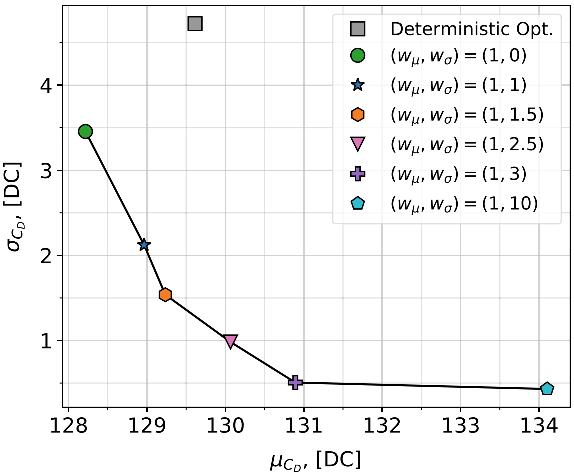


Fig. 14.8 Pareto Front of standard deviation and mean of drag coefficient for optimum configurations

iterations of the gradient-based optimizer. At each iteration, 14 to 16 CFD samples are required to accurately obtain the statistics of the drag through the surrogate approach. Then, a total of 200 to 400 CFD evaluations are required to obtain a optimum robust configuration.

The Pareto-Optimal solutions in terms of mean and standard deviation of drag are shown in Fig. 14.8. The deterministic optimum configuration behaves poorly under uncertainty, and has both higher mean and standard deviation than two of the robust configurations. From an engineering point of view, the configuration with similar weights in mean and standard deviation, ($w_\mu = 1, w_\sigma = 1.5$) looks appealing. By slightly increasing the mean value of the drag, its variability can be reduced by half. There is a clear trade-off between configurations less sensitive to drag, and configurations with a good average performance. Keeping in mind that the gradient based method only guarantees local optimality, the framework is able to provide a set of non-dominated robust solutions in which a designer can choose from. This can only be achieved when the accuracy of the statistics (specially the standard deviation) and its gradients is high.

The probability distributions and box plots of the stochastic drag for the different configurations are shown in the violin plot of Fig. 14.9. On top of each distribution, the mean value is also highlighted in white. The deterministic solution (grey) has a mean value of 129.6 drag counts and a standard deviation of 4.7 drag counts, while the robust solution with focus on the mean value displaces further down the histogram towards a mean value of 128.2 and standard deviation of 3.5 drag counts. However, in both cases a large tail is present towards higher values of drag. When more importance is placed in the standard deviation, solutions have a peaky distribution and the tail is decreased, at an expense of a larger mean value, as previously shown in the Pareto front.

The different airfoil shapes are shown in Fig. 14.10. All the optimum configurations have an increased curvature near their leading edge compared to the baseline

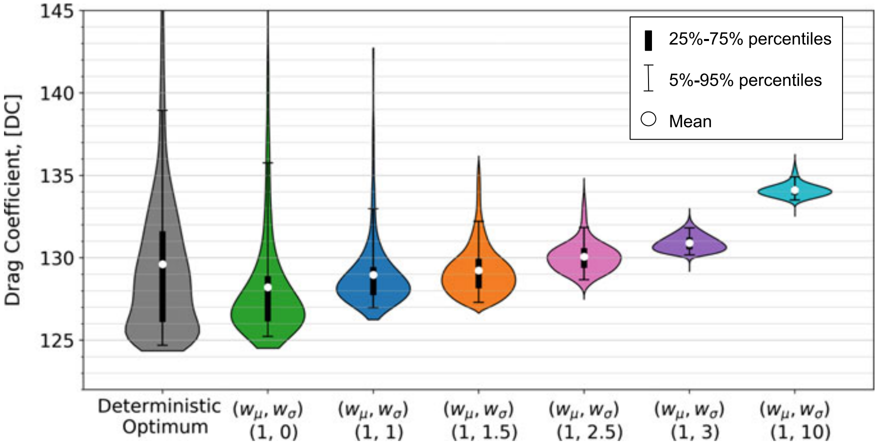


Fig. 14.9 Violin plot of drag coefficient for the configurations of interest

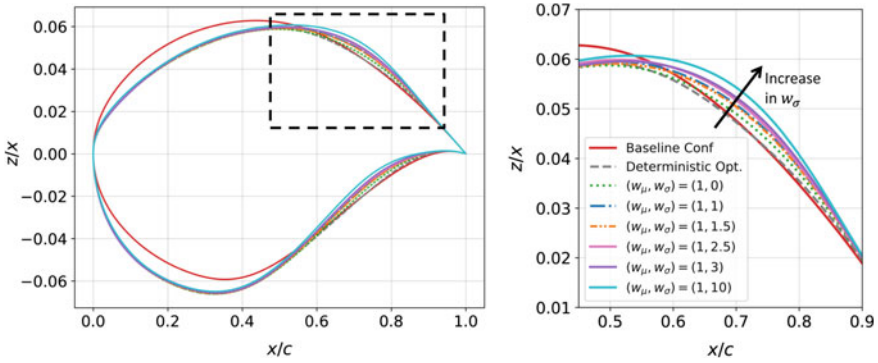


Fig. 14.10 Airfoil shapes of the configurations of interest

airfoil. This allows for a better expansion of the flow and elimination of the strong shock wave over the upper surface. Despite small, there are some differences between the deterministic and robust airfoils.

The robust airfoils have an increased curvature of around 60–70% of the chord. This is similar to adding a "shock control bump" device, that is able to smear stronger shock waves when the Mach and Lift randomly increase w.r.t. nominal conditions. The curvature or "bump" is larger in the designs when variability must be minimized.

The standard deviation of the pressure field is shown in Fig. 14.11 for the robust optimum with focus on the mean ($w_\mu = 1, w_\sigma = 0$, configuration A) and for the one with strong focus on the variation, ($w_\mu = 1, w_\sigma = 10$, configuration B). For each configuration, the field has been obtained by superimposing 300 snapshots of the flow solution computed with Monte Carlo.

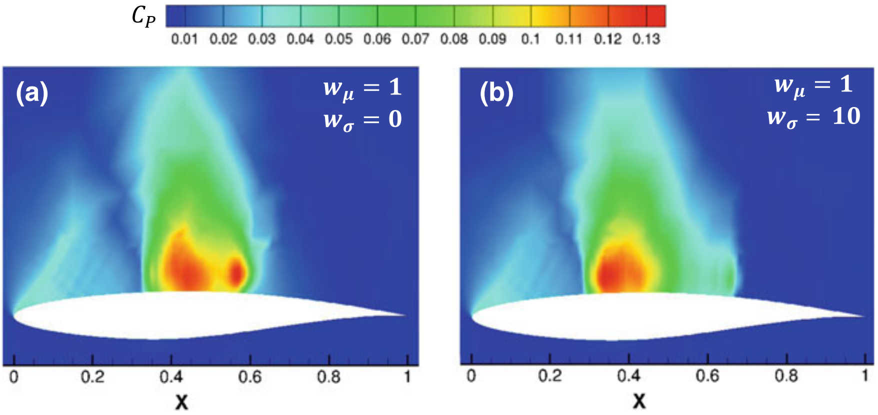


Fig. 14.11 Standard deviation of the pressure coefficient field for Robust configurations: **a** Focus on mean; **b** Focus on standard deviation

There is a larger longitudinal variation of the shock wave along the airfoil in configuration A, as the focus was on the mean drag and not on the standard deviation. In addition, this variation is stronger. In this case, two shock wave patterns can be present around 40–60% of the chord. Configuration B on the other hand reduces the displacement of the shock wave and moves it further upstream, around 35–45% of the chord, due to the stronger curvature previously discussed. However, as the shock wave is further upstream, the average drag increases.

14.5 Conclusions

A novel gradient-based robust optimization method has been presented and applied to a test case. The combination of a CFD adjoint code with Gaussian Process can be used to efficiently obtain the gradients of the mean and standard deviation of the drag coefficient with respect to the design parameters. This reduces both the number of optimization iterations and the samples required for uncertainty quantification.

The application to aerodynamic shape optimization shows that the deterministic optimum under uncertainty behaves poorly. In order to come up with more realistic configurations, a robust formulation is required. A multi-objective optimization in which the mean and standard deviation of the drag compete against each other is an attractive approach for the design of robust configurations. There is a clear trade-off among configurations with good average performance and those with less variability against uncertainties.

This method is preferred in optimizations in which the number of design parameters is much larger compared to the number of uncertainties. With respect to deterministic gradient-based optimization, the addition of uncertainties increases the computational time by a factor of 10 to 15. However, as the framework is independent to the number of design parameters, it is readily available for the robust optimization of more complex three dimensional configurations.

Under more uncertainties, the use of Gradient Enhanced Kriging, which takes the gradients of the uncertain parameters to build the surrogate in the stochastic space, will increase the accuracy of uncertainty quantification. In the future, other robustness measures such as the quantile will be investigated. The framework will also be applied to the optimization of 3D wings under a large number of design parameters, where it will show its full potential.

Acknowledgments This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement number 722734.

References

1. Duvigneau R (2007) Aerodynamic shape optimization with uncertain operating conditions using metamodels. resreport RR-6143, INRIA
2. Schulz V, Schillings C (2013) Optimal aerodynamic design under uncertainty. In: Notes on numerical fluid mechanics and multidisciplinary design. Springer, Heidelberg, pp 297–338
3. Maruyama D, Liu D, Görtz S (2016) An efficient aerodynamic shape optimization framework for robust design of airfoils using surrogate models. In: Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016). NTUA, Greece
4. Kumar D, Raisee M, Lacor C (2018) Combination of polynomial chaos with adjoint formulations for optimization under uncertainties. In: Uncertainty management for robust industrial design in aeronautics. Springer International Publishing, pp 567–582
5. Shan S, Wang GG (2009) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct Multi Optim 41:219–241
6. Giles MB, Pierce NA (2000) An introduction to the adjoint approach to design. Flow, Turbul Combust 65:393–415
7. Maruyama D, Görtz S, Liu D (2018) General introduction to surrogate model-based approaches to UQ. In: Uncertainty management for robust industrial design in aeronautics. Springer International Publishing, pp 203–211
8. Hicks RM, Henne PA (1978) Wing design by numerical optimization. J Aircr 15:407–412
9. Gerhold T (2015) Overview of the hybrid RANS code TAU. In: MEGAFLOW—numerical flow simulation for aircraft design. Springer Berlin Heidelberg, pp 81–92
10. Sabater C, Görtz S (2019) An efficient bi-level surrogate approach for optimizing shock control bumps under uncertainty. In: AIAA Scitech 2019 forum. American Institute of Aeronautics and Astronautics
11. Gerhold T, Neumann J (2006) The parallel mesh deformation of the DLR TAU-code. In: Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM). Springer Berlin Heidelberg, pp 162–169
12. Brezillon J, Dwight RP (2011) Applications of a discrete viscous adjoint method for aerodynamic shape optimisation of 3d configurations. CEAS Aeronaut J 3:25–34
13. Dwight R (2006) Efficiency improvements of rans-based analysis and optimization using implicit and adjoint methods on unstructured grids. In: DLR Deutsches Zentrum fur Luft- und Raumfahrt e.V. - Forschungsberichte

14. Reuther J, Jameson A, Farmer J, Martinelli L, Saunders D (1996) Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. In: 34th aerospace sciences meeting and exhibit. American Institute of Aeronautics and Astronautics
15. Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. Progress Aerosp Sci 45:50–79
16. Maruyama D, Liu D, Görtz S (2018) Surrogate model-based approaches to UQ and their range of applicability. In Uncertainty management for robust industrial design in aeronautics. Springer International Publishing, pp 703–714
17. Sobol I (1967) On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput Math Math Phys 7:86–112
18. Han Z-H, Görtz S, Zimmermann R (2013) Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. Aerosp Sci Technol 25:177–189
19. Dwight R, Han Z-H (2009) Efficient uncertainty quantification using gradient-enhanced kriging. In 50th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference. American Institute of Aeronautics and Astronautics

Chapter 15
A Multi Layer Evidence Network Model for the Design Process of Space Systems Under Epistemic Uncertainty

Gianluca Filippi and Massimiliano Vasile

Abstract The purpose of this paper is to introduce a new method for the design process of complex systems affected by epistemic uncertainty. In particular, a multi-layer network is proposed to model the whole design process and describe the transition between adjacent phases. Each layer represents a design phase with a particular detail definition, each node a subsystem and each link a sharing of information. The network is used to quantify and propagate uncertainty through the different layers (design phases) where, proceeding from phase A to phase F, the detail of the mathematical model is increased. Thus, it can be considered as a multi-fidelity approach for the design of a complex system affected by epistemic uncertainty. The framework of Dempster-Shafer Theory of Evidence (DST) is used to model epistemic uncertainty. The model is then called Multi-Layer Evidence Network Model (ML-ENM).

Keywords Multy-layer evidence network model · Evidence theory · Robust design

15.1 Introduction

The approaches to the design of engineering systems have been evolving in time at an equal pace with the development of technology and in particular with the increase of computational power. Within the *Design by Formula*, the first traditional approach in engineering design, the solution is generated by the engineer without the help of any tool and it is based only on the feedback given by the physical prototype. In the *Design by Analysis* [1] approach, virtual prototype (software analysis based on numerical methods) gives an important contribution, but still the design process is

G. Filippi (✉) · M. Vasile
Aerospace Centre of Excellence, Mechanical and Aerospace Engineering, University of Strathclyde, James Weir Building, 75, Montrose Street, Glasgow, UK
e-mail: g.filippi@strath.ac.uk

M. Vasile
e-mail: massimiliano.vasile@strath.ac.uk

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_15

based on a reductionist approach where subsystems and components are designed separately without a particular attention to their interfaces. A system level design, instead, is handled by the *Design by Optimisation* [2] with the use of numerical optimisation algorithms integrated with analysis tools. A fundamental improvement is given then by the *Design for Reliability and Robustness* and more in general by Multi-Disciplinary Optimisation (MDO) under Uncertainty [3, 4] that better model a real (engineering) system which is inevitably affected by uncertainty and imprecision. Design Under Uncertainty (DUU) makes designers able to handle higher degree of complexities but, on the other hand, it is particularly challenging due to its high computational cost. If one looks at the different types of uncertainty that a system can be subjected to, two macro-categories can be identified: *aleatory uncertainty* and *epistemic uncertainty* [5]. Aleatory uncertainty is natural randomness which cannot be reduced. Epistemic uncertainty is due to the lack of information or incomplete data. This type of uncertainty is reducible by acquiring more knowledge on the problem. Considering this, a further step forward have been proposed by the authors with the *Design by Resilience* [6]. Our proposed concept of Resilience Engineering extends and integrates the concepts of Design for Reliability and Design for Robustness and uses the framework of Dempster-Shafer theory of evidence (DST) [7] to include epistemic uncertainty.

We are here particularly interested in space systems. They are complex systems involving multiple interconnected components and disciplines with complex couplings: payload, structure, thermal analysis, attitude, control, etc. A system level optimal solution cannot be found by optimising the single subsystems independently. Furthermore, the design and optimisation of space systems have to account for epistemic uncertainty, in particular in the early design phase. In fact, knowledge about system and requirements is only acquired incrementally, but substantial commitments are made upfront, essentially in the unknown.

Even if the research field is demonstrating to be very active in proposing new and promising methodologies for the DUU of complex systems, space industry, on the other side, has a conservative approach that is based on traditional methods. In fact, the most common and well-established approach to handle uncertainty in space systems engineering is to use safety margins and redundancies [8]. These traditional methods, however, present two critical problems that affect the result of the design process. There is a lack of an appropriate quantification of uncertainty that brings to an overestimation or an underestimation of the effect of uncertainty (increase in costs and development time or occurrence of undesirable events). There is also a lack of an holistic view on the system performance and evolution.

In this paper, then, we propose a methodological advancement to solve those two problems with specific application to the design of space systems. The novelty is given by a mathematical model, in the form of a multi-layer graph, that simulate the evolution in time of the space system during the design process and is able to quantify and propagate epistemic uncertainty through the different design phases. The model, called Multi-layer Evidence Network Model (ML-ENM) is a generalisation of the Evidence Network Model (ENM) already suggested by the authors. The ENM formulation was first introduced in Ref. [9]. The method was extended in

Ref. [10] to make ENM computationally more efficient. Reference [11] introduced a time-dependent reliability measure in the ENM and finally Ref. [6] introduced the concept of resilience. The ML-ENM allows to a rigorous and fast propagation of epistemic uncertainty [6, 9–11] and gives an holistic view to the whole design process. Each layer represents a different phase in the design process, each node represents a subsystem or a component at a particular level of granularity and each link is a sharing of information.

In particular, this paper proposes a method to propagate uncertainty through the ML-ENM from the last design phase to the first one. Then the system is optimised for robustness with the min-max algorithm [12, 13]. Evidence Theory is applied to quantify uncertainty on the optimal solution [6, 9–11]. It is finally shown that the optimal solution at phase A is robust against the uncertainty in the next design phases.

15.2 Evidence Framework for Epistemic Uncertainty

As previously stated in Sect. 15.1, the design process of a space system is affected (particularly in the early phases) by epistemic uncertainty that can not be quantified by probability distributions. To model this type of uncertainty we propose the use of DST which is getting an increasing attention and has shown to be useful [7].

Under the assumptions of independence of the sources of information and uncorrelation of uncertainties, we can define the set Θ of all the mutually exclusive and collectively exhaustive elementary events (or hypotheses) $\Theta = \left\{\theta_1, \theta_2, \ldots, \theta_i, \ldots, \theta_{|\Theta|}\right\}$. The collection of all non empty subsets of Θ is the Power Set $2^\Theta = (\Theta, \cup)$. One can now assign a probability mass, called basic probability assignment (*bpa*) to the elements of 2^Θ. Each element of 2^Θ with a non-zero *bpa* is called a *Focal Element* (FE) and is represented with the symbol γ in the following. The pair $\langle \Gamma, bpa_\Gamma \rangle$—where $\Gamma \ni \gamma$ and $bpa_\Gamma \ni bpa_\gamma$—is called the *Body of Evidence*.

We can now define the performance index of the system we want to analyse as:

$$f(\mathbf{d}, \mathbf{u}) : D \times U \subseteq \mathbb{R}^{m+n} \to \mathbb{R} \tag{15.1}$$

where D is the design space for the decision or design parameters $\mathbf{d}$, of dimension m, and $U = 2^\Theta$ the event space for the uncertain parameters $\mathbf{u}$, of dimension n, that we call the *Uncertain Space*.

DST measures the influence of uncertainty on the quantity f, for a fixed design vector $\mathbf{d}^*$, by means of two functions, *Belief* and *Plausibility*, that generalise the concept of Probability measure given in classical probability theory. If we are interested in the amount of evidence associated to the event $f(\mathbf{d}, \mathbf{u}) \in \Phi$ we can define

$$\Omega = \{\mathbf{u} \in U | f(\mathbf{d}, \mathbf{u}) \in \Phi\} \tag{15.2}$$

as the corresponding set in U and then compute the cumulative Belief and Plausibility associated to that event:

$$Bel(\Omega) = \sum_{\gamma_i \subset \Omega, \gamma_i \in U} bpa(\gamma_i), \tag{15.3}$$

$$Pl(\Omega) = \sum_{\gamma_i \cap \Omega \neq 0, \gamma_i \in U} bpa(\gamma_i). \tag{15.4}$$

From Eqs. (15.3) and (15.4) we can state that the belief in the realisation of the event $f(x) \in \Phi$ is the sum of the *bpa* of all the FEs totally included in Ω, while the Plausibility is the sum of all the FEs that have a non-null intersection with Ω. More details about the DST can be found in Ref. [7].

15.3 Evidence-Based Robust Optimisation

This section explains the approach we use to incorporate epistemic uncertainty in the optimisation process and to design the system for robustness.

Given the performance index f in (15.1), Evidence-Based Robust Optimisation aims at finding the decision vector $\mathbf{d}^*$ that maximises the Belief in statement (15.2), given a body of evidence, and optimises the set Φ. The concept was introduced by the authors in Ref. [14] and extended in Ref. [15].

If one is interested in the minimisation of f under the satisfaction of a constraint function $C \leq \nu_C$, Eq. (15.2) translates in the following two sets of uncertain parameters:

$$\Omega = \{\mathbf{u} \in U \mid f(\mathbf{d}, \mathbf{u}) \leq \nu\} \tag{15.5}$$

$$\Omega_C = \{\mathbf{u} \in U \mid C(\mathbf{d}, \mathbf{u}) \leq \nu_C\} \tag{15.6}$$

where we want to minimise f and maximise the belief in the statement (15.5) while maintaining an hard condition on the constraint satisfaction:

$$\begin{array}{l} \max_{\mathbf{d} \in D} Bel(f(\mathbf{d}, \mathbf{u}) \leq \nu) \\ \min_{\nu \in \mathbb{R}} \nu \\ Bel(C(\mathbf{d}, \mathbf{u}) \leq \nu_C) > 1 - \epsilon \end{array} \tag{15.7}$$

Problem (15.7) requires the evaluation of the belief curve for both the functions f and C and it becomes easily intractable. In fact there is a dependence of the belief to the design vector $\mathbf{d}$ and the thresholds ν and ν_C thus for each new value of $\mathbf{d}$, ν and ν_C the belief has to be revalued. Furthermore the exact belief reconstruction requires a number of maximisations equal to the number of FEs and this number increases exponentially with the problem dimension.

Among all vectors $\mathbf{d}$ that solve problem (15.7) the most critical one, $\mathbf{d}^*$, corresponds to the minimum values of ν and ν_C such that $Bel(f(\mathbf{d},\mathbf{u}))$ is maximum and $Bel(C(\mathbf{d},\mathbf{u}) \le \nu_C)) = 1$. We call the search for $\mathbf{d}^*$, worst-case scenario optimisation and it can be formulated as the deterministic min-max optimisation problem [3]:

$$\begin{array}{l} \min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u}) \\ s.t. \\ \forall \mathbf{u} \in U : \quad C(\mathbf{d},\mathbf{u}) \le 0. \end{array} \tag{15.8}$$

Solving for the worst-case scenario renders the optimisation problem independent of the uncertainty quantification method, has a complexity that is independent of the number of FEs and does not require any particular assumption on the constraint functions.

15.4 Space Systems Project Life Cycle

Space missions are complex and expensive. The design process, in fact, requires several years (up to 15) to find the final optimal configuration. Also it involves different players who have different goals: end user or costumer, operators, developer and sponsor. In order to decompose the whole design process in smaller and more manageable pieces, the life cycle of a space mission traditionally proceeds through four main phases. The *concept exploration* broadly defines the space mission and its components. The *detailed development* defines more precisely the system's components and possibly tests software and hardware. The *production and deployment* constructs and launches the system. The *operations and support*, finally, daily supports the mission and brings it to its end of life [16]. Depending, then, on the mission's sponsor (NASA, ESA, DoD, commercial enterprise, ...), these phases are further divided and labelled differently. For example, NASA divides the project life cycle in seven incremental pieces [17, 18]. The *Pre-Phase A* (concept study), the *Phase A* (concept and tecnolongy development), the *Phase B* (Preliminary Design and Technology Completion), the *Phase C* (Final Design and Fabrication), the *Phase D* (System Assembly, Integration and Test, Launch), the *Phase E* (Operations and Sustainment) and the *Phase F* (Closeout). Phases A, B, …, F are separated by Key Decision Points (KDPs) that are events in which the authority, based on the progress state, the achieved results, the requirements and the budget, approve or reject the project with a "go" or "not go" decision.

15.4.1 Pre-formulation

This phase is not part of the project life-cycle. It is nevertheless of fundamental importance. Feasibility and desirability are here preferred to optimality. Engineers are interested in a broad analysis of risks, cost, feasibility. A variety of possible scenarios and ideas are analysed.

15.4.2 Formulation

During the formulation, that includes phase A and phase B, the full range of implementation options are explored and finally a promising design concept is proposed. The formulation includes the development of the system architecture. Mission and preliminary design are finalised thanks to trades between conducting safety, technical, cost, and schedule risk. A the end of the *formulation*, the project plan is prepared for the implementation phase.

15.4.3 Implementation

The project implementation consists of phases C, D, E and F. During phase C, there is the completion of the final system design, the fabrication and the test of components, assemblies and subsystems. Phase D, instead, includes the system assembly integration and test, the pre-launch activities, the launch, on-orbit check out, and the initial operations. Phase E controls the operation during the mission life-time. Finally, phase F concludes.

15.5 Multi-layer Evidence Network Model (ML-ENM)

This section introduces the concept of ML-ENM that can be used to quantify and propagate epistemic uncertainty through the complex system and the different phases of the whole design process.

ML-ENM generalises the ENM that has been presented in Refs. [6, 9–11]. ENM is a framework for a decomposition procedure that evaluate Belief and Plausibility curves with a computational cost that is polynomial and not exponential with the problem dimension.

Although a network representation for complex systems is a common approach in MDO, the presented (ML-)ENM gives new and important advantages with respect to the commonly used Design Structure Matrix (DSM) [19]. In particular, within (ML-)ENM the correlations between nodes are represented by scalar values that model in a compact way the influence of many uncertain parameters and weight the different links. Also (ML-)ENM allows for an easier representation of sub-networks and clusters.

ML-ENM is a multi-layer network where each layer represents a different phase in the design process, each node represents a subsystem or a component at a specific level of granularity and each link is a sharing of information. As the design process proceeds from pre-phase A to phase D, an increasing level of detail is needed in the analysis, the focus is shifted from the subsystem level to the component level, more precise mathematical models are implemented and the number of nodes increases. On the other side, studying how real projects evolve, there is a high level of confidence that between a phase and the following one, unforeseen circumstances require a modification of the design requirements and goals. Furthermore, different players collaborate in the project and, usually, a good communication between them is not an easy task. Based on the results of the single design phase and on the uncertainty on the whole process' evolution, the designers take decisions that bring them to the next phase and that will drive the design process.

Looking at Fig. 15.1, for example, during phase A three subsystems are considered and optimised. During phase B the number of considered components is increased. The point is that the number of sub-divisions and the types of the components in phase B depend on the designers' choices and each decision brings the design process to a different final solution. Also, the number of possible final configurations increases exponentially with the number of layers and possible choice that can be selected between each couple of layers.

More formally, a ML-ENM with N_L layers is a pair (G, C) where G = $\{G_\alpha; \alpha \in \{1, .., N_L\}\}$ is a family of directed and weighted graphs $G_\alpha = (X_\alpha, E_\alpha)$ and $C = \{E_{\alpha\beta} \subset X_\alpha \times X_\beta; \alpha, \beta \in \{1, 2, \ldots, N_L\}, \beta = \alpha + 1\}$ is the set of interconnections between nodes of different layers. The *intralayer* links in E_α represents the sharing of information between subsystems and components of the space system (complex system). The *interlayer* links in $E_{\alpha\beta}$ model the decision process tree between different design phases.

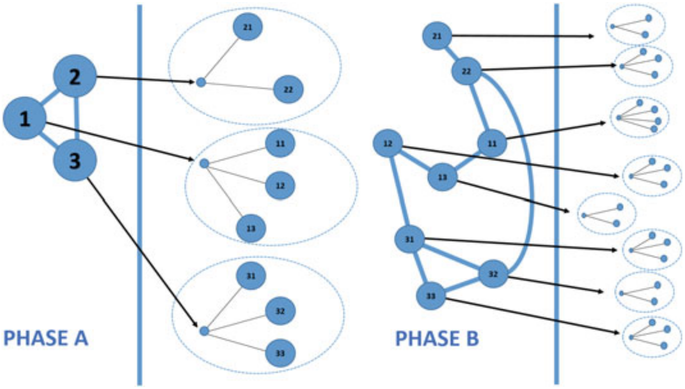


Fig. 15.1 Evolution of the ENM between phase A and B: each node in phase A is decomposed in two or more nodes in phase B. The number of nodes and the mathematical model associated to them depend on the designers' choices. The process is then repeated for the next phases

Design $\mathbf{d}$ and uncertain $\mathbf{u}$ vectors are decomposed in two components: $\mathbf{d} = [\mathbf{d}^d, \mathbf{d}^s]^T$ and $\mathbf{u} = [\mathbf{u}^d, \mathbf{u}^s]^T$ where the former ($[\mathbf{d}^d, \mathbf{u}^d]^T$) are related to the *interlayer decision* process, between a layer and the next one, and the latter ($[\mathbf{d}^s, \mathbf{u}^s]^T$) describe the *intralayer* physical model of the space *system* at a particular level of resolution.

At each layer $\alpha \in \{\text{pre-}A, B, \ldots, F\}$ of the ML-ENM, the performance index can be defined as:

$$f^\alpha(\mathbf{d}, \mathbf{u}) = \sum_{i=1}^{N} g_i^\alpha(\mathbf{d}_i^{\alpha s}, \mathbf{u}_i^{\alpha s}, \boldsymbol{\varphi}_i^{\alpha s}(\mathbf{d}_i^{\alpha s}, \mathbf{u}_i^{\alpha s}, \mathbf{d}_{ij}^{\alpha s}, \mathbf{u}_{ij}^{\alpha s})), \tag{15.9}$$

In Eq. (15.9) N is the number of nodes of the network in layer α and $\boldsymbol{\varphi}_i^\alpha(\mathbf{d}^\alpha, \mathbf{u}_i^\alpha, \mathbf{d}_{ij}^{\alpha s}, \mathbf{u}_{ij}^\alpha)$ is the vector of scalar exchange functions $\varphi_{ij}^\alpha(\mathbf{d}^\alpha, \mathbf{u}_i^\alpha, \mathbf{d}_{ij}^{\alpha s}, \mathbf{u}_{ij}^\alpha)$ that represent the input/output of the nodes, with $j \in J_i^\alpha$, and J_i^α the set of indexes of nodes connected to the i-th node of that layer. Equation (15.9) decomposes the uncertain components $\mathbf{u}^s$ in two categories: the uncoupled components $\mathbf{u}_i^{\alpha s}$ that affect only subsystem i, and the coupled variables $\mathbf{u}_{ij}^{\alpha s}$ shared among subsystem i and one or more subsystems j.

15.6 Problem Formulation

The ML-ENM has been here applied to the design for robustness of a spacecraft through the phases A, B and C (pre-phase A is considered in the figures for clarity). Each node of the ML-ENM is associated to a mathematical function modelling a subsystem or a component. Their list and the classification between the different phases is presented in Table 15.1. The quantity of interest is the overall mass of the satellite and it is given by the sum of the masses of all the subsystems (phase A) or components (phase B and C). The network can be visualised in Figs. 15.2, 15.3, 15.4 and 15.5: the nodes correspond to the models of the system (node 1 at pre-phase A), sub-systems (nodes 2–6 at phase A) and components (nodes 7–18 at phase B and nodes 19-30 at phase C). The links, instead, correspond to their intra-layer and inter-layer connection. In particular coloured arrows define inter-layer (hierarchical) dependencies while grey lines indicate intra-layer dependencies. Red lines show the dependence of nodes at layer A from the node at layer pre-A (pre-$A \rightarrow A$), yellow lines show the dependence of nodes at layer B from nodes at layer A ($A \rightarrow B$) and purple lines of nodes at layer C from layer B ($B \rightarrow C$). Each node in a generic layer, in fact, can be decomposed in two or more nodes in the next layer. Furthermore, the number of parameters and the complexity increase through the process as Table 15.2 shows. Gray lines instead represent couplings between nodes in the same layer α through the linking functions φ_i^α as in Eq. 15.9.

Table 15.1 ML-ENM nodes

Node	Pre-phase A
1	Spacecraft
	Phase A
2	Attitude and Orbit Control (AOCS)
3	Payload
4	Power
5	Thermal
6	Telemetry and Telecommand (TTC)
	Phase B
7	Magnetorquers
8	Thrusters
9	Reaction wheels
10	Payload
11	Batteries
12	Harness
13	Power Conditioning and Distribution Unit (pcdu)
14	Solar array
15	Thermal
16	Antenna
17	Radio Frequency Distribution Network (rfdn)
18	Transponder
	Phase C
19	Magnetorquers
20	Thrusters
21	Reaction wheels
22	Payload
23	Batteries
24	Harness
25	Power Conditioning and Distribution Unit (pcdu)
26	Solar array
27	Thermal
28	Antenna
29	Radio Frequency Distribution Network (rfdn)
30	Transponder

Fig. 15.2 2D representation of the design process as a decision tree. The phases (A, B and C) are indicated with different colours. Subsystem's and component's models are represented as nodes

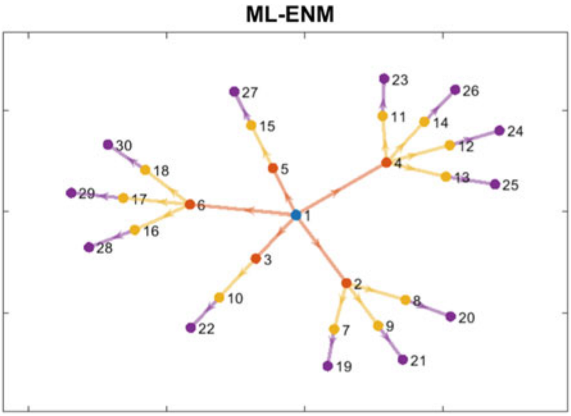


Fig. 15.3 3D representation of the design process as a tree

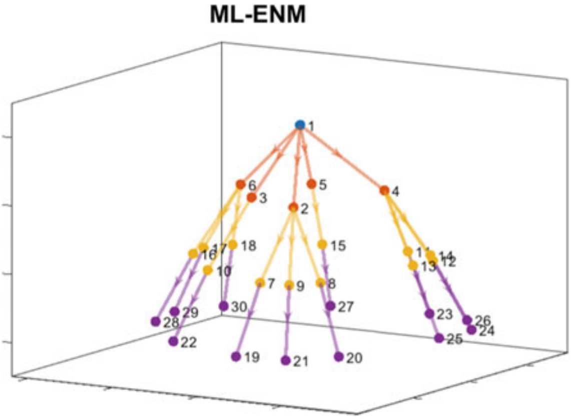
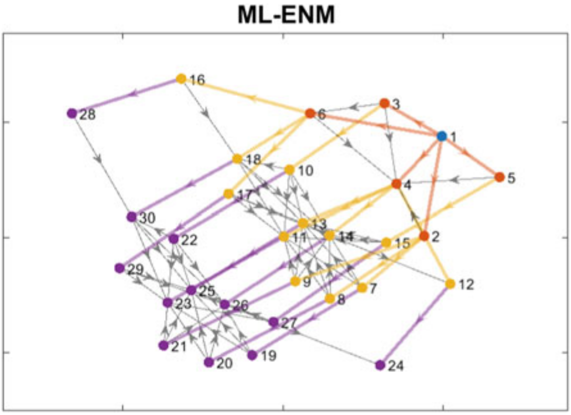


Fig. 15.4 Representation of the design process as a graph. Coloured arrows define inter-layer dependencies while grey lines indicate intra-layer dependencies within the same design phase



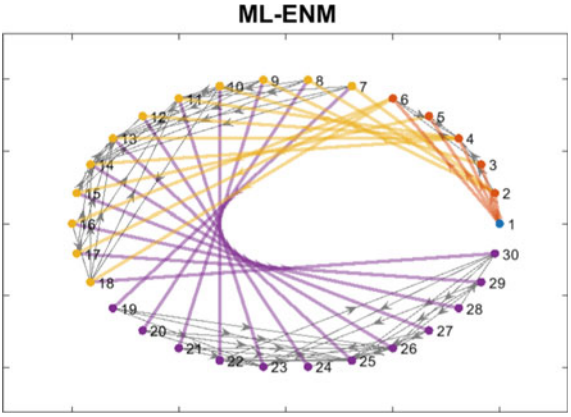


Fig. 15.5 Circular representation of the ML-ENM with both inter-layer and intra-layer dependencies

Table 15.2 Model dimention

Phase	Path 1		Path 2	
	dim_d	dim_u	dim_d	dim_u
A	6	16	6	17
B	13	35	14	34
C	21	43	21	47

15.7 Method

For the defined ML-ENM, the *Body of Evidence* presented in Sect. 15.2 can be populated at the last phase of the ML-ENM, here phase C, by a process of knowledge elicitation. For the proposed application, available data from previous publications has been used [9].

In this example there are only two possible paths that the design process can explore from phase A to phase C. They correspond to the choice between node 7 (*Magnetorquers*) and 8 (*Thruster*) at phase B. The choice brings, respectively, to node 19 and 20 at layer C.

For each chosen path the uncertainty structure at phase C is propagated back to phase A exploiting the inter-layer dependencies A → B and B → C. A minimisation and a maximisation have been run to reconstruct the lower and upper bounds of each uncertain parameter at layer α that incorporate two or more parameters of the layer $\alpha + 1$. In this manner, the reconstructed *Body of Evidence* at phase A incorporates the uncertainty that affect the more complex and detailed models at phase B and C.

Then, the system is optimised for robustness at phase A. In particular, the min-max algorithm is used to evaluate the optimal design vector $\mathbf{d}_A^*$. For more details about the method please refer to Refs. [12, 13].

For the evaluated optimal design solution $\mathbf{d}_A^*$, the decomposition approach presented in Refs. [6, 9–11] has been applied to the ML-ENM in order to propagate uncertainty through the spacecraft model and reconstruct a good approximation of the belief curve with a fraction of the computational cost required for the exact one (Fig. 15.12).

The effect of uncertainty at phases B and C is finally analysed in correspondence with the robust design solution $\mathbf{d}_A^*$.

15.8 Results

Considering the first path in the ML-ENM (node 7 at phase B and node 19 at phase C), the worst case optimal solution $(\mathbf{d}_1^*, \mathbf{u}_1^*)$ gives a mass of 166.43 kg. Figures 15.6, 15.7 and 15.8 show the effect of uncertainty at phases A, B and C for the fixed $\mathbf{d}_1^*$. The second path (node 8 at phase B and node 20 at phase C) brings to the robust solution $(\mathbf{d}_2^*, \mathbf{u}_2^*)$ with a corresponding mass of 230.12 kg. Figures 15.9, 15.10 and 15.11 show the effect of uncertainty at phases A, B and C for the fixed $\mathbf{d}_2^*$. In particular, Figs. 15.6 and 15.9 concern the system level (the whole mass of the satellite), Figs. 15.7 and 15.10 the sub-systems level and Figs. 15.8 and 15.11 the components level. The

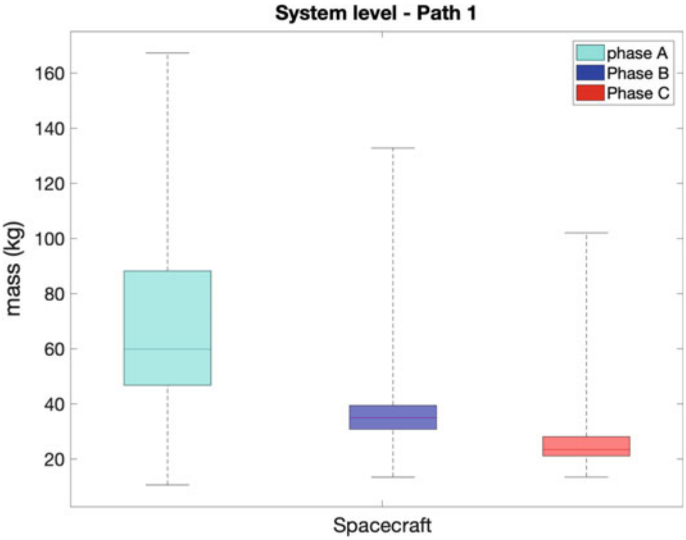


Fig. 15.6 Effect of uncertainty at the system's level in phases A, B and C for the first considered path. The design vector is fixed at the optimal solution

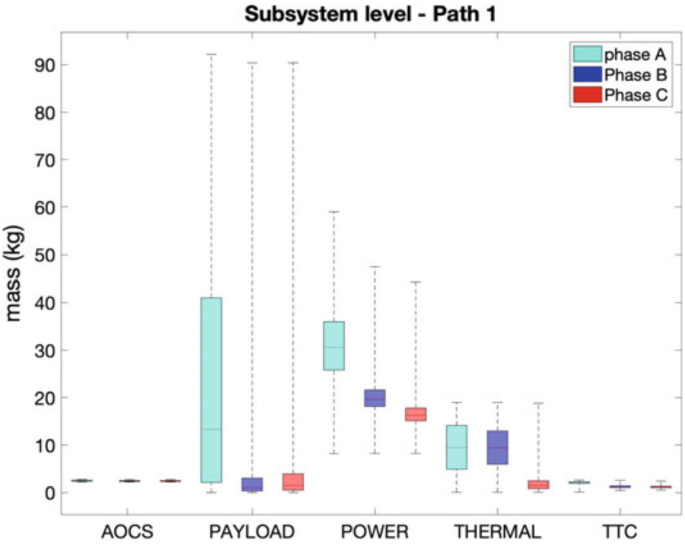


Fig. 15.7 Effect of uncertainty at the sub-system's level in phases A, B and C for the first considered path. The design vector is fixed at the optimal solution

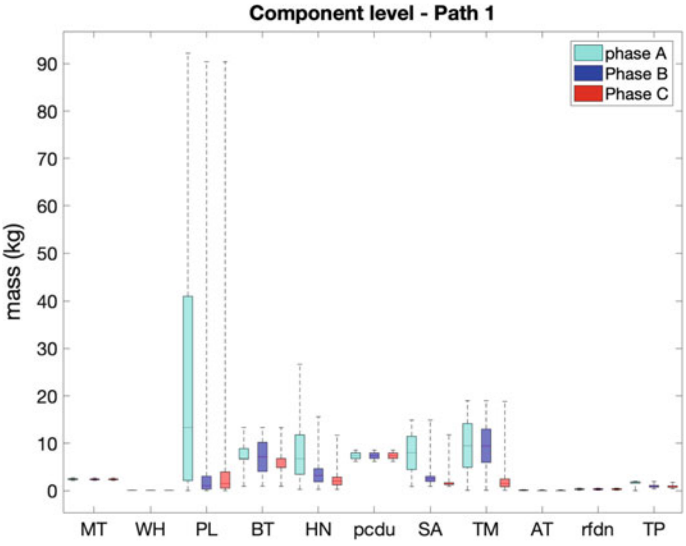


Fig. 15.8 Effect of uncertainty at the component's level in phases A, B and C for the first considered path. The design vector is fixed at the optimal solution

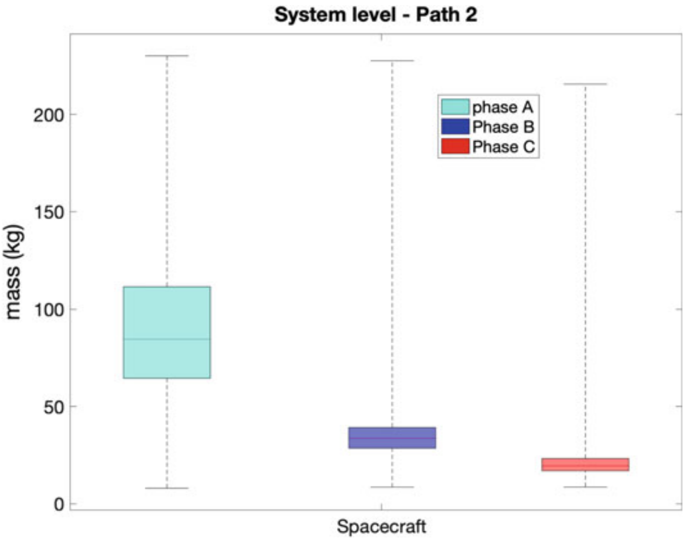


Fig. 15.9 Effect of uncertainty at the system's level in phases A, B and C for the second considered path. The design vector is fixed at the optimal solution

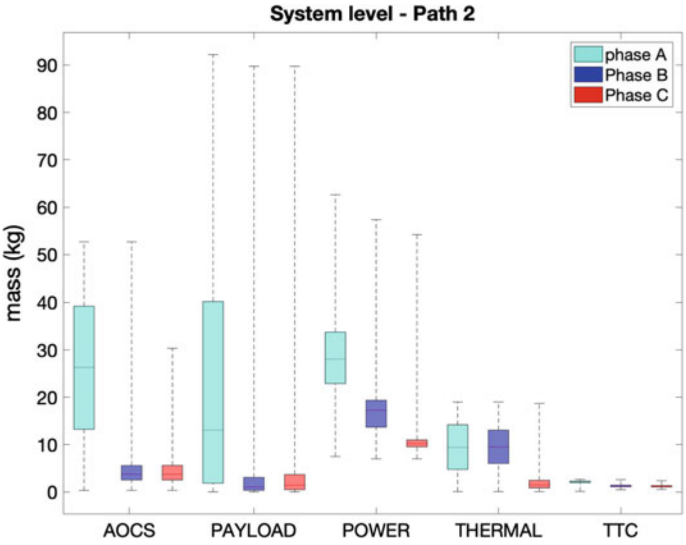


Fig. 15.10 Effect of uncertainty at the sub-system's level in phases A, B and C for the second considered path. The design vector is fixed at the optimal solution

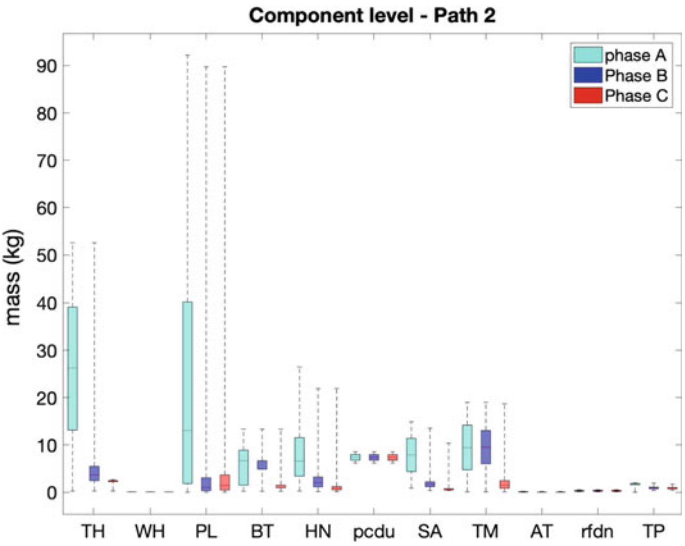


Fig. 15.11 Effect of uncertainty at the component's level in phases A, B and C for the second considered path. The design vector is fixed at the optimal solution

boxplots have been evaluated with a Monte Carlo simulation over the uncertain space with 10^6 function evaluation. For each box a maximisation and a minimisation have been run in order to be sure that the boxes include all the possible values of the mass for the given uncertainty structure. These figures show that the spacecraft model at phase A with the back-propagation of uncertainty, incorporates for the chosen path all the uncertainty in phases B and C. The worst case optimal solution $\mathbf{d}^*$ at phase A, then, results to be robust through the design process.

For $\mathbf{d}_1^*$, finally, the decomposition approach has been applied to the ML-ENM and Fig. 15.12 presents the reconstructed belief curve. The decomposition method allows to a fast and good evaluation of the belief as demonstrated in Ref. [10]. In this problem, in fact, the exact evaluation of the belief curves require 65536 maximisations (one for each focal element). Instead, the curve in Fig. 15.12 has been evaluated with 234 maximisations (0.36 %).

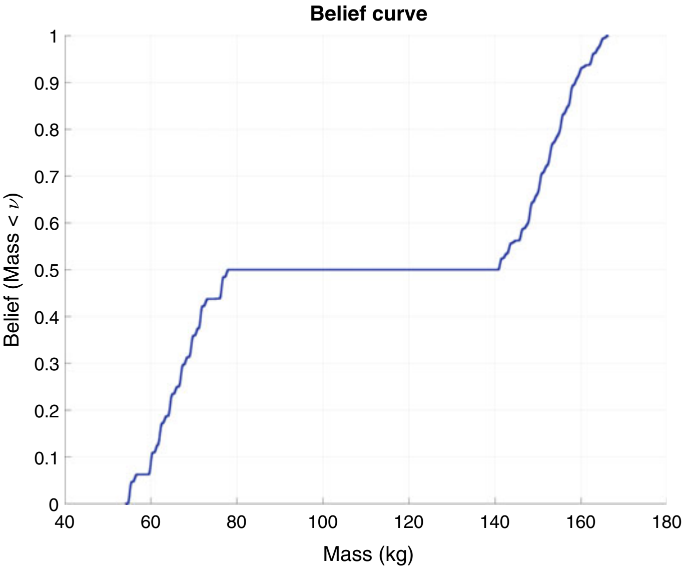


Fig. 15.12 Cumulative Belief curve of the optimal worst case solution at phase A for the first considered path

15.9 Conclusion

This paper proposes a new approach for the design process of a space system affected by epistemic uncertainty. The main novelty is given by the use of the ML-ENM to quantify and propagate uncertainty between different phases of the design process. ML-ENM is a multi-layer network representation of the complex system where each layer takes into account the couplings between subsystems (or components) at a particular design phase. The evolution of the design process is then modelled by the sequence of layers.

It is here presented a method for the definition of uncertainty at the first phase (phase A) of the process such that the optimal solution at that phase is robust against the uncertainty in the following phases.

The method is applied to the design of a space system. The model is optimised for robustness and finally a decomposition methodology is applied to the network in order to reduce the computational cost of the epistemic uncertainty propagation and the belief reconstruction with the use of Evidence Theory.

It has be shown that the optimal design solution at phase A defined in such a way, is robust against the propagation of uncertainty through the design process.

References

1. Pedersen P, Laursen CL (1982) Design for minimum stress concentration by finite elements and linear programming. J Struct Mech 10(4):375–391
2. Nicolich M, Cassio G (2014) System models simulation process manangement and collaborative multidisciplinary optimization. In: INCOSE Italian Chapter Conference on Systems Engineering (CIISE2014). Rome, Italy
3. Beyer H-G, Sendhoff B (2007) Robust optimisation: a comprehensive survey. Comput Methods Appl Mech Eng 196:3190–3218
4. Zio E (2009) Reliability engineering: old problems and new challenges. Reliab Eng Syst Saf 94(2):125–141
5. Helton JC, Johnson JD, Oberkampf WL, Sallaberry CJ (2010) Representation of analysis results involving aleatory and epistemic uncertainty. Int J Gener Syst 39(6):605–646
6. Filippi G, Vasile M, Krpelik D, Korondi PZ, Marchi M, Poloni C (2019) Space systems resilience optimisation under epistemic uncertainty. Acta Astronaut
7. Shafer G (1976) A mathematical theory of evidence. Princeton University Press, Princeton
8. Division S (2010) Space engineering: engineering design model data exchange (CDF). Technical report ECSS-TM-E-10-25A 20 October 2010 First
9. Alicino S, Vasile M (2014) Evidence-based preliminary design of spacecraft. In: 6th international conference on systems concurrent engineering for space applications. Vaihingen Campus, University of Stuttgart, Germany
10. Vasile M, Filippi G, Ortega C, Riccardi A (2017) Fast belief estimation in evidence network models. In: EUROGEN. Madrid
11. Filippi G, Marchi M, Vasile M, Vercesi P (2018) Evidence-based robust optimisation of space systems with evidence network models. In: 2018 IEEE Congress on Evolutionary Computation (CEC). Rio de Janeiro, IEEE, pp 1–8
12. Vasile M (2014) On the solution of min-max problems in robust optimization. In: The EVOLVE international conference. Jian-Guo Hotel, China
13. Filippi G, Vasile M (2019) A memetic approach to the solution of constrained min-max problems. In: IEEE congress on evolutionary computation. Wellington, New Zealand
14. Vasile M (2005) Robust mission design through evidence theory and multiagent collaborative search. Ann N Y Acad Sci 1065:152–173
15. Alicino S, Vasile M (2014) An evolutionary approach to the solution of multi-objective min-max problems in evidence-based robust optimization. In: Proceedings of the 2014 IEEE congress on evolutionary computation, CEC 2014
16. Wertz J, Larson W (1999) Space mission analysis and design, 3rd edn. Microcosm Press
17. NASA NPR7120.5. NPR 7120 . 5, NASA Space Flight Program and Project Management Handbook. NASA's Procedural Requirements (2010)
18. Hirshorn S (2016) NASA system engineering handbook SP-2016-6105 Rev2. 297
19. Hu X, Chen X, Lattarulo V, Parks GT (2015) Multidisciplinary optimization under high-dimensional uncertainty for small satellite system design

Chapter 16
Solving Multi-objective Optimal Design and Maintenance for Systems Based on Calendar Times Using NSGA-II

Andrés Cacereño, Blas Galván, and David Greiner

Abstract Due to technical progress and business competition, design alternatives and maintenance strategies have to be contemplated to optimize the performance of physical assets when new facilities are projected and built. That combined optimization (Design & Maintenance) is required by all industrial installations to develop their activity in an increasingly competitive environment. The Design and Maintenance combined optimization process is a complex problem which requires research and development. The objectives to optimize are Unavailability (due to production losses) and Maintenance Cost (due to overcharge when it is not optimal). The Design and Maintenance strategy for a technical system are optimized jointly by modifying its Functionability Profile, which is closely related to the system's availability. The Functionability Profile is generated by applying Monte Carlo Simulation that allows characterizing the process' randomness until the failure and to modify that Functionability Profile by the optimal Maintenance strategy. An application case is presented, where several configurations of the elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) are used to optimize the multi-objective problem, successfully finding non-dominated solutions with optimum performance for the simultaneous Design and Maintenance strategy combination.

Keywords Reliability · Availability · Multi-objective optimization · Preventive maintenance · Design · NSGA-II · Evolutionary multi-objective algorithms

A. Cacereño (✉) · B. Galván · D. Greiner
Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain
e-mail: acacereno@iusiani.ulpgc.es

B. Galván
e-mail: blas.galvan@ulpgc.es

D. Greiner
e-mail: david.greiner@ulpgc.es

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_16

16.1 Introduction

System Reliability is defined as the probability of being operating under particular conditions during a certain period [1]. The problem of systems design optimization based on their Reliability has been dealt by several authors, both in single-objective [2, 3] and multi-objective [4] cases, as an application of the well-known use of evolutionary algorithms/metaheuristics to solve complex problems in engineering design [5, 6]. However, it is still a live problem because of technical advances, the increase in the complexity of systems and the demand of consumers (among other aspects) [7].

The parameter which includes the process until the failure and recuperation for repairable systems is Availability. In repairable systems, information about the probability of being available at certain time to achieve their functions is given by their Availability.

System's Availability can be deduced through its Functionability Profile. An example of Functionability Profile is shown in Fig. 16.1. The better the system Reliability is, the better its Availability will be. A priority objective in the industry is to obtain the maximum availability because if a system is "available", resources will be being generated. However, when a system is not "available", not only resources are not being generated, but also resources are being consumed until to recover the "available" state. When the system is not "available", it is driven into unproductive phase [8].

The main reasons why a continuous operation system stops are a failure (after that, a recovery time is required) or a scheduled stop to perform a maintenance activity. The global improvement of system's Reliability and Availability is possible through preventive maintenance [9]. If a preventive maintenance activity is performed, the unproductive phase will be more controlled than when reparations have to be performed because of a failure. Therefore, it is interesting to identify the optimum moment to make a stop to develop a preventive maintenance activity. In an ideal way, it has to be done before the occurrence of a failure but as near as possible to maximize the system's "available" time. The Maintenance Optimization problem has been studied extensively [10].

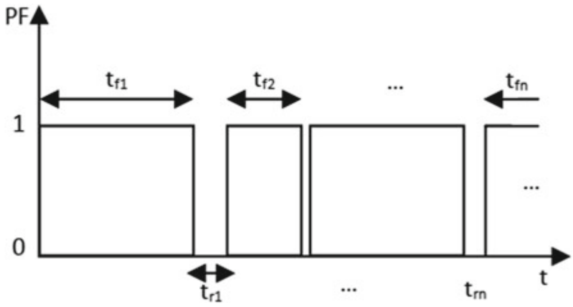


Fig. 16.1 Functionability profile of a component (or device, or system)

From the foregoing, it can be deduced that both the system's optimum design and maintenance strategy improve its Reliability and Availability. Traditionally, the problem of improving Reliability by optimizing the system's design and maintenance strategy has been treated separately.

However, there are some works in which they have been jointly studied. In C.P. de Paula et al. [11] system's Availability and Cost are optimized through a decision process in which the number of redundant elements for a system (design) and the percentage of total resources allocated to maintain it are decided.

In the present paper, we face an unpublished problem, where the multi-objective optimization problem of minimization of the cost and maximization of the availability (or minimization of the unavailability) are handled: a set of optimal balanced solutions between Availability and Cost are provided, on the one hand, from the elements potentially included in the design, and on the other hand from the identification of the optimum moment in which the maintenance activity has to be performed. To obtain that, Functionability Profiles for system's devices have to be readjusted and, consequently, the system's Functionability Profile. Those Functionability Profiles, which are built and adjusted by using Discrete Events Simulation, are product of the Design and Maintenance Strategy.

The paper is organized as follows. Section 16.2 resumes the Methodology. Section 16.3 presents an application case. In Sect. 16.4 results are shown, and finally Sect. 16.5 introduces conclusions.

16.2 Methodology

16.2.1 Availability and Functionability Profile

Reliability is an intrinsic characteristic to a component (or device, or system, depending on disaggregation level, from now on device) which is related to the way in which the device has been designed and built. Maintainability can be intrinsic to devices when it is related to conditions of design (a piece that is difficult to access will be more complex to maintain) or extrinsic, for example, when it is related to availability of spares or to human team who has to perform the maintenance operation.

In Availability, those two parameters (Reliability and Maintainability) are related to define the way in which the device is able to fulfill the function for which it was designed during a period. In the present paper, the system's Availability is characterized by using its Functionability Profile. An example of Functionability Profile is shown in Fig. 16.1.

Functionability Profiles depend on times to failures (t_{f1}, t_{f2},…, t_{fn}) and recovery times (..,…, t_{rn}). In continuous operation devices, when Functionability Profiles are set to logical 1, it is considered that devices are operating. Conversely, when Functionability Profiles are set to logical 0, it is considered that devices are stopped (they are being maintaining or repairing after the failure). It is possible to deduce from Fig. 16.1 that after an operation time (time to failure or time to perform a scheduled maintenance activity), a recovery time is necessary (time to repair after failure or time to perform a preventive maintenance activity).

As previously mentioned, Availability is tightly related to Functionability Profiles. Availability is characterized through the relation between device's operation times and the hoped operation total time for that device. That device will be able to fulfill its purpose during t_f times, so it is possible to characterize Availability $A(t)$ by using Eq. 16.1.

This approximation to characterize the Availability is called Operational Availability. Andrews and Moss [12] explain that Availability is an important measure of performance for repairable devices, which is represented in Eq. 16.2.

$$A(t) \cong \frac{t_{f1} + t_{f2} + \cdots + t_{fn}}{t_{f1} + t_{f2} + \cdots + t_{fn} + t_{r1} + t_{r2} + \cdots + t_{rn}} \tag{16.1}$$

$$A(t) = \frac{MTTF}{MTTF + MTTR} \tag{16.2}$$

Mean Time To Failure ($MTTF$) and Mean Time To Repair ($MTTR$) are distinguished in Eq. 16.2. The approach shown in Eq. 16.2 is the base of the approximation that allows using Eq. 16.1. Availability ($A(t)$) is a variable with value between 0 and 1. The opposite of Availability is Unavailability ($Q(t)$), so $A(t) + Q(t) = 1$ and $Q(t) = 1 - A(t)$.

A priori, operation and recovery times are not known. They are random variables so they allow a statistical treatment. If a historic of both times is compiled and a statistical analysis is performed, these variables could be defined as probability density functions and probability distribution functions through their respective parameters. Functions can arise from a specific typology (exponential, Weibull, normal, for example). There are several Data Bases in the market (OREDA [13], CCPS [14]) which supply the characteristic parameters for the refereed functions, so operation and recovery times can be characterized for different failure modes of devices.

The economic Cost is a variable directly associated to recovery times. When systems are operating, economic income is generated. Conversely, when systems are recovering, economic cost is generated to return it to its operation state. If we want to avoid long recovery times, it is necessary to carry out a preventive maintenance activity ideally before the failure. Because of that stop is scheduled (for reasons such as human personnel are willing and trained, or spare parts are available) recovery times will be shorter. Therefore, it is possible to modify Functionability Profiles for system's devices by including preventive maintenance activities.

16.2.2 Building Functionability Profiles

As we want to analyze the system's Availability, we are going to show how it is possible to build Functionability Profiles for devices by using Discrete Events Simulation. With this end, information about how to characterize operation times to failure (TF) and recovery times after failure (TR) is needed. Characteristic parameters about their probability distribution laws are needed. In this book chapter, all possible device's failures are grouped in a unique failure mode. From the characterization of probability density and probability distribution functions both for operation times (TF) and recovery times (TR), Functionability Profiles for system's devices will be built by generating random times (Discrete Events Simulation). To modify Functionability Profiles, attending to preventive maintenance activities, operation times to preventive maintenance (TR) and recovery times due to preventive maintenance (TRP) will be introduced by generating random times. The process is shown below:

1. System's Life Cycle has to be decided and then, the process continues for all devices.
2. The device's Functionability Profile has to be initialized.
3. A time to preventive maintenance (TP) is extracted from the individual of the population that is being evaluated and a recovery time for preventive maintenance (TRP) have to be randomly generated, between limits previously fixed.
4. Attending to the device's distribution probability law, an operation time to failure (TF) has to be randomly generated, between limits previously fixed.
5. If TP < TF, a preventive maintenance activity is performed before a failure occurs. In this case, as many logical "ones" as TP units followed by as many logical "zeros" as TRP units have to be added to the device's Functionability Profile.
6. If TP > TF, a failure occurs before a preventive maintenance activity would be done. In this case, attending to the device's distribution probability law, a recovery time after failure (TR) has to be randomly generated, between limits previously fixed. Then, as many logical "ones" as TF units followed by as many logical "zeros" as TR units have to be added to the device's Functionability Profile.
7. Steps 4 to 6 have to be repeated until the end of the device's Life Cycle.
8. Steps 2 to 7 have to be repeated until Functionability Profiles have been built for all devices.
9. After to build Functionability Profiles, attending to the logic due to the serial (AND) or parallel (OR) distribution for the system's devices, the system's Functionability Profile has to be built.
10. Finally, system's Availability will be established by using Eq. 16.1, while the system operation cost is computed by adding partial costs due to recovery times.

Economic costs due to recovery times after failure and for preventive maintenance activities have to be established. With this purpose, a cost will be associated to unavailable time units. That cost will be bigger for recovery times after failure due to lack of foresight. The cost has to be computed while device's Functionability Profiles are built.

16.2.3 Multi-objective Optimization

Optimization results useful in practically all areas of our life. Our activities have to be optimized when we want to get the best possible result. However, when we have to solve complex problems we become aware of the suitability of employing that methodology. Optimization is very useful specially when the number of potential solutions is high and getting the best solution is very difficult. However, it will be possible to obtain sufficiently good solutions [15].

Optimization problems can be minimized or maximized for one or more objectives. In most cases, real world problems present various objectives for optimising at the same time (frequently in conflict each other). These problems are so-called "multi-objective" and their solutions arise from a solution set which represent the best compromise between objectives (Pareto optimal set) [16, 17]. This kind of problems are described by Eq. 16.3 (considering a minimization problem in this case) [15].

$$\min_{x} f(x) = \min_{x}[f_1(x), f_2(x), \ldots, f_k(x)] \tag{16.3}$$

In Optimization problems defined by this way, the k functions have to be optimized at the same time. Classical optimization methods suggest converting the multi-objective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at time. Due to their ability to find multiple Pareto-optimal solutions in one single simulation run, a number of multi-objective evolutionary algorithms (MOEAs) were suggested after. In this paper, a MOEA is used to optimize an application problem. This algorithm is the so-called Non-dominated Sorting Genetic Algorithm II [18] (NSGA-II). The selection method in this algorithm is based on the concept of non-dominance.

In this paper, the problem is to optimize the Design and Maintenance strategy for an industrial system based on two different objectives in conflict, Availability and Cost. We wish maximum Availability and minimum maintenance Cost. The more investment in maintenance, the greater system's Availability will be obtained. However, this policy implies a higher unwanted cost, being this the conflict between objectives. Not only maintenance strategy is considered but also the system's design is optimized too based on Availability and its influence in Costs due to Maintenance strategy. The process is discussed below.

16.3 Application Case

The proposed methodology has been applied to a fluid injection system from industry, based on[4] as an example. That system is basically formed by cut valves (V_i) and impulsion pumps (P_i) as is shown in Fig. 16.2.

As it was exposed above, optimization objectives are, on the one hand, to maximize the system's Availability and, on the other hand, to minimize Costs due to system's unproductive phases (both because the system is being recovered and because the system is being maintained). To do that:

- For all system's devices, the optimum moment to perform a preventive maintenance activity has to be established.
- Including redundant devices as P2 and/or V4 has to be decided by evaluating Design alternatives. Including redundant devices will improve the system's Availability but it will worsen its Maintenance Cost.

Population individuals for the Optimization process will be characterized by its chromosome. Chromosomes will be shaped by real number strings with 0 as minimum value and 1 as maximum value (decision variables). They will be codified as $[B_1 B_2 T_1 T_2 T_3 T_4 T_5 T_6 T_7]$, where the presence of redundant devices, P2 and V4, is decided by B_1 and B_2, respectively, and optimum times to perform a preventive maintenance activity to devices are represented by T_1 to T_7. Data set for system's devices used to the optimization process are shown in Table 16.1.

The Software Platform PlatEMO [19] (programmed in MATLAB) was used to optimize the problem. The open source platform PlatEMO includes more than 50 multi-objective evolutionary algorithms, more than 100 multi-objective test problems, along with several widely used performance indicators. In this case, the reliability and maintenance analysis software has been developed and implemented to solve the problem described above in the platform.

The parameters set used to configure the simulation process is shown in Table 16.2. The evolutionary multi-objective algorithm used in this paper is the so-called Non-dominated Sorting Genetic Algorithm II (NSGA-II), a method based on the concept of non-dominance. The method was configured with several parameters. All cases were running five times with a stopping criterion of 5,000,000 evaluations, with Simulated Binary Crossover (SBX), and crossover distribution and mutation distribution indexes of 20. Two population sizes were analysed with 50 and 100 individuals. Mutation probabilities were changed between 0.5, 1 and 1.5 genes per chromosome

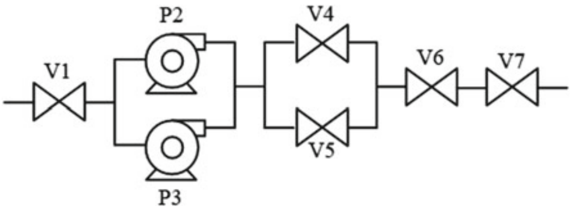


Fig. 16.2 Application case: fluid injection system

Table 16.1 Data set for the system's devices

Parameter	Quantification	Source
Life Cycle	700,800 h	–
Recovery Cost	0.5 units	Expert judgement
Maintenance Cost	Recovery Cost/4	Expert judgement
Pump TF min	1 h	Expert judgement
Pump TF max	70,080 h	Expert judgement
Pump λ Exponential Law	$159.57 \cdot 10^{-6}$ h	OREDA 2009
Pump TR min	1 h	Expert judgement
Pump TR max	24.33 h	$\mu + 4\sigma$
Pump TR μ Normal Law	11 h	OREDA 2009
Pump TR σ Normal Law	3.33 h	(μ–TRmin)/3
Pump TP min	2,920 h	Expert judgement
Pump TP max	8,760 h	Expert judgement
Pump TRP min	4 h	Expert judgement
Pump TRP max	8 h	Expert judgement
Valve TF min	1 h	Expert judgement
Valve TF max	70,080 h	Expert judgement
Valve λ Exponential Law	$44.61 \cdot 10^{-6}$ h	OREDA 2009
Valve TR min	1 h	Expert judgement
Valve TR max	20.83 h	$\mu + 4\sigma$
Valve TR μ Normal Law	9.5 h	OREDA 2009
Valve TR σ Normal Law	2.83 h	(μ – TRmin)/3
Valve TP min	8,760 h	Expert judgement
Valve TP max	35,040 h	Expert judgement
Valve TRP min	1 h	Expert judgement
Valve TRP max	3 h	Expert judgement

Table 16.2 Simulation configuration parameters

Parameter	Configuration
Method	NSGA-II
Evaluations	5,000,000
Population	50–100
Crossover probability	1
Crossover distribution index	20
Mutation probability	0.055–0.111–0.166
Mutation distribution index	20
Executions	5

(0.055, 0.111 and 0.166 respectively). Six cases (combination of two population sizes and 3 mutation rates) were finally evaluated.

16.4 Results

The different configurations for the optimization method were executed five times each. The Hypervolume [20] (HV) average value evolution (among five executions and for each configuration) is shown in Fig. 16.3. The higher the number of evaluations, the higher the improvement of the Hypervolume is observed.

The detail of the last evaluations is shown in Fig. 16.4. It is possible to check that the parameters configuration with population of 100 individuals and mutation probability of 0.055 (0.5 gen per chromosome) finally presents the higher Hypervolume average value.

The values of the main measures obtained for the final evaluations are shown in Table 16.3. These are the Average, Median, Minimum Value, Maximum Value and Standard Deviation of the Hypervolume metric. Firstly, the parameters configuration with population of 50 individuals and mutation probability of 0.055 (0.5 gen per chromosome) presents the higher median of the Hypervolume value. Secondly, the parameters configuration with population of 100 individuals and mutation probability of 0.055 (0.5 gen per chromosome) presents the higher average and minimum of the Hypervolume value. Thirdly, the parameters configuration with population of 100 individuals and mutation probability of 0.111 (1 gen per chromosome) presents the higher maximum of the Hypervolume value. Finally, the parameters configuration

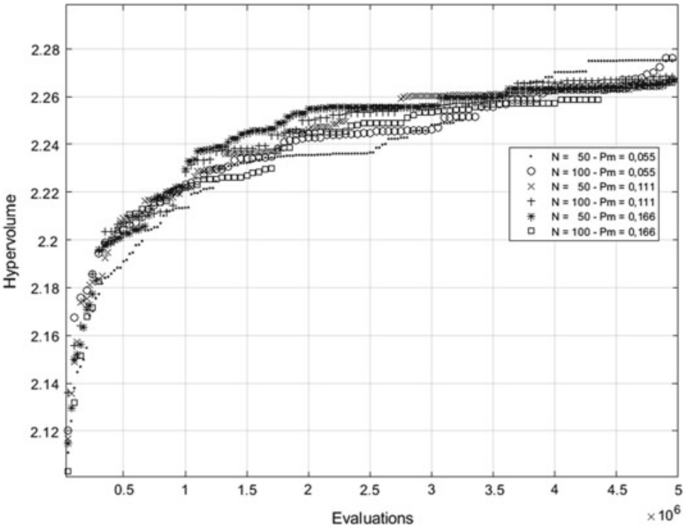


Fig. 16.3 Hypervolume average value evolution

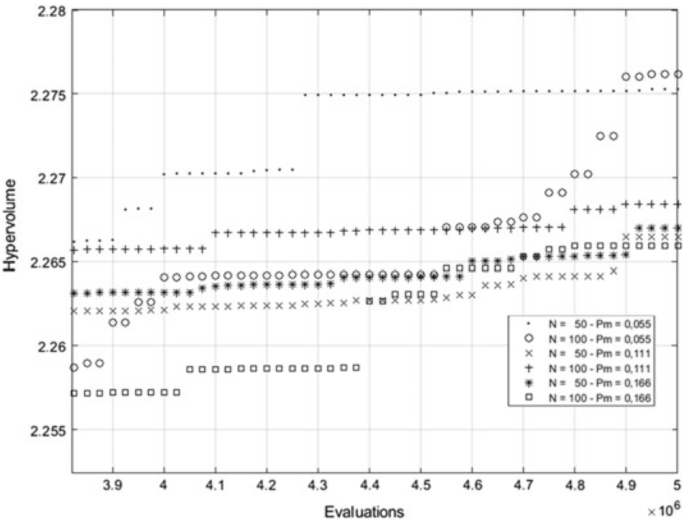


Fig. 16.4 Hypervolume average value evolution (detail)

with population of 50 individuals and mutation probability of 0.166 (1.5 gen per chromosome) presents the lowest standard deviation of the Hypervolume value.

Box plots of the final Hypervolume value distribution for the last evaluation are shown in Fig. 16.5. It is possible to observe some details described above, related to average, median, minimum, maximum and the standard deviation of the final Hypervolume values. The parameters configuration with population of 50 individuals and mutation probability of 0.055 presents the highest median of the Hypervolume value. The parameters configuration presents the higher minimum of the final Hypervolume value. The parameters configuration with population of 100 individuals and mutation probability of 0.111 presents the highest maximum of the final Hypervolume value. The parameters configuration with population of 50 individuals and mutation probability of 0.166 presents the lowest standard deviation of the final Hypervolume value.

In order to establish if any of the six parameter configurations works better than others, a statistical significance hypothesis test was conducted. Particularly, the procedure starts detecting significant differences among the results obtained by applying the Friedman's test. It responds the question: *"Are there results with different median?"* When there are two or more result sets, the null hypothesis (H_0) claims that median are equals (no differences among methods). If H_0 is rejected, differences among methods exist, and a post hoc test is run in order to find the concrete pairwise comparisons which produce differences. In our case, the average rank computed through the Friedman's test is shown in Table 16.4.

The parameters configuration with population of 100 individuals and mutation rate of 0.055 presents the lowest average rank computed through the Friedman's test (the best in this case, as a maximization problem is analyzed -maximum Hypervolume

Table 16.3 Hypervolume statistics of the optimization results

ID	Parameter	N = 50—Pm = 0.055
1	Average	2.2753
	Median	**2.2775**
	Minimum Value	2.2572
	Maximum Value	2.2879
	Standard Deviation	0.0124
ID	**Parameter**	**N = 100—Pm = 0.055**
2	Average	**2.2762**
	Median	2.2732
	Minimum Value	**2.2676**
	Maximum Value	2.2934
	Standard Deviation	0.0101
ID	**Parameter**	**N = 50—Pm = 0.111**
3	Average	2.2665
	Median	2.2700
	Minimum Value	2.2442
	Maximum Value	2.2909
	Standard Deviation	0.0199
ID	**Parameter**	**N = 100—Pm = 0.111**
4	Average	2.2726
	Median	2.2680
	Minimum Value	2.2542
	Maximum Value	**2.2939**
	Standard Deviation	0.0167
ID	**Parameter**	**N = 50—Pm = 0.166**
5	Average	2.2671
	Median	2.2667
	Minimum Value	2.2558
	Maximum Value	2.2746
	Standard Deviation	**0.0071**
ID	**Parameter**	**N = 100—Pm = 0.166**
6	Average	2.2659
	Median	2.2643
	Minimum Value	2.2413
	Maximum Value	2.2896
	Standard Deviation	0.0173

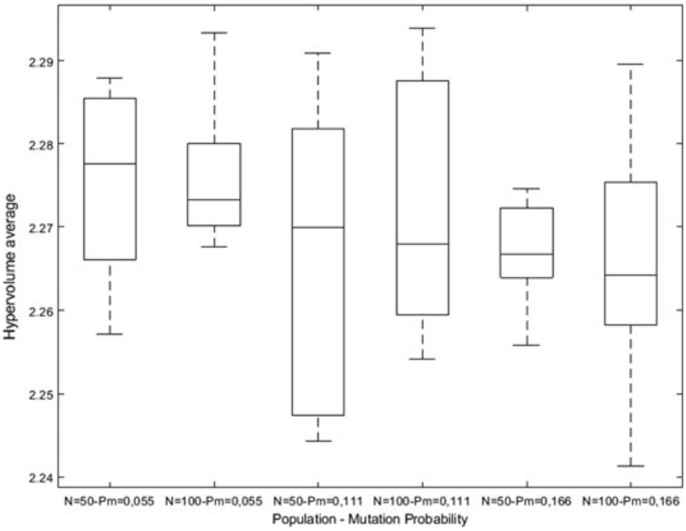


Fig. 16.5 Box plots of the final hyper volume value distribution

Table 16.4 Average rank computed through the Friedman's test (best in bold type)

ID	NSGA-II Configuration	Average Rank
1	N = 50—Pm = 0,055	3.00
2	N = 100—Pm = 0,055	**2.60**
3	N = 50—Pm = 0,111	4.40
4	N = 100—Pm = 0,111	3.20
5	N = 50—Pm = 0,166	3.60
6	N = 100—Pm = 0,166	4.19

is desired-). However, the p-value computed by Friedman's test is 0.6212. This p-value is higher than the level of significance α (0.05) so the null hypothesis *"median are equals"* can't be rejected. This implies it is not possible to establish that any parameter configuration performs better than any other. In the conditions in which the experiment was developed, there aren't significant differences between performances from different configurations. A procedure for conducting multiple comparisons involving all possible pairwise comparisons, as, e.g. described by Garcia S. and Herrera F. in [21], is therefore here not neccesary.

The possible solutions to the problem provided through the last generation of the evolutionary process of the five accumulated executions for all configurations are shown in Fig. 16.6. Some optimum solutions belonging to the obtained non-dominated front are shown in Table 16.5 (these solutions are rounded and numbered in Fig. 16.6). Unavailability is shown in fraction, Cost is shown in economic units and the rest of variables represent, for the respective devices, optimum times to perform a preventive maintenance activity in hours.

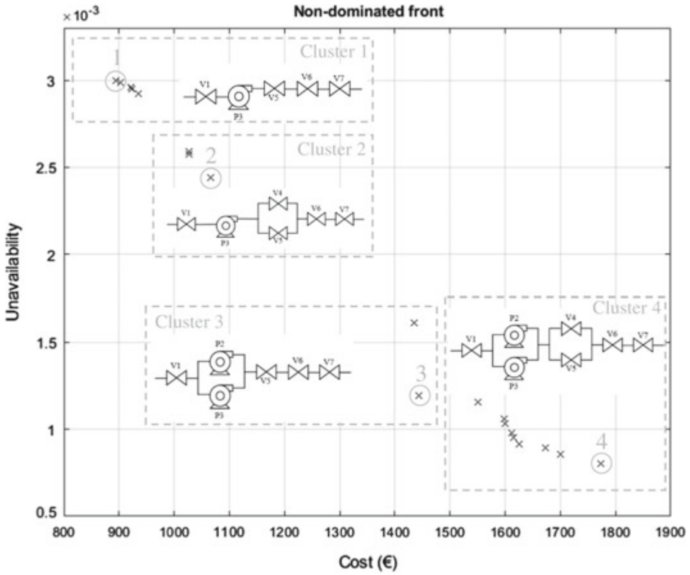


Fig. 16.6 Non dominated solutions (black crosses), and their configuration designs, clustered. Chosen representative solutions (Table 16.5) are additionally circled and numbered

Table 16.5 Sample of some optimum solutions

ID	Unavailability	Cost [un]	V1 [h]	P2 [h]	P3 [h]	V4 [h]	V5 [h]	V6 [h]	V7 [h]
1	0.0029979	894.20	30,887	∄	8,344	∄	29,391	24,051	33,860
2	0.0024687	1,066.92	21,430	∄	8,718	10,299	28,043	31,234	31,442
3	0.0011928	1,443.00	29,592	8,179	8,597	∄	20,005	33,923	29,773
4	0.0008019	1,772.59	34,766	8,386	8,467	29,272	34,531	32,968	31,000

The solution with the lowest Cost (ID1) (894.20 economic units) presents the biggest Unavailability (0.0029979). These values are followed by periodic optimum times (hours) measured from the moment in which the Life Cycle starts (time for performing the preventive maintenance activity (TR) is not included). In that case, it is possible to observe that periodic optimum times to preventive maintenance for devices P2 and V4 are not supplied. It is caused because the design alternative did not consider including such devices. The opposite case shows the biggest Cost (ID4) (1,722.59 economic units) and the lowest Unavailability (0.0008019). In this case, periodic optimum times to perform preventive maintenance activities are supplied for all devices. It is caused because the design alternative considered including devices P2 and V4. Other optimum solutions were found between those two solutions (ID2 and ID3). Decision makers, attending to their requirements, will have to decide which design is the preferable to choose.

Moreover, solutions have been clustered in Fig. 16.6 attending to their final design. Solutions contained by Cluster 1 are the solutions in which non redundant devices have been included in the design. Solutions contained by Cluster 2 are the solutions in which a redundant valve has been included in the design. Solutions contained by Cluster 3 are the solutions in which a redundant pump has been included in the design. Finally, solutions contained by Cluster 4 are the solutions in which both a redundant valve and a redundant pump have been included in the design. Final designs for each Cluster are shown in Fig. 16.6.

16.5 Conclusions

A successful methodology has been presented and demonstrated by a practical test case where proper non-dominated solutions for minimum unavailability and cost objectives have been generated. It has been possible by generating functionability profiles for several designs of the analyzed technical system, using Discrete Events Simulation, and varying those functionability profiles with the inclusion of maintenance activities before the failure. The evolutionary multi-objective algorithm NSGA-II was used to perform the optimization process. This method allowed obtaining optimum solutions attending to the design and maintenance strategy for the technical system. The goal for devices included in the design, was to obtain the sets of optimum times between maintenance activities with the best unavailability-cost relations. A system test case with 7 possible devices was used, including pumps and valves.

A set of different evolutionary multi-objective algorithm parameters configuration has been tested with the purpose of determining its effect in the optimization process. The best non-dominated solutions were archived. A test hypothesis was built with the objective of determining what parameter configuration presents the best performance. It is possible to conclude that significant differences were not found so, in the conditions defined for the experiment, no parameter configuration worked better than any other.

As future work, a comparison among several state of the art evolutionary multi-objective optimizers (EMO) will be performed, including, as stated in e.g. [22], a representative of each of the different three main paradigms of evolutionary multi-objective optimizers attending to their selection method.

Acknowledgments A. Cacereño is recipient of a contract from the Program of training for Predoctoral research staff of University of Las Palmas de Gran Canaria. The authors are grateful for this support.

References

1. Misra KB (2008) Reliability engineering: a perspective. handbook of performability engineering, vol 2008. Springer, pp 253–259
2. Kuo W, Prasad VR (2000) An annotated overview of system-reliability optimization. IEEE Trans Reliab 49(2):176–87
3. Kuo W, Wan R (2007) Recent advances in optimal reliability allocation. Computational intelligence in reliability engineering, vol 2007. Springer, pp 1–36
4. Greiner D, Galván B, Winter G (2003) Safety systems optimum design by multicriteria evolutionary algorithms. Evolutionary multi-criterion optimization. Lecture Notes in Computer Science, vol 2632. Springer, pp 722–736
5. Greiner D, Periaux P, Quagliarella D, Magalhaes-Mendes J, Galván B (2018) Evolutionary algorithms and metaheuristics: applications in engineering design and optimization. Math Probl Eng 2018:1–4
6. Greiner D, Galván B, Périaux P, Gauger N, Giannakoglou K, Winter G (2015) Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences. Computational Methods in Applied Sciences, vol 36. Springer
7. Coit DW, Zio E, The evolution of system reliability optimization. Reliab Eng Syst Saf. https://doi.org/10.1016/j.ress.2018.09.008
8. Boliang L, Jianping W, Ruixi L, Jiaxi W, Hui W, Xuhui Z (2019) Optimization of high-level preventive maintenance scheduling for highspeed trains. Reliab Eng Syst Saf 183:261–275
9. Gao Y, Feng Y, Zhang Z et al (2015) An optimal dynamic interval preventive maintenance scheduling for series systems. Reliab Eng Syst Saf 142:19–30
10. Faddoul R, Raphael W, Chateauneuf A (2018) Maintenance optimization of series systems subject to reliability constraints. Reliab Eng Syst Saf 180:179–188
11. De Paula CP, Visnadi LB, De Castro HF (2019) Multi-objetive optimization in redundant system considering load sharing. Reliab Eng Syst Saf 181:17–27
12. Andrews J D, Moss T R. Reliability and risk assessment 2nd Edition. Professional Engineering Publishing Limited, London and Bury St Edmunds, UK. ISBN 1 86058 290 7
13. OREDA participants. OREDA – Offshore reliability data handbook. 5th Edition. Published by: OREDA participants. Prepared by: SINTEF, Distributed by: Det Norske Veritas (DNV). ISBN 978-82-14-04830-8
14. Center for Chemical Process Safety. Guidelines for process equipment reliability data with data tables. Center for Chemical Process Safety of the American Institute of Chemical Engineers. New York: ISBN 0-8169-0422-7
15. Simon D (2013) Evolutionary optimization algorithms. John Wiley & Sons, Hoboken, New Jersey
16. Coello CA (2015) Multi-objective evolutionary algorithms in real-world applications: some recent results and current challenges. In: Greiner D et al (eds) Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences, Computational Methods in Applied Sciences, vol 36, Springer, pp 3–18
17. Emmerich M, Deutz A (2018) A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat Comput 17(3):585–609
18. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
19. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87
20. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132
21. García S, Herrera F (2008) An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons. J Mac Learn Res 9:2677–2694
22. Greiner D, Periaux P, Emperador J, Galván B, Winter G (2017) Game theory based evolutionary algorithms: A review with nash applications in structural engineering optimization problems. Arch Comput Meth Eng 24:703–750

Chapter 17
Assessment of Exergy Analysis of CFD Simulations for the Evaluation of Aero-Thermo-Propulsive Performance of Aerial Vehicles

Christelle Wervaecke, Ilias Petropoulos, and Didier Bailly

Abstract The purpose of this paper is to present an exergetic approach which provides a good complement to classical drag computation in order to assess aerodynamical performances. Unlike drag methods and, on a more general level, unlike any force-based analysis, no distinction between drag and thrust is required. Thus the exergy approach can be applied to a great variety of novel configurations for which the propulsion system is highly integrated within the airframe such as configurations with boundary layer ingestion for example. It also provides information about thermal effects which can not be extracted from drag computation. This paper aims at giving an insight into the exergetic approach and analysing its sensibilities to numerical parameters such as CFD computation convergence and mesh refinement, an assessment which is important as a basis for the improvement of the method's accuracy.

Keywords Aerial vehicles · CFD simulations · Aeorodynamical performance

17.1 Introduction

Today more than ever, world energy resources seem limited and we have to carefully manage them. This is the obvious reason why future aerial transport concepts will be driven by energy efficiency criteria. This is both an economical and an ecological key

C. Wervaecke (✉)
ONERA Department of Aerodynamics, Aeroelasticity and Acoustics,
The French Aerospace Lab, 92320 ChÃétillon, France
e-mail: christelle.wervaecke@onera.fr

I. Petropoulos · D. Bailly
ONERA Department of Aerodynamics, Aeroelasticity and Acoustics,
The French Aerospace Lab, 92320 Meudon, France
e-mail: ilias.petropoulos@onera.fr

D. Bailly
e-mail: didier.bailly@onera.fr

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_17

issue. The exergy concept has been introduced so as to translate sources of entropy generation into power losses. Power, by definition, is the rate at which energy is consumed. Thus it is an essential information in the question of energetic systems optimization. Moreover, the exergy concept provides a common comparison metric which can be of great interest for multidisciplinary optimisation purposes.

Innovation and improvement require understanding. Nowadays Computation Fluid Dynamics (CFD) has become relevant to predict aerodynamic performances and the farfield drag decomposition method, proposed by Van Der Vooren and Destarac [1], has already provided valuable information about aerodynamical physical phenomena encountered in aircraft aerodynamics. Drag decomposition is now a common practice for aircraft design. Moreover farfield drag extraction approach is still a living research topic. For example, extensions for postprocessing unsteady flow fields have been proposed by Toubin et al. [2] and Gariepy et al. [3] and a new approach based on a Lamb vector approach is currently under study [4, 5]. However, future commercial aircraft are likely to get highly integrated propulsion systems. Thereby, application of farfield drag methods will become difficult as those approaches require a clear separation between thrust and drag.

The need for a tool applicable to very innovative aircraft design or/and highly-integrated propulsion systems along with the research of new methodologies enabling aerodynamic engineers to increase their understanding of physical flows have led the ONERA to develop a new postprocessing tool named FFX [6–8] based on the exergetic approach. This tool is no more based on a mechanical balance as previous mentioned approaches but it is based on an energetic balance.

This paper gives attention to theoretical and numerical aspects of exergy analysis from solutions of the Reynolds-Averaged Navier-Stokes equations. Note that previous work has been presented at the AIAA AVIATION Forum by Petropoulos [9]. It was then a first attempt at analysing numerical behavior of our postprocessing tool and it proposed hints to find way to reduce numerical errors. The work presented in the current paper gives additional cases and analysis in order to deal with numerical error in depth and particularly to give a greater focus on 3D cases. It is organised as follows: the first part addresses the motivations for an exergy approach, the second part describes the exergy decomposition implemented in the FFX tool and the third part examines the relationship between CFD parameters such as simulation convergence or mesh refinement and the accuracy of the FFX decomposition in order to find ways to propose a more robust formulation.

17.2 Why Exergy?

The exergy analysis has been proposed by Arntz [6] as an extension to the energy analysis method introduced by Drela [10]. It is an analysis which is conducted by the coupling of the first and second law of thermodynamics and that provides information about the energy amount that is theorically available. Whereas most previous field analysis methods have focused on forces information as drag and thrust, the exergy analysis focus on the idenfication and quantification of available useful work: sources, sinks and interaction. Thus it provides a very different kind of measurement that can enhance the physic flow comprehension of the aerodynamic engineer. As the formulation does not require any separation between drag and thrust, it can be applied even for high-integrated propulsive system.

To attain the climate global warming targets defined by ACARE (Advisory Council for Aeronautics Research in Europe) [11], there is a major focus from aircraft manufacturers to build more efficient aircraft. Aircraft manufacturers are urged to search for more efficient solutions and even disruptive technologies to reach the performance improvements required. All the subsystems which constitute the aircraft along with their mutual influence have to be taken into account in order to resolve these problems. The energy analysis method, derived from the first law of thermodynamic, is used in the design of aerospace systems. Fundamentally, an aircraft transforms chemical energy in other forms of energy. Kinetic, potential and mechanical work can be considered as conservative form of energy which can be converted into other forms without loss. However, heat, chemical and radiation energy cannot be completely converted in other forms of energy. For example, heat cannot be completely converted into work even from an idealised reversible cycle. So, exergy analysis consists in considering energy as the sum of two components: exergy and anergy. Exergy represents the available mechanical energy while anergy is the part of energy which cannot be transformed due to irreversible processes.

Furthermore, the exergy approach is based on a balance equation derived from the first and the second law of thermodynamics. It is possible to locally evaluate not only the losses associated with each physical phenomenon in the considered system, but also those associated with phenomena occurring outside it and that can be considered as a measure of waste. Overall system improvement can be achieved by reducing losses from internal irreversibilities and generally the total waste of the system. So exergy appears to be well adapted to improve complex systems where different energy transformations take place. This methodology enables to study a complete aircraft as being constituted of subsystems of different nature by using a common metric.

17.3 Formulation

In a nutshell, exergy is the energy that is available and that can be transformed to a useful form of energy. By combining the first and the second laws of thermodynamics, it handles energy and entropy together and can be defined as:

$$\mathcal{E} = \Delta h_i - T\Delta s \tag{17.1}$$

where the term h_i denotes specific total enthalpy, $\Delta h_i = (h_i - h_{i,\infty})$ and the subscript ∞ indicates reference conditions which are usually taken as the atmosphere free-stream flow. The term s denotes entropy and $\Delta s = (s - s_\infty)$. FFX provides

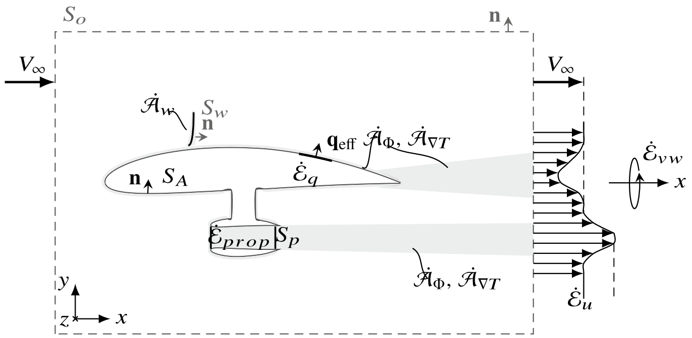


Fig. 17.1 Notations of the exergy decomposition

the exergy-based formulation summarized within the Eq. (17.2) for aeropropulsive performance assessment developed by A. Arntz during his PhD thesis [6]. More details of the theoretical development are available in reference [8]. The general exergy-based formulation can be written as:

$$\dot{\mathcal{E}}_{prop} + \dot{\mathcal{E}}_q = W\dot{\Gamma} + \dot{\mathcal{E}}_m + \dot{\mathcal{E}}_{th} + \dot{A}_{tot} \tag{17.2}$$

Each term of this exergy decomposition will be described more specifically hereafter. Figure 17.1 provides a scheme depicting notations in FFX's framework. It should be noted that the method aims at evaluating the time-averaged changes in exergy hence the introduction of exergy rate terms denoted by $\dot{\mathcal{E}}$. The destroyed exergy has been called anergy and the rate of anergy, i.e. the unusable part of energy, is denoted by $\dot{A}$.

The left hand side of Eq. (17.2) represents the total exergy supplied to the flow by the propulsion system or by heat conduction. The first term $\dot{\mathcal{E}}_{prop}$ is defined as follows:

$$\dot{\mathcal{E}}_{prop} = \int_{S_p} -\rho\delta h_i \mathbf{V}\cdot\mathbf{n}\,dS + T_\infty \int_{S_p} \rho\delta s \mathbf{V}\cdot\mathbf{n}\,dS \tag{17.3}$$

It is the rate of exergy supplied by the propulsion system and S_p is the surface delimiting it. The first part is the power supplied to the flow while the second one represents thermodynamic losses that have occurred within the propulsion system. The other term of supplied exergy, $\dot{\mathcal{E}}_q$ is written as:

$$\dot{\mathcal{E}}_q = \int_{S_A} -\mathbf{q}_{eff}\cdot\mathbf{n}\,dS + \int_{S_A} \frac{T_\infty}{T}\mathbf{q}_{eff}\cdot\mathbf{n}\,dS \tag{17.4}$$

It corresponds to the rate of exergy transfer by heat conduction through the surface of the airplane. The first term is the heat transferred by conduction and the second one is the associated anergy. Both of these integrals are non zero only on non-adiabatic surfaces.

The right hand side of Eq. (17.2) corresponds to the sum of exergy consumed by the airplane, exergy remaining in the flow and exergy destroyed through irreversible processes (which is anergy). First, the term $W\dot{\Gamma}$ is defined as follows:

$$W\dot{\Gamma} \;=\; \int_{S_o} \rho(u - V_\infty)V_\infty \mathbf{V}\cdot\mathbf{n} \;+(p - p_\infty)\mathbf{V}_\infty\cdot\mathbf{n}\,dS \tag{17.5}$$

It represents the mechanical exergy part consumed by the airplane. As the exergy balance equation is written assuming a steady flow, the term $W\dot{\Gamma}$ represents the energy consumed to maintain a steady path: whether cruise, climb or descent. Considering an unpropelled configuration ($\dot{\mathcal{E}}_{prop}$ and $\dot{\mathcal{E}}_q$ are both zero), $W\dot{\Gamma}$ matches with the corresponding drag coefficient when choosing a suitable nondimensionalization. It should be underlined that the above expression provides no distinction between thrust and drag.

Among sources of exergy still available in the flow, the term $\dot{\mathcal{E}}_m$ represents the rate of mechanical exergy:

$$\dot{\mathcal{E}}_m = \dot{\mathcal{E}}_u \;+\dot{\mathcal{E}}_{vw} \;+\; \dot{\mathcal{E}}_p \tag{17.6}$$

where

$$\dot{\mathcal{E}}_u \;= \int_{S_o} \frac{1}{2}\rho u^2\,\mathbf{V}\cdot\mathbf{n}\,dS \tag{17.7}$$

$$\dot{\mathcal{E}}_{vw} \;= \int_{S_o} \frac{1}{2}\rho(v^2 + w^2)\,\mathbf{V}\cdot\mathbf{n}\,dS \tag{17.8}$$

$$\dot{\mathcal{E}}_p \;= \int_{S_o} (p - p_\infty)\,(\mathbf{V} - \mathbf{V}_\infty)\cdot\mathbf{n}\,dS \tag{17.9}$$

It is the sum of the streamwise kinetic energy ($\dot{\mathcal{E}}_u$), the transverse kinetic energy deposition ($\dot{\mathcal{E}}_{vw}$) and a third term described as the exterior pressure-work ($\dot{\mathcal{E}}_p$). The other term of available exergy is the term $\dot{\mathcal{E}}_{th}$ which is the rate of thermal exergy:

$$\begin{aligned}\dot{\mathcal{E}}_{th} = \int_{S_o} \rho\delta e\mathbf{V}\cdot\mathbf{n}\,dS \;+ \int_{S_o} p_\infty\mathbf{V}\cdot\mathbf{n}\,dS \\ -\; T_\infty\int_{S_o} \rho\delta s\mathbf{V}\cdot\mathbf{n}\,dS\end{aligned} \tag{17.10}$$

Finally, the term $\dot{A}_{tot}$ denotes the rate of anergy generation, or equivalently of exergy destruction, by irreversible phenomena which are viscous dissipation (first term), thermal conduction (second term) and shockwaves (third term):

$$\dot{A}_{tot} = \dot{A}_\Phi + \dot{A}_{\nabla T} + \dot{A}_w \tag{17.11}$$

where

$$\dot{A}_{\Phi} = \int_{v} \frac{T_\infty}{T} \Phi_{eff}\, dv \tag{17.12}$$

$$\dot{A}_{\nabla T} = \int_{v} \frac{T_\infty}{T^2} K_{eff} (\nabla T)^2\, dv \tag{17.13}$$

$$\dot{A}_{w} = T_\infty \int_{S_w} \rho \delta s \mathbf{V} \cdot \mathbf{n}\, dS \tag{17.14}$$

Note that all terms of the decomposition (17.2) are nondimensionalized by the coefficient : $\frac{1}{2}\rho_\infty {\mathbf{V}_\infty}^3 S_{ref}$. The term S_{ref} denotes a reference surface.

17.4 Accuracy Assessment

17.4.1 Implementation

The exergy decomposition has been implemented in ONERA's in-house postprocessing tool named FFX. The implementation is strongly coupled with the Cassiopée library [12]. It benefits of its Python/C++ environment and can handle both structured and unstructured meshes along with cell-centered or vertex-centered solutions. As the exergy formulation is still under study, it is necessary to maintain a very flexible implementation framework such as to be able to test and modify rapidly some terms if required. The Cassiopée platform is appropriate for such investigations. Moreover, the FFX tool has already been applied by some of ONERA's industrial partners on complex configurations see Tailliez [13], Couilleaux [14] and Wiart [15]. The first results seem quite encouraging.

17.4.2 Sensitivity Analysis

This section deals with academic applications investigated in order to assess the FFX tool accuracy. All RANS computations were performed with the elsA solver of ONERA [16].

17.4.2.1 NACA0012 Case

In order to assess the sensitivity of the exergy decomposition to mesh refinement and CFD computation convergence, this section focuses on a 2D academic test case: the flow around a NACA0012 airfoil. There is a lot of information for this test case as it was used as a verification test case for Drag Prediction Workshop 5 (DPW-

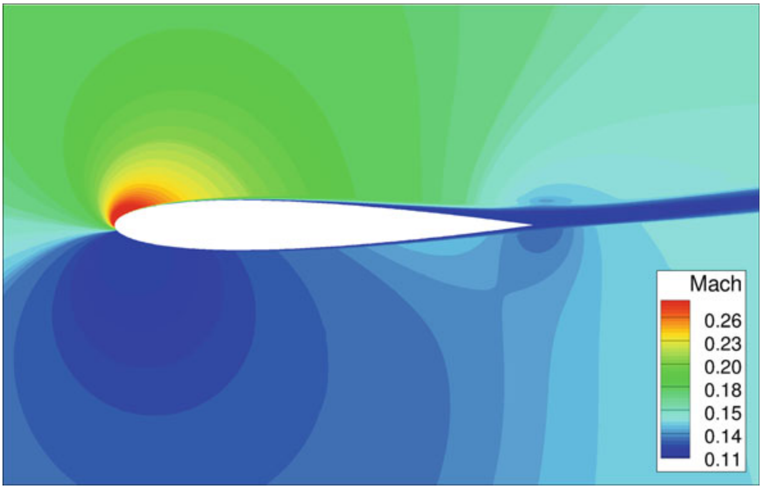


Fig. 17.2 Mach number field around the NACA0012 airfoil

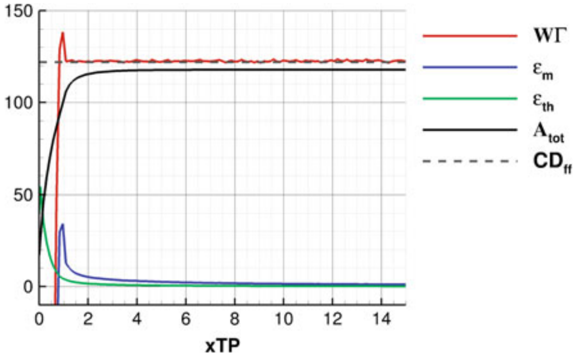
Table 17.1 NACA0012—Size of the 7 nested structured grids

Mesh	1	2	3	4	5	6	7
Ncell	3 584	14 336	57 344	229 376	917 504	3 670 016	14 680 064

5) and Drag Prediction Workshop 6 (DPW-6) [17]. A sequence of nested grids are provided on the NASA Turbulence Modeling Resource website [18], the number of cells ranging from 3 500 to 14.7 10^6 and each coarser grid is exactly composed of every-other-point of the next finer grid. The grids have a farfield extent of about 500c. The reference state conditions are Mach number M = 0.15, the Reynolds number per chord length is Re = 6 million and the angle of attack is $\alpha = 10°$. Figure 17.2 shows the Mach number field around the NACA0012 airfoil and Table 17.1 gives the number of cells for each grid considered in this study.

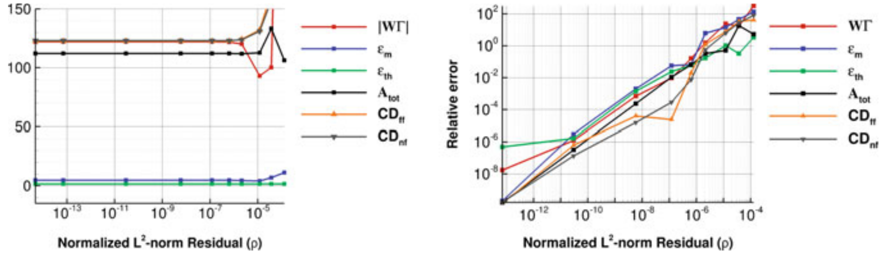
Although the flow around a 2D profile is a very simple case, it provides a very good frame to assess a postprocessing tool accuracy as CFD convergence can be reached (indeed the more complex the case is, the more touchy the convergence becomes) and mesh refinement is easily achieved. Moreover, a deeper understanding of the exergetic decomposition terms is much easier for such a case and gives insight into its physical meanings. As we consider an unpowered configuration, the terms $\dot{\mathcal{E}}_{prop}$ and $\dot{\mathcal{E}}_q$ of the balance Eq. (17.2) are zero. The term $W\dot{\Gamma}$, appropriately nondimensionalized, matches the total drag coefficient. Finally, the amount of exergy still available within the flow is composed of the mechanical and the thermal exergy. These terms decrease in the wake as the amount of anergy, generated by viscous and thermal phenomena (no shockwave is formed at these conditions), increases. Figure 17.3 shows the evolution of the exergy decomposition's terms with respect to the location of the downstream limit of the control volume (driven by the x_{TP} parameter) for mesh 5. There is a good

Fig. 17.3 NACA0012—Mesh 5. Evolution of the exergy decomposition's terms with respect to the location of the downstream limit of the control volume



agreement between the term $W\dot{\Gamma}$ and the farfield drag coefficient cd_{ff}, evaluated by the ONERA far-field drag code, and exergy and anergy terms behave as expected (note that it takes a very long distance for the mechanical exergy to vanish). The rate of mechanical exergy decreases towards zero as the Trefftz plane moves downstream of the body. Its decomposition shows that after some chords the total mechanical exergy reduces to the streamwise kinetic energy of the wake. In the same way, the rate of thermal exergy decreases towards zero. These behaviors are due to irreversible process occurring in the wake (dissipation). These exergy losses are counterbalanced by the increase of the total anergy term.

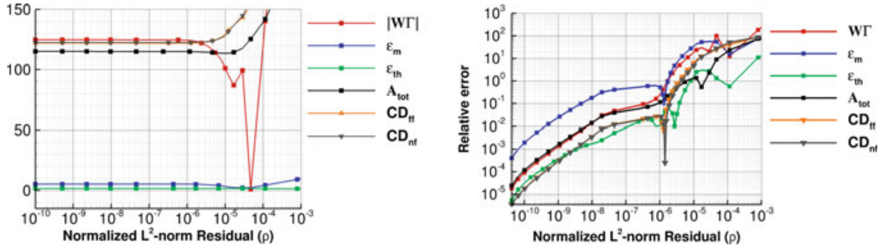
In the first part of this NACA0012 case analysis, the exergetic balance is estimated several times as the residuals of the CFD computation decrease. The aim is to analyse whether some terms are more sensitive to CFD convergence and which level of convergence is required to attain a good exergetic decomposition accuracy. The downstream limit of the control volume is set up at $x_{TP} = 2$ for the following results. The Figures 17.4 and 17.5 provide some results for mesh 3 and mesh 5 (see Table 17.1). The relative error is estimated as the difference between final value, i.e. when CFD convergence is achieved, and current value for each term of the formulation (in percentage). The absolute value of each term is also given so as to indicate their order of magnitude: thermal and mechanical exergy are smaller in absolute value than the other terms. Figure 17.6 gives the residual for the equation of mass conservation with respect to the number of iterations. It shows that a residual around $\sim 10^{-4}$, $\sim 10^{-5}$ are quickly reached. That is why it is the starting point for plots on exergy relative errors. In Figs. 17.4 and 17.5, nearfield (cd_{nf}) and farfield (cd_{ff}) drag coefficients are also given. They have been computed by the ONERA in-house FFD (Far Field Drag) tool. It enables to compare exergy balance and drag evaluation requirements on CFD convergence. When looking for a relative error of 1%, the total anergy term $\dot{A}_{tot}$ behaves as well as cd_{ff} or cd_{nf} on both meshes and even reached this threshold faster than these terms on the coarsest mesh (mesh 3). Smaller levels of relative error are more quickly reached by cd_{ff} and cd_{nf} afterward. It is not obvious from results of mesh 3 but for all finer meshes, the thermal exergy term $\dot{\mathcal{E}}_{th}$ converges faster than the other terms whereas the mechanical exergy term $\dot{\mathcal{E}}_m$ converges a little bit slower.



(a) Values of the FFX decomposition with respect to CFD convergence

(b) Relative error in FFX's terms with respect to CFD convergence

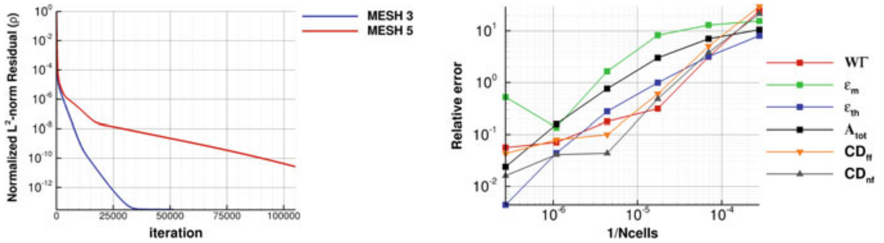
Fig. 17.4 NACA0012—Results on mesh 3



(a) Values of the FFX decomposition with respect to CFD convergence

(b) Relative error in FFX's terms with respect to CFD convergence

Fig. 17.5 NACA0012 -Results on mesh 5



(a) Convergence of the CFD computation for mesh 3 and mesh 5.

(b) Numerical error with respect to mesh refinement.

Fig. 17.6 NACA0012

It should be kept in mind that a relative error of 1% means a difference of 5.3 10^{-2} *pc* (pc means power count where power count is, by analogy with drag count, equal to a dimensionless exergy coefficient of 0.0001) for the mechanical exergy term and 1.6 10^{-2} *pc* for the thermal exergy term whereas it implies a difference of 1.1 *pc* for the total anergy term or for the term $W\dot{\Gamma}$. Plots have to be interpreted cautiously.

The other interesting part of this study is the grid convergence analysis. Are all terms equally affected by mesh refinement or are there terms that can be estimated

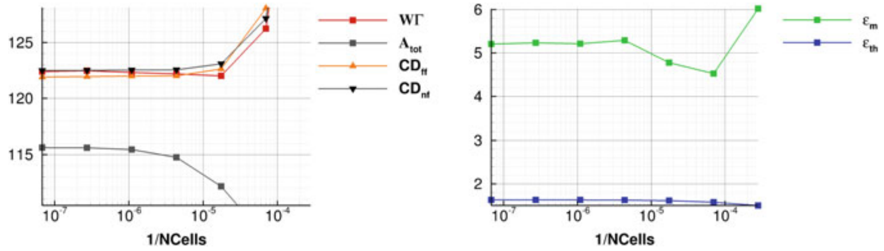


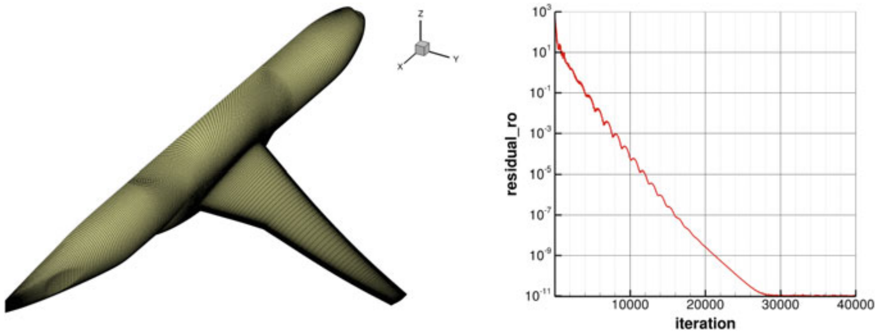
Fig. 17.7 NACA0012—Values of the FFX's terms with respect to mesh refinement

with good confidence even on a coarse mesh? Figure 17.7 shows values of the FFX decomposition with respect to mesh refinement. Plots have been split to give more clarity in the two different scales: $W\dot{\Gamma}$ and $\dot{A}_{tot}$ on one hand, $\dot{\mathcal{E}}_m$ and $\dot{\mathcal{E}}_{th}$ on the other hand. While the two first meshes are quite too coarse to provide reliable values, specifically for the $W\dot{\Gamma}$ term, mesh 3 and finer meshes give reasonable accuracy. They are within 1 counts of the finest mesh value for the $W\dot{\Gamma}$ value. Although a theoretically exact exergy decomposition is not available for this case, mesh 7 is considered fine enough (with 14,7M of cells) to provide reference values. Figure 17.6 gives the relative differences between the current values and these reference values (in percentage) for each mesh. The reduction of relative error seems to behave almost linearly with the number of cells.

The analysis of these cases shows that there is no great difference between far field drag terms behaviour and exergetic terms behaviour when dealing with CFD convergences. As ONERA's far field drag approach has been studied for more than fifteen years and has been proved to be quite effective, these first results then give confidence in the exergetic approach.

17.4.2.2 NASA-CRM Case

As it has been widely experimentally and numerically studied, the wing-body configuration of the Common Research Model (CRM) is a good case to assess the accuracy and sensibility of the FFX post-processing tool. Exergy decomposition has already been studied on this configuration during the early development of the formulation in Arntz [19]. At that time, the post-processing FFX tool was a prototype implemented in a FORTRAN code. Nowadays, it is a Python/C++ tool which is more mature and does not introduce any correction to account for spurious exergy. What is called spurious exergy is exergy having no physical meaning and only generated by numerical errors. Instead, it has been decided to investigate more carefully this term in order to get a better understanding. Figure 17.8a gives an overview of this configuration. The freestream aerodynamic conditions are Mach number of $M = 0.85$, $CL = 0.5$ and Reynolds number $Re = 5\ 10^6$. Note that it is a transonic case which introduces a shock wave phenomenon in addition to 3D effects compared to the previous subsonic



(a) Overview of the CRM-NASA configuration

(b) Residual on the mass conservation equation with respect to iteration numbers

Fig. 17.8 CRM case

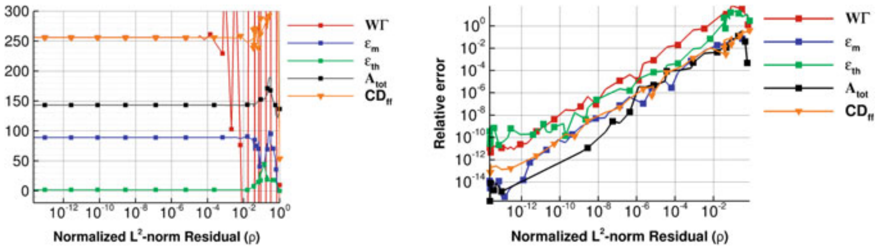
NACA0012 case. Several meshes are available for this configuration at ONERA and CFD computations and drag analyses have been documented in Hue [20]. These meshes, carefully generated in the context of the Drag Prediction Workshop, have good qualities in term of grid spacing, stretc.hing ratio and grid orthogonality. Note that they have been built in order to meet the drag prediction requirements. It is far from sure that they are also suitable for exergy evaluation. This is a question that our study aims at addressing: what is a good quality mesh from an exergy analysis point of view? Finally they belong to a family of six grids built by coarsening the finest one, which is a good point for mesh convergence analysis. In order to guarantee a CFD convergence down to machine precision, computations has been carried on for a great number of iterations. Figure 17.8b shows the residual of the mass conversation equation with respect to the number of iterations and confirms that machine precision is achieved.

Table 17.2 gives the value of the FFX's components for four meshes composed of 2M, 5M, 17M and 41M of cells. The thermal exergy part is quite well predicted for every mesh (relative error is less than 5.3%) but it should be noted that this term takes small values. Good accuracy is also obtained for the mechanical exergy part with less than 2.6% relative error even for the coarsest mesh. The amount of total anergy is less accurately predicted on the coarsest mesh (7.1% of relative error) but this error decreases as the mesh get finer and is less than 2% on L4. Finally the term $W\dot{\Gamma}$ still seems to be the most difficult to estimate accurately as the error is higher on L3 than on L2 and reaches 10% . It is consistent with observations on the NACA0012 case.

As for the NACA0012 test case, the accuracy of the FFX decomposition with respect to CFD convergence has been studied on meshes L2 to L4. The Fig. 17.9 only shows the result for mesh L3 but the terms have a very similar behaviour on all three meshes. The total anergy term provides the fastest convergence and is even faster than the far-field drag term. The mechanical exergy, which is far from being

Table 17.2 Contributions to the exergetic balance for different meshes

CRM	L’2	L’3	L’4	L’5
Ncells	2 156 544	5 111 808	17 252 352	40 894 464
$W\dot{\Gamma}$	257.97	256.18	254.63	254.35
$\dot{\mathcal{E}}_m$	87.86	89.14	90.77	91.85
$\dot{\mathcal{E}}_{th}$	1.84	1.81	1.86	1.87
$\dot{A}_{tot}$	139.05	143.10	147.04	149.40



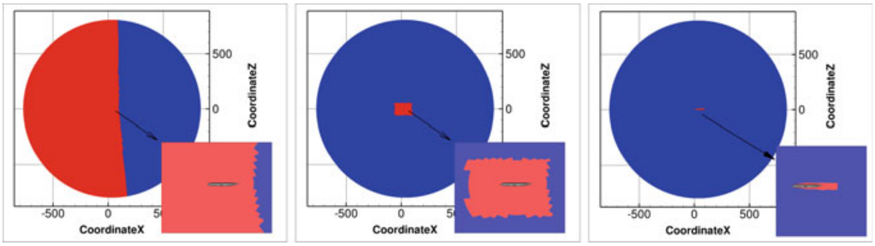
(a) Values of the FFX decomposition with respect to CFD convergence

(b) Relative error in FFX’s terms with respect to CFD convergence

Fig. 17.9 CRM—Results on mesh LP3

a negligible term for this case, shows a very good convergence too. As previously explained, the thermal exergy part takes too small values to be able really to appraise its convergence performance. Finally the $W\dot{\Gamma}$ is more demanding in terms of CFD convergence in order to achieve the same precision as the other terms.

Another aspect that could help improving the post-processing accuracy is the definition of the control volume, where the exergy balance is calculated. Destarac [21] and al. have carefully designed control volumes for the far-field drag method. The objective is to exclude any volume where no physical phenomenon occurs and only spurious drag can be generated. The Fig. 17.10 gives illustration of the three definitions of control volume considered in this study. It should be reminded that a single FFX balance is achieved for a fixed downstream Trefftz plane (defined by the x_{TP} parameter). All other boundaries of the control volume can be freely defined as long as they are far enough for the flow to get back to thermodynamic equilibrium. The first volume ($V1$) extends up to the farfield boundary condition in all directions (except for the x_{TP} plane as explained). The Table 17.3 gives FFX terms for the three volumes. The second volume ($V2$) is defined by limiting the control volume in x, y and z direction as $x_{min} = z_{min} = -50$ and $y_{max} = z_{max} = 50$. There are very few differences in the results between $V1$ and $V2$ and even negligible differences (values are absolutely the same up to 2 decimal digits) for the anergy terms. It means that, for this case, there is not a significant amount of spurious anergy generated outside this box. Finally the volume $V3$ is a volume similar to one of the volumes used for far-field drag analysis. It is built as a viscous volume defined by physical sensors,



(a) Classical control volume without restriction (V1)

(b) Restriction in all 3 directions (V2)

(c) FFD-like control volume (V3)

Fig. 17.10 Control volumes for the CRM L2 case. The control volume is marked by the red contour region

Table 17.3 FFX balance at $x_{TP} = 90$ for 3 control volumes: V1 is the classical FFX control volume, V2 is the control volume with restriction in all directions and V3 is the FFD-like control volume

Control volume	$W\dot{\Gamma}$	$\dot{\mathcal{E}}_m$	$\dot{\mathcal{E}}_{th}$	$\dot{A}_{\Phi}$	$\dot{A}_{\nabla T}$
V1	257.97	87.86	1.84	122.72	11.96
V2	258.78	88.25	1.60	122.72	11.96
V3	257.82	466.73	−375.30	122.70	11.96

plus a shock volume also identified thanks to physical sensors. Whereas viscous and thermal anergy ($\dot{A}_{\Phi}$ and $\dot{A}_{\nabla T}$) still have quite the same values for $V3$, mechanical and thermal exergy are badly predicted. This is a direct result of thermodynamic equilibrium not being is not yet achieved on the boundaries on this control volume.

To sum up, the convergence of the term $W\dot{\Gamma}$ is the trickiest one as it has been also confirmed for other test cases (not presented here). Moreover, the evaluation of the anergy terms suffers with greater error as it deals with gradients computation and integration in a larger domain.

17.5 Conclusion

A new promising post-processing method has been presented based on the exergy concept. Such an analysis can be used as a basis for the construction of objective functions for optimization processes. Although it has already been widely studied from a physical point of view by Arntz [6], there was still a need for an improved numerical understanding. Indeed, as long as numerical methods are concerned, discretization of the continuous flow field, truncation error, convergence of iterative processes imply that the result can not be free from error. So the aim of this paper was to evaluate the loss of accuracy associated with mesh discretization and imperfect CFD converged solutions and to find strategies to minimize it. Of course each flow simulation and analysis will have its own distinctive characteristics and no conclusion can be drawn which would be universal. However it has been found that the term $W\dot{\Gamma}$ was the most sensitive and it will be of great interest to look further into this issue. Comparisons with the farfield approach prove that, in term of mesh convergence and sensitivity to CFD convergence, the exergy decomposition's terms show a quite equivalent behaviour. Finally some attempts at increasing the post-processing accuracy have been achieved through a reduction of the control volume. Although no huge differences are obtained for the CRM case presented in this paper, it seems wise to exclude from the control volume region where the mesh is too coarse to accurately predict the flow. Indeed these are locations where spurious anergy will be produced. So the control volume has to be large enough for the flow to return to thermodynamic equilibrium but small enough to exclude zones where the mesh become too coarse. Some additional work will be carried on to automatically define a control volume that meets this requirement.

References

1. Van der Vooren J, Destarac D (2004) Drag/thrust analysis of a jet-propelled transonic transport aircraft: definition of physical drag components. Aerosp Sci Technol 8:545–556
2. Toubin H, Bailly D (2015) Development and application of a new unsteady far-field drag decomposition method. AIAA J 53(11):3414–3429
3. Gariépy M, Trépanier J-Y, Malouin B (2013) Generalization of the Far-Field Drag Decomposition Method to Unsteady Flows. AIAA Journal 51(6):1309–1319
4. Mele B, Tognaccini R (2014) Aerodynamic force by Lamb vector integrals in compressible flow. Phys Fluids 26:1–16
5. Wu J, Liu L, Liu T (2018) Fundamental theories of aerodynamic force in viscous and compressible complex flows. Prog Aerosp Sci 99:27–63
6. Arntz A (2014) Civil aircraft aero-thermo-propulsive performance assessment by an exergy analysis of high-fidelity CFD-RANS Flow Solutions. Ph.D. Dissertation, Lille 1 Université - Sciences et Technologies, Lille, France, (2014)
7. Arntz A, Atinault, O, Destarac D, Merlen A (2014) Exergy-based aircraft aeropropulsive performance assessment: CFD application to boundary layer ingestion. 32nd AIAA applied aerodynamics conference, AIAA AVIATION forum, AIAA paper 2573 (2014)
8. Arntz A, Atinault O, Merlen A (2015) Exergy-based formulation for aircraft aeropropulsive performance assessment: theoretical development. AIAA J 53(6) (2015)
9. Petropoulos I, Wervaecke C, D, B,T, D (2019) Numerical investigations of the exergy balance method for aerodynamic performance evaluation. AIAA AVIATION forum (17–21 June 2019, Dallas, Texas)
10. Drela M (2009) Power balance in aerodynamic flows. AIAA J 47(7):1761–1771
11. ACARE - FlightPath 2050 Goals. https://www.acare4europe.org/sria/flightpath-2050-goals
12. Benoit C, Péron S, Landier S (2015) Cassiopee: a CFD pre- and post-processing tool. Aerosp Sci Technol 45:272–283
13. Tailliez C, Arntz A (2018) CFD assessment of the use of exergy analysis for losses identification in turmbomachines flows. In: 53rd 3AF international conference on applied aerodynamics, Salon de Provence, France (March 2018)
14. Couilleaux A, Arntz A (2018) Exergy analysis for a CFD-based turbofan exhaust mixer performance improvement. In: 53rd 3AF international conference on applied aerodynamics, Salon de Provence, France (March 2018)

15. Wiart L, Negulescu C (2018) Exploration of the AirbusâĂNAUTILIUSâĂ İ engine integration concept. In: 31st congress of the international council of the aeronautical sciences, Belo Horizonte, Brazil (September 2018)
16. Cambier L, Heib S, Plot S (2013) The Onera elsA CFD software: input from research and feedback from industry. Mech Ind 14(3):159–174
17. Drag Prediction Workshop website. https://aiaa-dpw.larc.nasa.gov/
18. NASA Turbulence Modeling Resource website. https://aiaa-dpw.larc.nasa.gov/
19. Arntz A, Hue D (2016) Exergy-based performance assessment of the NASA common research model. AIAA J 54(1):88–100
20. Hue D, Chanzy Q, Landier S (2018) DPW-6: drag analyses and increments using different geometries of the common research model airliner. J Aircraft 55(4):1509–1521
21. Destarac, D. Far-Field / Near-Field Drag Balance and Applications of Drag Extraction in CFD, In: CFD-based Aircraft Drag Prediction and Reduction. *VKI Lecture Series 2003, Von Karman Institute for Fluid Dynamics, Rhode Saint GenÃ̃lse* (February 3-7, 2003)

Chapter 18
Surrogate-Based Shape Optimization of Centrifugal Pumps for Automotive Engine Cooling Systems

R. De Donno, A. Fracassi, G. Noventa, A. Ghidoni, and S. Rebay

Abstract This paper investigates the capability of a surrogate-based optimization technique for the fast and robust design of centrifugal pumps. The centrifugal pump considered in this work is designed for automotive cooling system and consists of an impeller and a volute. A fully three-dimensional geometry parametrization based on Bézier surfaces for the impeller and the volute is presented. The optimization strategy is based only on open-source software (with the exception of the mesh generation process), i.e. Scilab for the geometric parametrization, OpenFOAM for the CFD simulations and DAKOTA for the optimization. To assess the potential and robustness of the proposed methodology, the initial geometry was chosen very far from the optimum design, having an impeller with straight blades. The operating conditions have been provided by the Italian company *Industrie Saleri Italo S.p.A.* and are typical of a Diesel engine.

Keywords Centrifugal pump · Automotive · Surrogate-based optimization · Computational fluid dynamics (CFD)

R. De Donno (✉)
Industrie Saleri Italo S.p.A., via Ruca 406, 25065 Lumezzane, BS, Italy
e-mail: remo.dedonno@saleri.it

A. Fracassi · G. Noventa · A. Ghidoni · S. Rebay
Department of Mechanical and Industrial Engineering (DIMI), University of Brescia,
via Branze 38, 25123 Brescia, Italy
e-mail: a.fracassi004@unibs.it

G. Noventa
e-mail: giarmaria.noventa@unibs.it

A. Ghidoni
e-mail: antonio.ghidoni@unibs.it

S. Rebay
e-mail: stefano.rebay@unibs.it

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_18

18.1 Introduction

Centrifugal pumps are widely used for many applications and, therefore, must be suited for a wide range of pressure ratios and flow rates. Their design and performance prediction is not trivial, being influenced by many free geometric parameters. Experimental (modification of prototypes and previous models) and numerical techniques have been applied to their design and analysis. However, the former approach is expensive and time-consuming, while the latter makes available numerical results which are not easily related to the pump performance.

In the last decade, to overcome the problems shown by the previous techniques, many researchers proposed the coupling of CFD (Computational Fluid Dynamics) codes and optimization algorithms for the fast and robust design of turbomachinery [1–5]. The shape optimization techniques have been also successfully applied to the pump design [6–8], even if considering some simplifications, such as a 2D geometry, or the parametrization of a single component (impeller or vaned diffuser or volute).

The novelty of this work is to propose an approach for the optimal design from scratch of a 3D centrifugal pump for the automotive cooling system, driven by a surrogate-based optimization technique, where the working point (WP) has to correspond to the best efficiency point (BEP). A detailed three-dimensional geometry parametrization based on Bézier surfaces for the impeller and the volute has been defined and presented, which allows to control the complete pump geometry. The single objective genetic algorithm SOGA, available in the software Dakota [9], is applied to a surrogate model, built with a Kriging method, in order to find the global optimum of the objective function, *i.e.* the hydraulic efficiency η. The efficiency and pressure ratio of the pump are evaluated through incompressible steady-state RANS simulations, exploiting the CFD solver available in OpenFOAM [10].

The optimization method has been assessed considering an initial geometry very far from the optimum design, having straight impeller blades and the shape of the volute defined by classical empirical correlations [11]. The operating conditions, defined by the flow rate, impeller rotational speed, and pump pressure ratio, have been kept constant.

A brief description of a centrifugal pump is given in Sect. 18.2, while Sects. 18.3, 18.4, and 18.5 give a detailed description of the optimization procedure, including geometry parametrization, flow computation and optimization algorithms. Section 18.6 shows the optimization results and presents the optimized geometries. Finally, in Sect. 18.7 the conclusions of the work are discussed.

18.2 Centrifugal Pump

Centrifugal pumps are turbomachinery used worldwide for many different applications. Figure 18.1 shows an example of centrifugal pump for the automotive field, designed and manufactured by the Italian company Industrie Saleri Italo S.p.A. [12].

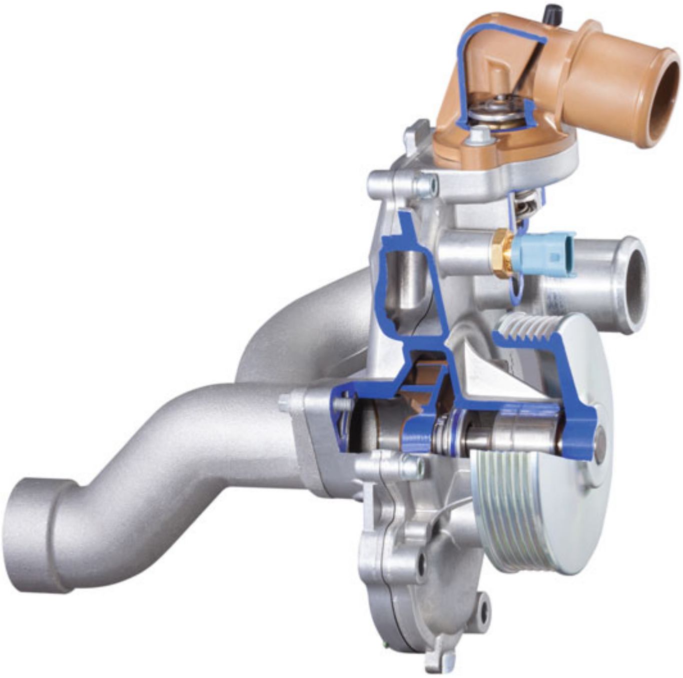


Fig. 18.1 Centrifugal pump assembly for the automotive field produced by Industrie Saleri Italo S.p.A

The main task of this family of centrifugal pumps is to pump the coolant through the cooling circuit to control the engine temperature. They have often severe geometric constraints, due to the ever-smaller engine packages. The pump manufacturers usually can optimize only the impeller and the volute, since the suction pipes are usually prescribed by the engine. Furthermore, the vaneless diffuser is usually not present, because of the small gap in the radial direction, while the vaned diffuser is not considered because it could decrease the pump performance in off-design conditions.

The main components of the pump are highlighted in the picture: the pulley of the pump receives the rotation from the engine by means of a belt connection. The motion is transferred to the impeller through the bearing; wet and dry parts of the machine are separated by means of a mechanical seal. The liquid to be pumped flows through the suction pipe to the impeller that transfers the energy necessary to transport the fluid and accelerates it in the circumferential direction. The fluid exiting the impeller is decelerated in the volute, increasing its static pressure.

The parameters used to describe the pump performance are the hydraulic efficiency

$$\eta = Q \Delta p_t / W, \tag{18.1}$$

and the total pressure rise coefficient

$$\psi = 2(p_{t4} - p_{t0})/\rho U_2^2, \tag{18.2}$$

where Q $[m^3/s]$ is the volumetric flow rate, Δp_t $[Pa]$ the total pressure rise across the pump, W $[W]$ the power at the impeller, p_{t4} the total pressure at the volute outlet, ρ $[kg/m^3]$ the density, U_2 $[m/s]$ the peripheral velocity at the impeller outlet, and p_{t0} the total pressure at the suction pipe.

18.3 Geometry Parametrization

Automotive centrifugal pumps (see Sect. 18.2) are made up of two main components, *i.e.* the impeller and the volute, which are parameterized with the open-source software Scilab [13]. A total amount of 25 design variables perturb the pump geometry during the optimization process, 13 for the impeller and 12 for the volute.

18.3.1 Impeller

The impeller geometry considered in this parametrization has an axial inflow and a radial outflow. During the optimization process, 13 design variables (DVs) perturb the impeller geometry:

- 3 DVs control the general dimensions of the machine: blades number, impeller diameter and blade height at the trailing edge
- 4 DVs control the meridional channel. Hub and shroud contours are fully defined by Bézier curves of sixth order, the design variables perturb the r and z coordinates of control points h4 and s4 shown in Fig. 18.2
- 6 DVs control the blade camber surface. The camber surface is fully defined by a Bézier surface of fourth order in radial direction and third order in axial direction. According to Fig. 18.2 and considering the nomenclature proposed by Van den Braembussche [1], the six design variables perturb the θ-coordinate of the following control points:
  - one design variable perturbs b11, one perturbs b31 and one perturbs b51 in order to control the camber line at the span 0%
  - one design variable perturbs b14 in order to control the relative position of the blade camber line at span 100% with respect to span 0%
  - one design variable perturbs the control points b12, b22, b32, b42, b52 in order to control the camber line twisting at span 25%
  - one design variable perturbs the control points b13, b23, b33, b43, b53 in order to control the camber line twisting at span 75%

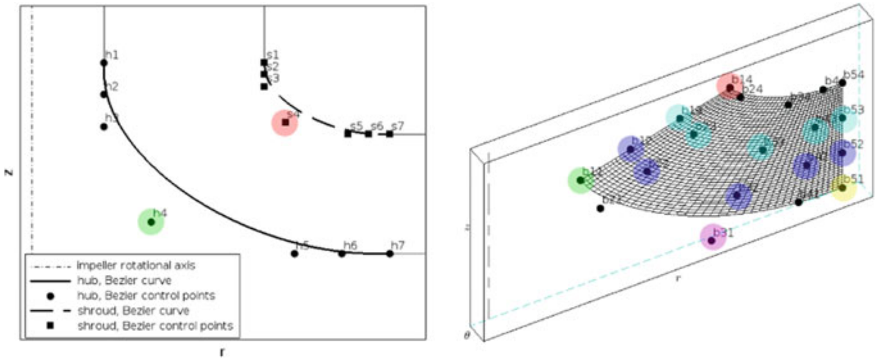


Fig. 18.2 Meridional channel definition (left) and blade camber surface definition (right) with marks on the control points perturbed by the design variables

The Bézier curves order has been set in order to represent properly a large number of impeller geometries produced by Industrie Saleri. Once the blade camber surface is defined, a thickness function is applied to determine the blade suction and pressure side. Any symmetrical 4-digit NACA (National Advisory Committee for Aeronautics) profile can be used, here the NACA0012 is adopted with a modification to the last coefficient (i.e. to -0.05) to ensure a feasible thickness at the blade trailing edge.

The optimization algorithm necessitates of a baseline geometry for the impeller: the meridional channel is only defined by the first three DVs controlling the general dimensions of the machine, while the blades are straight.

18.3.2 Volute

The geometrical parametrization of the volute is based on the approach proposed by Heinrich and Schwarze [15], 12 design variables perturb the volute geometry during the optimization process.

The shape of the volute cross-sectional area at the outflow is first defined, using a Bézier curve of seventh order to take into account large geometric modifications as shown in Fig. 18.3. Then, the areas decrease linearly until the volute tongue, as shown in Fig. 18.4. The design variables control the cross sectional area at the outlet, by perturbing the x and z coordinates of control points v2, v3, v4, v5, v6 and v7 shown in Fig. 18.3. Control points v1 and v8 are not perturbed in order to connect properly the volute with the impeller.

The optimization algorithm necessitates of a baseline geometry for the volute, which is built considering the following parameters: the flow rate and pressure head at the working point, the impeller blade height at the trailing edge and the coordinates

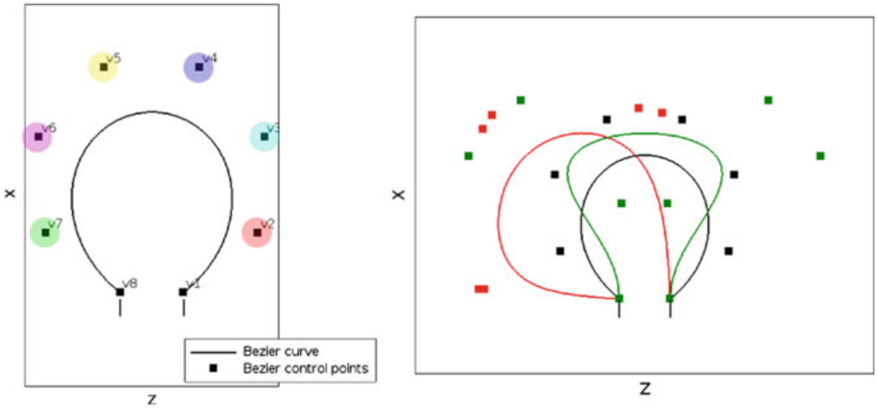
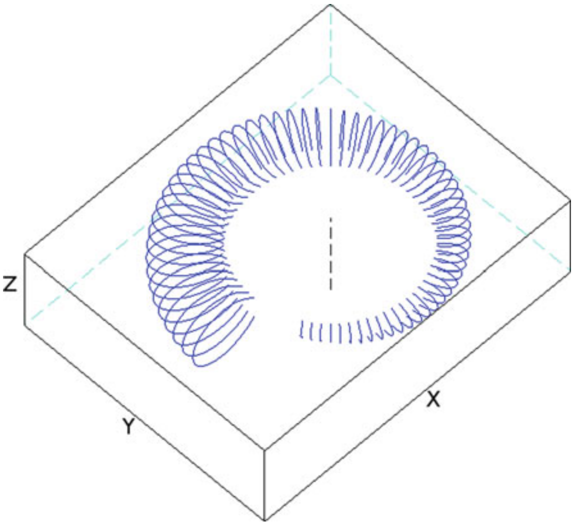


Fig. 18.3 Volute cross sectional areas definition with marks on the control points perturbed by the design variables (left) and different shapes of the cross-sectional area at the volute outflow (right) where the baseline is represented in black and two random geometries in red and green

Fig. 18.4 3D cross-sectional areas evolution of the volute



of the interface with the impeller. The control points lie on the circle, whose radius is calculated by means of the Stepanoff's law [11]:

$$r^* = F^* \sqrt{\frac{A_{360^\circ}}{\pi}},$$

where $A_{360^\circ} = \frac{Q}{c_3}$ is the area of the cross section at the end of the volute development, F^* a corrective factor equal to 1.3 (the area of the resulting geometry is smaller than the area calculated with the Stepanoff's law), Q the pump flow rate, $c_3 = K3\,\sqrt{2gH}$

the averaged volute velocity, g the gravitational acceleration, H the pump head, and $K3$ an experimental design factor varying between 0.15 and 0.5, function of the pump specific speed. Figure 18.3 (right) shows different shapes of the cross-sectional area at the volute outflow that can be defined by the geometric parametrization.

18.4 CFD Simulation

The flow-field has been computed using the incompressible steady state solver MRF-SimpleFoam available in OpenFOAM [10], which solves the RANS (Reynolds Average Navier Stokes) equations coupled with the SST (Shear Stress Tensor) turbulence model [16] .

The computational domain inlet has been obtained extruding along the axial direction the impeller inlet section three times the length of its diameter to avoid possible disturbance at the inflow due to the blade leading edge. The mesh of the impeller is generated using the software TurboGrid 18.2 [17], while the mesh of the volute is generated using the software cfMesh 1.1.1 [18]. The size of the elements adjacent to the solid walls is equal to a non-dimensional distance $y^+ \approx 1$, to compute the boundary layer accurately up to the wall.

At the domain inlet the volumetric flow rate Q, the turbulence intensity Tu_1, and specific dissipation rate ω_1 are prescribed, while at the outflow a static pressure p_4 is set. The no-slip condition is applied to the blade walls, hub and shroud. At the wall k_w is set to zero, while ω_w is computed by exploiting its asymptotic behavior.

Steady-state simulations are performed using the multiple reference frame (MRF) approach, which implies no relative mesh motion between the rotating and stationary parts. In the rotating reference frame, where the relative velocity is computed, the momentum equation is modified, adding Coriolis and centrifugal terms. The interface between moving and fixed domain is treated using the mixing plane approach.

The second order upwind discretization scheme is applied to the divergence of the velocity, while the first order upwind scheme is applied to the turbulent quantities. The Laplacian terms are evaluated using a linear second order bounded central scheme, while a central differencing method approximates the gradient term.

18.5 Optimization Strategy

The maximization of the centrifugal pumps efficiency is a highly non-linear problem, whereby a highly non-linear approximation model and a global optimization algorithm are required. In the literature, the single (SOGA [19]) or multi-objective genetic algorithms (MOGA [5]) are used for turbomachinery shape optimizations, due to their simplicity and robustness: objective functions derivatives are not requested and the probability to remain trapped in a local optimum is very low. To overcome the computational effort requested by genetic algorithms due to the large number of evaluations, the use of a surrogate model to approximate and evaluate the objective functions during the optimization process is mandatory. Studies show that Kriging (KRG) [20, 21] and artificial neural network (ANN) fit well the performance trend of the pump.

A preliminary study is performed to assess which surrogate better conforms to this problem. For this investigation a shape optimization of the impeller blade of the well known Ercoftac centrifugal pump [22] in a 2D configuration is taken into account. The Ercoftac pump includes an impeller and a vaned diffuser, but for this purpose only the impeller is considered for CFD simulations, to avoid the influence of the interaction rotor-stator on the shape optimization. The blade is shaped through three input variables, corresponding to three characteristic angles: i) the inlet angle ii) the outlet angle and iii) the stagger angle. The thickness function is fixed equal to the original one. The efficiency of the impeller is chosen as the objective function of the problem. A surrogate based optimization is performed starting from different number of training points and applying the KRG or the ANN as surrogate. The CFD evaluations are computed according to the setup shown in Sect. 18.4, without interface treatment since only the rotating domain is present. Calling n the number of design variables, at least $N = \frac{(n+1)(n+2)}{2}$ designs are calculated for the Design of Experiments (DoE), as suggested in the literature [9]. The surrogates are then applied to the training points defined by means of the DoE and the following results are reached:

- Using the KRG, the minimum number of training points that allows to have a sufficient accuracy of the surrogate is 1.5 times N.
- Using the ANN, with the same number of initial evaluation the surrogate is not sufficiently accurate to continue the optimization process.

Considering these results and aiming to exploit as low computational resources as possible, the KRG model is used in the following.

In the centrifugal pump performance, the hydraulic efficiency η and the total pressure rise Δp_t have a fundamental role and therefore are chosen as optimization objective and constraint, respectively. In particular the optimization algorithm maximizes η, while keeping Δp_t constrained to the operative point analyzed (Table 18.1) with a tolerance of $\pm 5\%$ on the pressure head.

A surrogate-based single objective genetic algorithm with a non-linear constraint is therefore applied. The whole optimization strategy is managed by the Dakota [9] software and is defined by the following steps.

Table 18.1 Pump working point

Rotational speed	n	8700	rpm
Flow rate	Q	305	lpm
Pressure head	Δp_t	3.3	bar

1. Computation of a Design of Experiments (DoE) to create a training points database. The DoE is generating using the Latin Hypercube Sampling (LHS) method, which allows to randomly and uniformly distribute the designs over the whole design space. A database of 500 training points is created for this work.
2. Training points evaluation. The link between geometry and perfomance is evaluated by means of CFD simulations, whose setup is described in Sect. 18.4.
3. Surrogate models construction (Kriging) for the approximation of pump efficiency and head.
4. Search for the maximum of the efficiency by means of the constrained SOGA applied to the surrogate model. Crossover rate and mutation rate are set equal to 0.8 and 0.1, respectively.
5. Verification of the maximum through CFD simulation.
6. If the convergence criterion is not met, add the maximum to the training points database and return to step 3. The convergence criterion used in this work stops the optimization procedure when the percentage error between the CFD results of three consecutive iterations is less than 0.5% for both η and Δp_t.

The iterative process described above improves continuously the surrogate accuracy and accelerates the optimization convergence.

18.6 Results and Discussion

This section contains the results of the surrogate-based shape optimization method described in the previous sections. The starting geometry is very far from the optimum design, having straight impeller blades and the shape of the volute defined by classical empirical correlations [11]. The working point chosen for testing the procedure is a typical operative point for six cylinder diesel engines, and is reported in Table 18.1. The geometric parametrization described in Sect. 18.3 has been adopted to model the impeller and the volute.

The meshes, as reported in Sect. 18.4, are generated with TurboGrid (impeller) and cfMesh (volute). The number of elements during the optimization is around 670,000 and 900,000 for the impeller and volute, respectively. The value of the boundary conditions are reported in Table 18.2, where ν denotes the kinematic viscosity, μ_t/μ the ratio between turbulent and molecular eddy viscosity equal to 10, β_1 a constant value equal to 0.075, and y the distance between the wall and the center of the cell adjacent to the wall. The kinetic energy at the inlet corresponds to a turbulence intensity $Tu_1 = 0.5\%$.

The relation between the objective function η and the nonlinear inequality constrained Δp_t is shown in Fig. 18.5, where the output of the computer experiments and the evolution of the genetic algorithm assisted by the surrogate model are represented. After the 500 training points evaluation, the optimization has run further 38 CFD simulations to reach the convergence and find the optimal solution. Then, Fig. 18.5 highlights the improvement from the initial to the optimal design in terms of pump efficiency and the convergence charts of the optimization procedure.

Table 18.2 Boundary conditions

Coolant		
T	100	°C
ρ	1020	kg/m^3
ν	7.8e−7	m^2/s
Inflow		
Q	0.00508	m^3/s
k_1	$\frac{3}{2}U_1^2 T u_1^2$=0.496	m^2/s^2
ω_0	$\frac{k_1}{\nu}\left(\frac{\nu_t}{\nu}\right)^{-1} = 4962$	1/s
Outflow		
p_3	0	m^2/s^2
Wall		
k_w	0	m^2/s^2
ω_w	$\frac{6\nu}{\beta_1 y^2}$	1/s

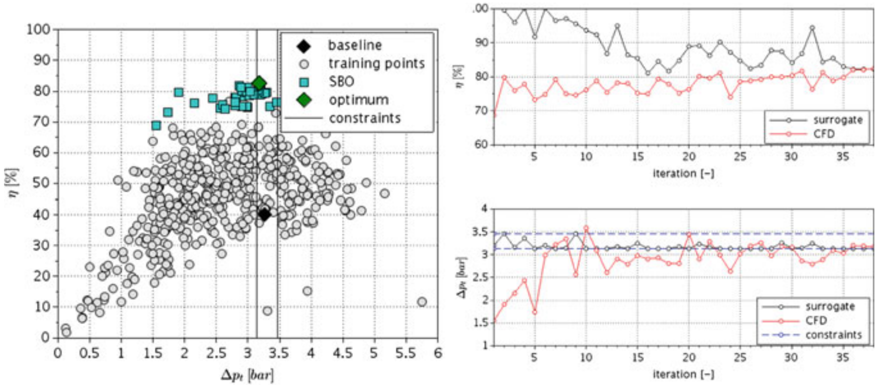


Fig. 18.5 CFD results of the surrogate based optimization (left) and optimization convergence charts (right)

Figure 18.6 (left) shows that the best design has the same number of blades with respect to the baseline (5) but with an important curvature, and that the impeller outer diameter has been enlarged. Then, Fig. 18.6 (center) shows that the blade height has been reduced and that the blade is twisted along the span direction for the best design. Furthermore, Fig. 18.6 (right) shows that the cross-sectional area at the volute outflow is bigger with a teardrop shape for the best design.

When comparing the pressure fields, Fig. 18.7 shows that the optimal design has less losses inside the volute. Furthermore, the sections of Fig. 18.8 show that the best design has a more uniform pressure distribution in the blade to blade passage as well as inside the volute.

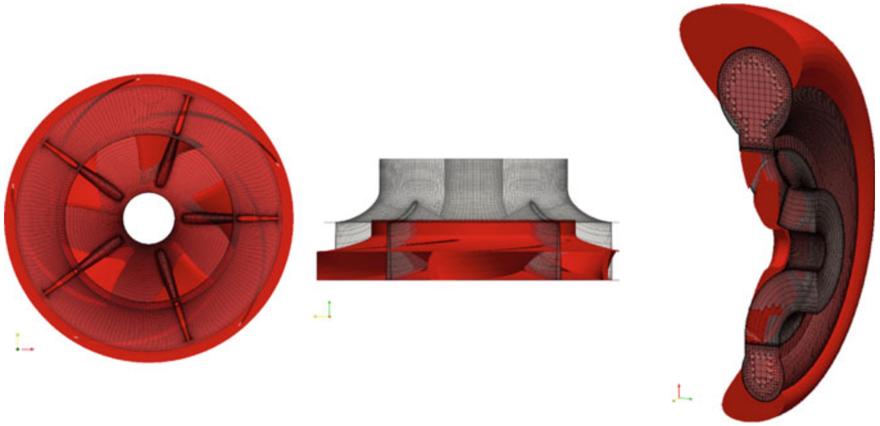


Fig. 18.6 Top view (left) and side view (center) of the impellers and section view of the pumps (right). Comparison between baseline configuration (black wireframe) and best design (red solid color)

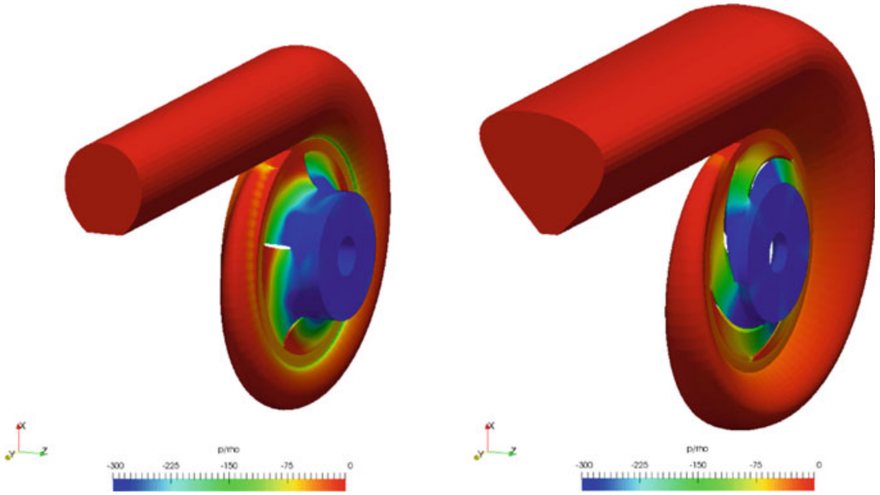


Fig. 18.7 Pressure field of the baseline (left) and best (right) design expressed in Pa/(kg/m^3)

Finally, when considering the velocity fields, Fig. 18.9 shows that the best design has an expected decrease of the flow velocity along the volute, for converting the kinetic energy in pressure, while the baseline design shows high velocity inside the volute, suggesting that for this configuration the volute cross-sectional areas are too small.

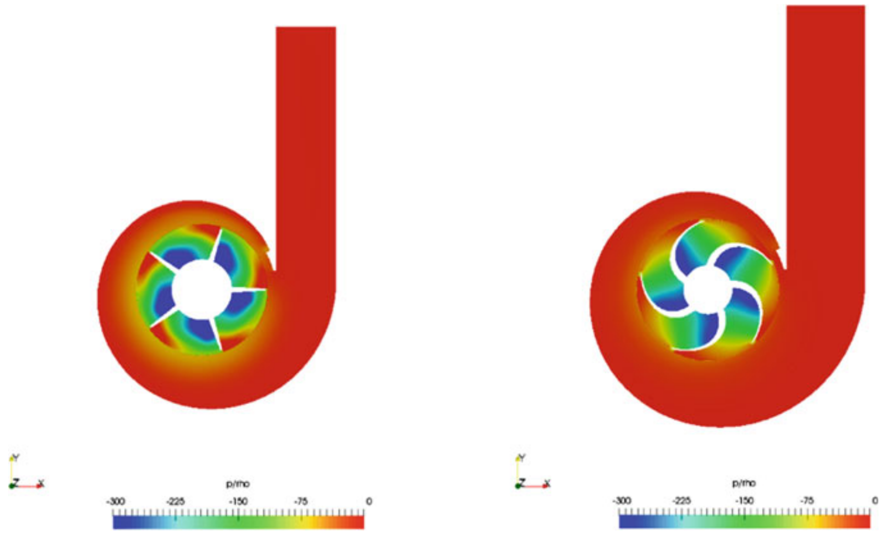


Fig. 18.8 Pressure field of the baseline (left) and best (right) design at span 50% of the impeller expressed in Pa/(kg/m^3)

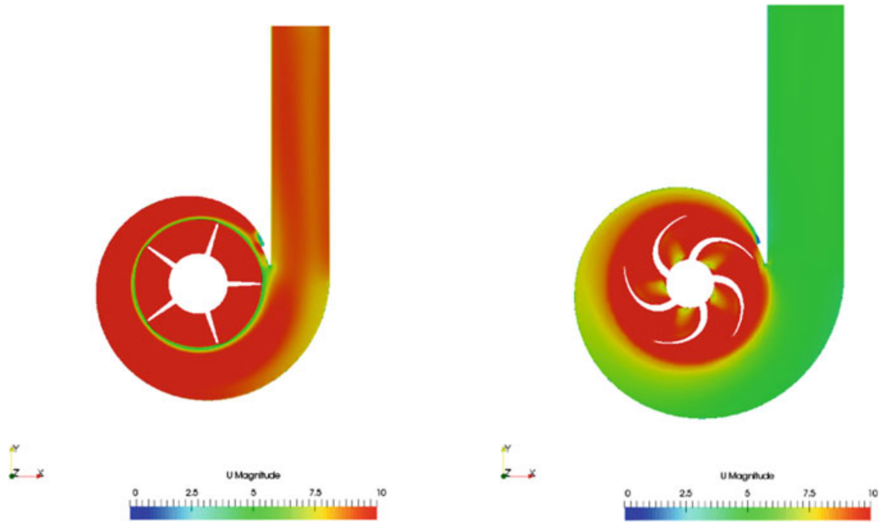


Fig. 18.9 Velocity field of the baseline (left) and best (right) design at span 50% of the impeller expressed in m/s

18.7 Conclusions

A fully automated surrogate-based optimization method has been presented for maximizing the efficiency of a centrifugal pump designed for the engine cooling system, as a tool for the design from scratch of this kind of turbomachine. The robustness of the method has been assessed optimizing an initial geometry consisting of an impeller with straight blades and a volute computed according to empirical correlations available in the literature. The working point of typical six cylinders Diesel engines has been considered. The geometry has been parametrized with Bézier polynomials and 25 design variables have been used for the optimization. The Kriging surrogate model has been adopted for this work and built on an initial population of 500 training points, while a single objective genetic algorithm has been set in order to maximize the pump efficiency coefficient η, while keeping constrained the pressure rise Δp_t. The surrogate-based optimization has reached the convergence in 38 iterations, improving the pump efficiency from 39.93% (baseline) to 82.46% (optimal design) with a total amount of about 22,000 cpu hours. The results of this work make the procedure here described a valuable tool for the design of centrifugal pumps. Future work will be dedicated to include the pressure losses of the real turbomachine in the optimization procedure, considering the gap between the impeller and volute, the balance holes of the impeller and the engine package constraints of the suction pipe and the volute. Furthermore, ongoing work is devoted to link the present optimization strategy to an in-house 0D code, which will provide a preliminary “optimized” design of the complete pump to speed up the convergence of the method.

Acknowledgments Research carried out with the support of resources of Industrie Saleri Italo S.p.A.

References

1. Van Den Braembussche RA (2006) Optimization of radial impeller geometry. In: Design and analysis of high speed pumps, number RTO-EN-AVT-143. RTO of NATO
2. Pasquale D, Ghidoni A, Rebay S (2013) Shape optimization of an organic rankine cycle radial turbine nozzle. J Eng Gas Turbines Power 135(4):042308-042308-13
3. Guo Z, Song L, Zhou Z, Li J, Feng Z (2015) Multi-objective aerodynamic optimization design and data mining of a high pressure ratio centrifugal impeller. J Eng Gas Turbines Power 137(9):092602-092602-14 09
4. Verstraete T, Alsalihi Z, Van den Braembussche RA (2010) Multidisciplinary optimization of a radial compressor for microgas turbine applications. J Turbomach 132(3):031004-031004-7 03
5. Olivero M, Pasquale D, Ghidoni A, Rebay S (2014) Three-dimensional turbulent optimization of vaned diffusers for centrifugal compressors based on metamodel-assisted genetic algorithms. Optim Eng 15(4):973–992
6. Pei J, Wang W, Yuan S (2016) Multi-point optimization on meridional shape of a centrifugal pump impeller for performance improvement. J Mech Sci Technol 30(11):4949–4960

7. Wang W, Pei J, Yuan S, Zhang J, Yuan J, Xu C (2016) Application of different surrogate models on the optimization of centrifugal pump. J Mech Sci Technol 30(567–574):02
8. An Z, Zhounian L, Peng W, Linlin C, Dazhuan W (2016) Multi-objective optimization of a low specific speed centrifugal pump using an evolutionary algorithm. Engineering Optimization 48:1251–1274
9. Dakota, 6.8 edition. https://dakota.sandia.gov/
10. OpenFOAM extend, 3.2 edition. https://sourceforge.net/projects/foam-extend/
11. Stepanoff A (1993) Centrifugal and axial flow pumps. Krieger Publishing Company
12. Industrie Saleri Italo S.p.A. http://www.saleri.it/
13. Scilab, 5.5.2 edition. http://www.scilab.org/
14. Peck JF (1968) Design of centrifugal pumps with computer aid. Proc Inst Mech Eng 183:321–351
15. Heinrich M, Schwarze R (2016) Genetic algorithm optimization of the volute shape of a centrifugal compressor. Int Jo Rotat Machi
16. Menter F (1993) Zonal two equation k-w turbulence models for aerodynamic flows. Fluid dynamics and co-located conferences. American Institute of Aeronautics and Astronautics AIAA Jul, 10.2514/6.1993-2906
17. TurboGrid, 2019 R2 edition. https://www.ansys.com/products/fluids/ansys-turbogrid/
18. cfMesh, 1.1.1 edition. https://cfmesh.com/cfmesh/
19. De Donno R, Ghidoni A, Noventa G, Rebay S (2019) Shape optimization of the ercoftac centrifugal pump impeller using open-source software. Optim Engi. https://doi.org/10.1007/s11081-019-09428-3
20. Giunta A, Swiler L, Brown S, Eldred M, Richards M, Cyr E (2006) The surfpack software library for surrogate modeling of sparse irregularly spaced multidimensional data. In: 11th AIAA/issmo multidisciplinary analysis and optimization conference (AIAA Paper 2006–7049, Portsmouth, VA, 2006)
21. De Donno R, Rebay S, Ghidoni A (2019) Surrogate-based shape optimization of the ercoftac centrifugal pump impeller. Comput Methods Appl Sci 49:227–246
22. Petit O, Page M, Beaudoin M, Nilsson H (2009) The ERCOFTAC centrifugal pump OpenFOAM case-study

Chapter 19
Towards an Open-Source Framework for Aero-Structural Design and Optimization Within the SU2 Suite

Rocco Bombardieri, Ruben Sanchez, Rauno Cavallaro, and Nicolas R. Gauger

Abstract Ongoing efforts to develop a fully open-source framework for the aero-structural design and optimization of wings, including aerodynamic and structural geometric nonlinearities, are presented. The framework is self-contained and relies on the well-established SU2 suite for the computation of the aerodynamic part of the problem. SU2 is a python-wrapped C++ suite for multiphysics problems, able to compute accurate adjoint sensitivities by means of Algorithmic Differentiation techniques. For the structural problem, a C++ library featuring a nonlinear FE beam is employed. The library is fully wrapped in python and coupled to SU2 by means of a python orchestrator and a splining module for force and displacement transferring. The applicability of this approach is demonstrated using a known aeroelastic test case based on the ONERA M6 wing geometry. The structural solver is differentiated by means of Algorithmic Differentiation and structural and coupled adjoint-based sensitivities are evaluated and validated by comparison to Finite Differences for a variety of cases. The final goal of this research is to provide an integrated infrastructure for aeroelastic design and optimization of wings by means of coupled adjoint sensitivities, including challenging cases in which wings are operating in non-linear aerodynamic regimes, e.g., transonic flows, and subject to large displacements.

R. Bombardieri (✉) · R. Cavallaro
Department of Bioengineering and Aerospace Engineering,
Universidad Carlos III de Madrid (UC3M), Madrid, Spain
e-mail: rocco.bombardieri@uc3m.es

R. Cavallaro
e-mail: rauno.cavallaro@uc3m.es

R. Sanchez · N. R. Gauger
Chair for Scientific Computing, TU Kaiserslautern, Kaiserslautern, Germany
e-mail: ruben.sanchez@scicomp.uni-kl.de; ruben@su2foundation.org

N. R. Gauger
e-mail: nicolas.gauger@scicomp.uni-kl.de

R. Sanchez
SU2 Foundation, Sunnyvale, CA, USA

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_19

Keywords Aeroelasticity · Aero-structural optimization · SU2 · Computational fluid dynamics · Adjoint sensitivities · Open-source framework

19.1 Introduction

Current trends in aircraft design aim at reproducing ab-initio and at a tighter level the multidisciplinarity of the physical problem with high-fidelity prediction tools. Traditionally, the conceptual and preliminary development phases do not include flexibility effects (aeroelastic) which are typically estimated by empirical relations and available data from previous designs [1] rather than by incorporating physics-based analysis. However, when new concepts are to be designed, a transition from experience and engineering judgment-based methods to more quantitative and physics-based approaches is required to guarantee a reliable design process.

Coupled aeroelastic analysis has become, therefore, an area of active research with a number of established computational tools such as FUN3D [2] from NASA, TAU [3] from DLR or ElsA [4] from ONERA. In the context of Computational Fluid Dynamics (CFD) frameworks, the open-source suite SU2 [5] has attracted much attention for multidisciplinary analysis and design in recent years, particularly due to its adjoint capabilities for complex, non-linear problems [6–11]. SU2 is able to tackle Fluid-Structure Interaction (FSI) problems via a native, solid mechanics solver [12], and is also capable of computing coupled adjoint sensitivities of the FSI problem for multidisciplinary optimization [13, 14].

Aim of this research is to develop an open-source framework for design and optimization of wings including aerodynamic and structural nonlinearities along the lines proposed by Sanchez et al. [13]. A nonlinear beam finite element solver, namely pyBeam, is implemented to incorporate structural deformations while ensuring a good compromise between efficiency and reliability. This effort is a first stage towards a fully-functional adjoint-based infrastructure for performing gradient-based optimization of aircraft wing configurations, coupling SU2 with pyBeam.

The structure of the proposed framework, based on a python-wrapped interface between the different solvers (i.e. CFD solver, beam solver and interpolation solver), is an ideal solution for a wider infrastructure in which more tools of various fidelity levels, provided with a standard interface, can be incorporated to perform both analysis and optimization at the different stages of the design process. It will also facilitate an easy access to high-fidelity multidisciplinary optimization tools for aircraft design to a potentially large user audience.

First, an overview of the theoretical background of the method is summarized in Sect. 19.2. Later, preliminary applications of the tool are shown in Sect. 19.3, using an aeroelastic test-case similar to the one used in a previous work by Bombardieri et al. [15] which features an ONERA M6 wing surface augmented with a synthetic structure. Section 19.4 contains some basics of adjoint based optimization featuring Algorithmic Differentiation (AD) method. Following, for pyBeam, first, and later on for the whole FSI solver, sensitivities are evaluated by means of AD and validation is proposed against a Finite Difference (FD) approach. Finally, the next steps in this research will be outlined in Sect. 19.5.

19.2 Background

A more-in-detail overview of the FSI framework is presented in this section. The method features an iterative procedure towards the evaluation of the static equilibrium of a flexible wing subjected to a given flow (static aeroelasticity).

19.2.1 Structural FEM Solver

The structural solver pyBeam relies on a 6-dof geometrically nonlinear beam model [16] based on the classic solid mechanics theory:

$$\mathscr{S}(\mathbf{u}) = \mathbf{0} \Leftrightarrow \begin{cases} \nabla \cdot \sigma + \mathbf{F_s} = \mathbf{0} \\ \varepsilon = \varepsilon(\mathbf{u}) \\ \sigma = \mathscr{C} : \varepsilon \end{cases} \tag{19.1}$$

where, in the continuum, σ is the Cauchy stress tensor, $\mathbf{F_s}$ are the structural body forces per unit volume, ε is the strain tensor, $\mathbf{u}$ is the displacement vector and $\mathscr{C}$ is the fourth order stiffness tensor.

The structural problem in Eq. 19.1 shows respectively the equilibrium equation, the strain-displacement equation featuring the geometrical nonlinearity and the constitutive equation for an elastic material. The formulation follows an Updated Lagrangian Approach [17], and small strains are identified from the large displacement field using a corotational strategy [17]. The Euler-Bernoulli beam kinematic assumption is considered.

19.2.2 CFD Solver

We focus on viscous, high-Mach flows around aerodynamic bodies governed by the compressible Navier-Stokes equations. For this purpose, we use the flow solver available in the open-source multiphysics suite SU2.[1] Following the work of Economon et al. [18], the governing equations formulated in conservative form including the energy equation can be written as

[1] https://su2code.g.ithub.io/.

$$\mathscr{F}(\mathbf{w}) = \frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}^c(\mathbf{w}) - \nabla \cdot \mathbf{F}^v(\mathbf{w}) - \mathbf{Q}(\mathbf{w}) = \mathbf{0} \tag{19.2}$$

where $\mathbf{w} = (\rho, \rho\mathbf{v}, \rho E)$ is the vector of conservative variables, ρ the flow density, $\mathbf{v}$ the flow velocity and E the total energy per unit mass. $\mathbf{Q}(\mathbf{w})$ is a generic source term, $\mathbf{F}^c(\mathbf{w})$ and $\mathbf{F}^v(\mathbf{w})$ are, respectively, the convective and viscous fluxes, and can be written as

$$\mathbf{F}^c(\mathbf{w}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \otimes \mathbf{v} + p\mathbf{I} \\ \rho E\mathbf{v} + p\mathbf{v} \end{pmatrix} \tag{19.3}$$

$$\mathbf{F}^v(\mathbf{w}) = \begin{pmatrix} \cdot \\ \boldsymbol{\tau} \\ \boldsymbol{\tau} \cdot \mathbf{v} + \mu^* C_p \nabla T \end{pmatrix} \tag{19.4}$$

where C_p is the specific heat at constant pressure and T is the temperature. The viscous stress tensor is written as

$$\boldsymbol{\tau} = \mu_{tot} \left(\nabla\mathbf{v} + \nabla\mathbf{v}^T - \frac{2}{3}\mathbf{I}(\nabla \cdot \mathbf{v}) \right) \tag{19.5}$$

where, based on the Boussinesq hypothesis [19], the total viscosity μ_{tot} is modelled as a sum of a laminar component which satisfies Sutherland's law [20] and a turbulent component μ_{turb} which is obtained from the solution of a turbulence model. Finally,

$$\mu^* = \frac{\mu_{lam}}{Pr_l} + \frac{\mu_{turb}}{Pr_t} \tag{19.6}$$

where Pr_l and Pr_t are the laminar and turbulent Prandtl numbers, respectively.

19.2.3 Splining Method

To transfer information between the non-conformal structural and CFD grids an in-house Moving Least Square algorithm is implemented [21, 22]. Briefly, given $\mathbf{x_s} \in \mathbb{R}^{N_s}$, the position of the structural nodes and $\mathbf{x_a} \in \mathbb{R}^{N_a}$, the position of the aerodynamic nodes on the moving boundary, it is possible to build a splining matrix $\mathbf{H}_{MLS} = \mathbf{H}_{MLS}(\mathbf{x_s}, \mathbf{x_a}) \in \mathbb{R}^{N_a \times N_s}$ such that:

$$\mathbf{u_a} = \mathbf{H}_{MLS} \cdot \mathbf{u_s}, \tag{19.7}$$

$$\mathbf{f_s} = \mathbf{H}_{MLS}^T \cdot \mathbf{f_a} \tag{19.8}$$

where $\mathbf{u_a}, \mathbf{f_a} \in \mathbb{R}^{N_a}$ and $\mathbf{u_s}, \mathbf{f_s} \in \mathbb{R}^{N_s}$ are, respectively, the displacements/forces defined on the aerodynamic/structural mesh. As already stated in the work of Quaranta et al. [23], employing the transpose of the splining matrix in Eq. (19.8) is enough to ensure the energy conservation. The tool has already been successfully applied to a variety of cases, including the transfer of combined rigid-elastic displacements and featuring mobile surfaces [24] and the interpolation of information between 1D (structural) models and 3D (aerodynamic) ones [15].

19.2.4 Fluid Mesh Deformation Solver

Provided the new position of the moving boundary, and assuming large deformations in the structural domain, it is required to take into account the deformation of the fluid mesh. This is carried out by the SU2 dedicated mesh deformation solver. In order to find the new position of the nodes in the fluid domain the mesh deformation problem can be treated as a pseudo-elastic linear problem [25],

$$\mathbf{K_m} \cdot \mathbf{z} = \tilde{\mathbf{f}} \tag{19.9}$$

where $\mathbf{K_m}$ is a fictitious stiffness matrix and the forces $\tilde{\mathbf{f}}$ are fictiuous forces which ensure the boundary displacement $\mathbf{u_a}$ as for Eq. (19.7).

19.2.5 Coupling Method

A partitioned approach is employed for the FSI solver. This approach, based on the principle of modularity of the different sub-solvers, can be advantageous for practical applications (especially industry-oriented ones) on realistic test-cases.

Defining the three fields under investigation respectively as structural $\mathscr{S}$, fluid $\mathscr{F}$ and mesh $\mathscr{M}$, the whole FSI system $\mathscr{G}$ can be expressed as a function of the state variables $\mathbf{u}$, $\mathbf{w}$ and $\mathbf{z}$, respectively, structural displacements, aerodynamic state variables and fluid mesh nodes displacements [26]

$$\mathscr{G}(\mathbf{u}, \mathbf{w}, \mathbf{z}) = \begin{cases} \mathscr{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}) = 0, \\ \mathscr{F}(\mathbf{w}, \mathbf{z}) = 0, \\ \mathscr{M}(\mathbf{u}, \mathbf{z}) = 0, \end{cases} \tag{19.10}$$

in which the coupling contributions given by the splining procedure have already been included. The structural beam solver $\mathscr{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}) = 0$ has been developed for this work and operates as a C++ library wrapped with python using SWIG [27]. The displacements of the structure are accessible from a python script which acts as an orchestrator. They are interpolated into the fluid boundary using Eq. (19.7) and the spline C++ library which is also wrapped to be accessed from the python orchestrator. The fluid boundary displacements are then imposed onto the mesh solver in SU2 via its Application Programming Interface (API) [28]. A new value of the aerodynamic forces on the boundary is obtained after a CFD simulation in SU2 and interpolated back into the structural beam model using Eq. (19.8).

Due to the nonlinear nature of the FSI problem and given the partitioned approach used, a Block–Gauss–Seidel (BGS) strategy is adopted in the python orchestrator, which allows the sequential solution of the three problems within the single FSI iteration. This corresponds to a linearization of the problem as

$$\begin{bmatrix} \frac{\partial \mathscr{S}}{\partial \mathbf{u}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{F}}{\partial \mathbf{w}} & 0 \\ \frac{\partial \mathscr{M}}{\partial \mathbf{u}} & 0 & \frac{\partial \mathscr{M}}{\partial \mathbf{z}} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{w} \\ \Delta \mathbf{z} \end{Bmatrix} = - \begin{Bmatrix} \mathscr{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}) \\ \mathscr{F}(\mathbf{w}, \mathbf{z}) \\ \mathscr{M}(\mathbf{u}, \mathbf{z}) \end{Bmatrix}, \tag{19.11}$$

in which the upper right part of the problem matrix has been set to 0 [29]. To ensure the stability of the method, a relaxation parameter α is applied to the boundary displacements:

$$\mathbf{u}_a^* = \alpha \mathbf{u}_a^n + (1 - \alpha)\mathbf{u}_a^{n-1}. \tag{19.12}$$

where n is the current and $n - 1$ is the previous BGS subiteration. An overview of the framework layout is given in Figure 19.1.

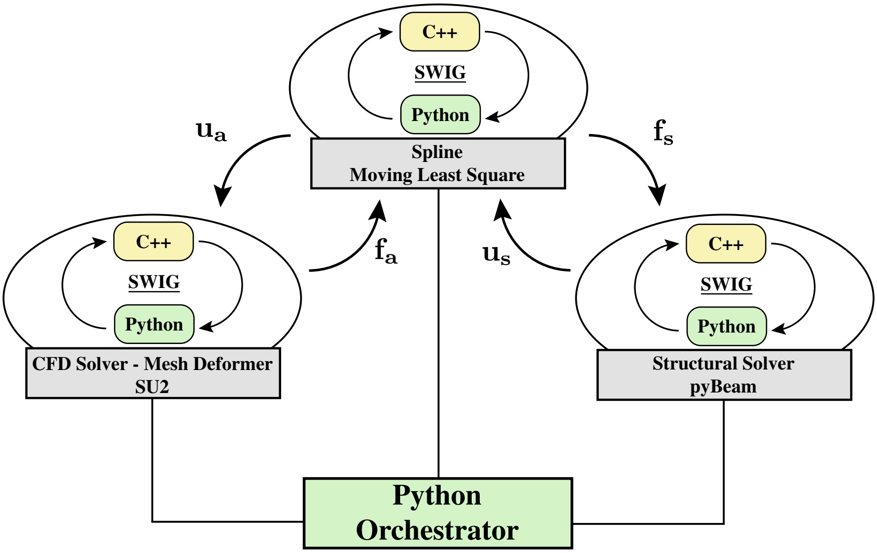


Fig. 19.1 Framework layout

19.3 Application

As a preliminary test case to test the framework, we employ an aeroelastic model based on the ONERA M6 wing geometry [30]. The structural model is similar to the one assembled for the purpose of aeroelastic analysis which has been used in a previous work by Bombardieri et al. [15]. It features a wing-box located at 1/4th of the chord of the wing and described by beam elements as shown in Fig. 19.2. For every structural node along the wing box, four nodes have been placed to reproduce the airfoil leading edge, trailing edge, upper and lower point positions. This solution has been found to be successful for a correct implementation of the spline algorithm introduced in Sect. 19.2.3 in order to transfer information between the structural and the CFD surface meshes [15].

Concerning the fluid part of the problem, for this first application, and without loss of generality, the flow has been modeled by the Euler solver of SU2, as a compromise between computational efficiency and accuracy of the results. The CFD mesh consists of 582,752 tetrahedral elements and 108,396 nodes. The wing boundary features 36,454 triangular elements and 18,285 nodes (Fig. 19.2). Among the different options provided by SU2 to perform the simulation, a 3 level Multi-Grid scheme has been used together with a 2nd order in space Jameson–Schmidt–Turkel (JST) scheme.

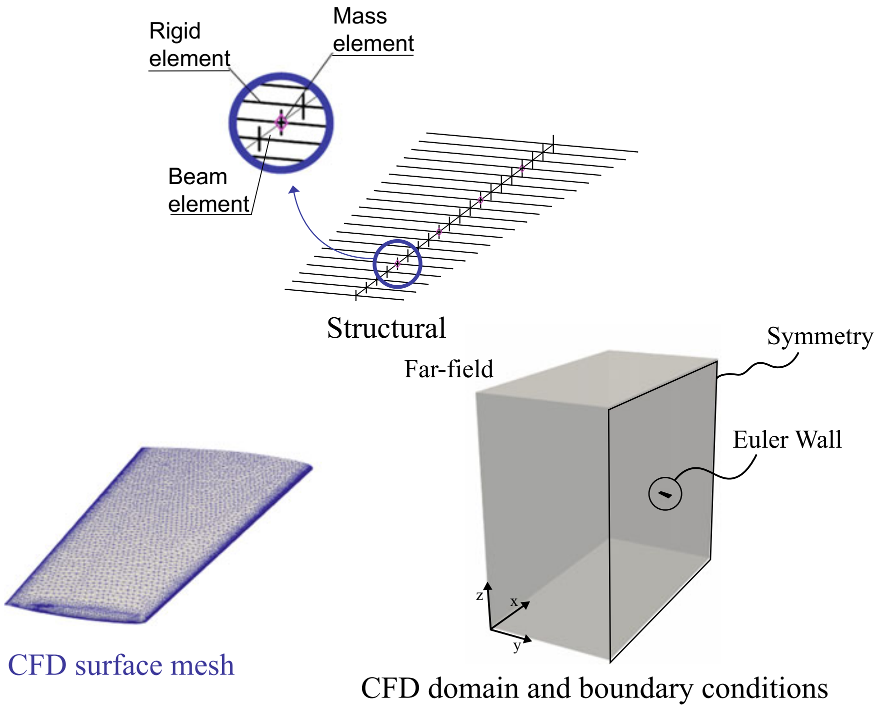


Fig. 19.2 Different meshes for the ONERA M6 test-case (from Bombardieri et al. [15])

19.3.1 Structural Solver Validation

The solver pyBeam is validated comparing the results with the ones computed using the nonlinear structural solver of the commercial solver *NASTRAN* [31] (SOL 106). Two test-cases are considered. In CASE 1 a load is applied to the wing box: $P = (0.8,\ 0,\ 0.16)\ [N]$ (Fig. 19.4) in correspondence of the tip. Such load features a component directed along x (flow direction) and one along z, normal to the wing plane direction and has been adequately chosen for the structure to exhibit nonlinear response. Figure 19.3 shows the deformed structure under the applied load. Displacements are compared for the five nodes describing the airfoil at the tip of the wing (Fig. 19.4). Table 19.1 shows the differences in percentage between pyBeam and *NASTRAN*, where a good agreement for all five nodes is found.

For CASE 2 a more realistic load set is employed, interpolating from the aerodynamic surface to the structural grid the pressure distribution resulting from a CFD simulation performed at Mach 0.839 and a wing Angle of Attack (AoA) of 3°. Table 19.2 shows the differences in percentage between the two solvers for the 5 considered nodes. It can be observed how, for both load cases, differences are negligible. It is also worth mentioning that, for this last case, the conservation of the forces interpolated from fluid surface grid to structural mesh has been verified.

Fig. 19.3 Deformed configuration for CASE 1 validation. In green the undeformed configuration, in blue the deformed one

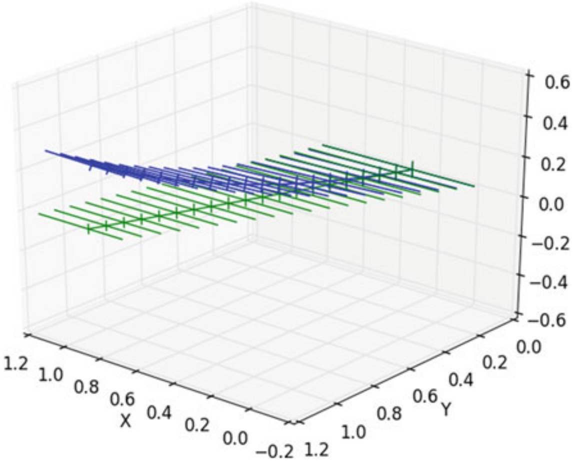


Fig. 19.4 Tip cross section. The five structural nodes used for comparison with *NASTRAN* and the load vector for CASE 1

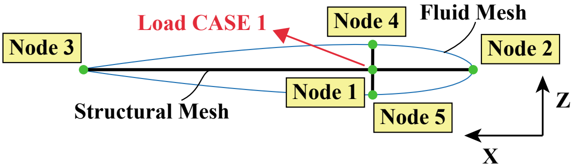


Table 19.1 Comparison between in-house structural solver and *NASTRAN* for CASE 1

	Δx (%)	Δy (%)	Δz (%)
Node 1	0.24	0.28	0.012
Node 2	0.24	0.32	0.14
Node 3	0.24	0.24	0.09
Node 4	0.25	0.27	0.12
Node 5	0.22	0.30	0.13

Table 19.2 Comparison between in-house structural solver and *NASTRAN* for CASE 2

	Δx (%)	Δy (%)	Δz (%)
Node 1	0.003	0.003	0.001
Node 2	0.004	0.005	0.002
Node 3	0.001	0.003	0.001
Node 4	0.003	0.002	0.001
Node 5	0.002	0.202	0.001

19.3.2 Primal FSI Solver

Applications of the primal solver are here shown for two cases: asymptotic flow at Mach 0.839 with AoA 3.06 and 2.5°. Figure 19.5 shows the converged configuration for the two cases compared with the undeformed one, together with the Mach distribution over the configuration at AoA 3.06°.

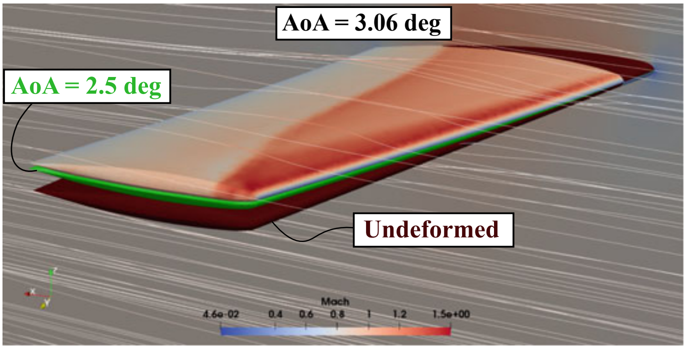


Fig. 19.5 Application of the primal solver. Undeformed configuration (in brown) is compared with the converged configuration at AoA of 2.5° (in green) and at AoA of 3.06° in case of $M_\infty = 0.839$. For the converged one with AoA of 3.06° also the Mach distribution over the surface is shown

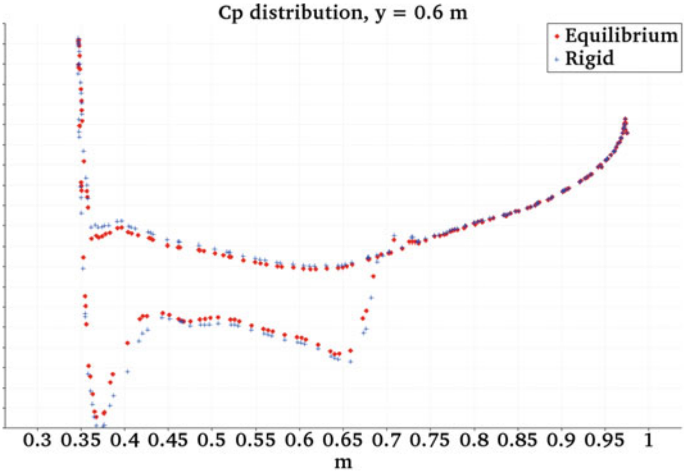


Fig. 19.6 Pressure distribution at section $y = 0.6\,\text{m}$ for the rigid and elastic configuration at AoA of 2.5°

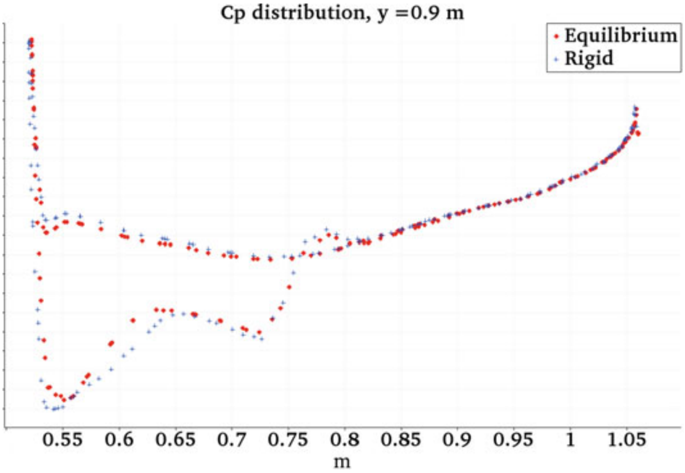


Fig. 19.7 Pressure distribution at section $y = 0.9\,\text{m}$ for the rigid and elastic configuration at AoA of 2.5°

Figures 19.6 and 19.7 show the comparison of the Cp distribution between the undeformed configuration and the converged configuration at AoA of 2.5° at two positions along the span ($y = 0.6\,\text{m}$ and $y = 0.9\,\text{m}$). It can be noticed, at both sections, differences in Cp distribution, especially in the area interested by the lambda shock.

19.4 Adjoint Based Optimization: Sensitivities Evaluation

Let the governing equations in Eq. (19.10) be rewritten in the form of fixed-point iterators,

$$\mathscr{G}(\mathbf{u},\mathbf{w},\mathbf{z})=0 \Leftrightarrow \begin{cases} \mathrm{S}(\mathbf{u},\mathbf{w},\mathbf{z})-\mathbf{u}=0 & (19.13\mathrm{a})\\ \mathrm{F}(\mathbf{w},\mathbf{z})-\mathbf{w}=0 & (19.13\mathrm{b})\\ \mathrm{M}(\mathbf{u})-\mathbf{z}=0 & (19.13\mathrm{c})\end{cases}$$

In previous works by Sanchez et al. [13] it was shown that rewriting the standard FSI problem into the form of Eq. (19.13) leads to an efficient redefinition of the adjoint problem in fixed-point form amenable to the use of AD, which provides a seamless infrastructure to compute coupled sensitivities in unified code-bases.

Defining the objective function $J(\mathbf{u},\mathbf{w},\mathbf{z},\boldsymbol{\alpha})$ and a set of design variables $\boldsymbol{\alpha}$, the Lagrangian operator, $\mathscr{L}$, is defined as:

$$\begin{aligned}\mathscr{L}(\mathbf{u},\bar{\mathbf{u}},\mathbf{w},\bar{\mathbf{w}},\mathbf{z},\bar{\mathbf{z}},\boldsymbol{\alpha}) = {} & J(\mathbf{u},\mathbf{w},\mathbf{z},\boldsymbol{\alpha})+\bar{\mathbf{u}}^T\left[\mathbf{S}(\mathbf{u},\mathbf{w},\mathbf{z},\boldsymbol{\alpha})-\mathbf{u}\right]\\ & +\bar{\mathbf{w}}^T\left[\mathbf{F}(\mathbf{w},\mathbf{z},\boldsymbol{\alpha})-\mathbf{w}\right]+\bar{\mathbf{z}}^T\left[\mathbf{M}(\mathbf{u},\boldsymbol{\alpha})-\mathbf{z}\right],\end{aligned} \tag{19.14}$$

where the Lagrangian multipliers $\bar{\mathbf{u}}$, $\bar{\mathbf{w}}$ and $\bar{\mathbf{z}}$ correspond to the adjoints of the state variables. Imposing the KKT conditions, the sensitivity of the objective function might be computed

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\alpha}}^T=\frac{\partial J}{\partial\boldsymbol{\alpha}}^T+\frac{\partial\mathbf{S}}{\partial\boldsymbol{\alpha}}^T\bar{\mathbf{u}}+\frac{\partial\mathbf{F}}{\partial\boldsymbol{\alpha}}^T\bar{\mathbf{w}}+\frac{\partial\mathbf{M}}{\partial\boldsymbol{\alpha}}^T\bar{\mathbf{z}}, \tag{19.15}$$

where the adjoint variables are obtained from the fixed-point adjoint equations

$$\bar{\mathbf{u}}=\frac{\partial J}{\partial\mathbf{u}}+\frac{\partial\mathbf{S}}{\partial\mathbf{u}}^T\bar{\mathbf{u}}+\frac{\partial\mathbf{M}}{\partial\mathbf{u}}^T\bar{\mathbf{z}}, \tag{19.16a}$$

$$\bar{\mathbf{w}}=\frac{\partial J}{\partial\mathbf{w}}+\frac{\partial\mathbf{F}}{\partial\mathbf{w}}^T\bar{\mathbf{w}}+\frac{\partial\mathbf{S}}{\partial\mathbf{w}}^T\bar{\mathbf{u}}, \tag{19.16b}$$

$$\bar{\mathbf{z}}=\frac{\partial J}{\partial\mathbf{z}}+\frac{\partial\mathbf{F}}{\partial\mathbf{z}}^T\bar{\mathbf{w}}+\frac{\partial\mathbf{S}}{\partial\mathbf{z}}^T\bar{\mathbf{u}}. \tag{19.16c}$$

The matrix-vector products in the general form of $\frac{\partial\mathbf{F}}{\partial\mathbf{x}}^T\bar{\mathbf{y}}$ can be evaluated using the AD tool CoDiPack [32]. In the case of a python infrastructure with non-conformal interfaces as the one proposed in this work, the crossed dependencies

$$\frac{\partial\mathbf{M}}{\partial\mathbf{u}}^T\bar{\mathbf{z}},\quad \frac{\partial\mathbf{S}}{\partial\mathbf{w}}^T\bar{\mathbf{u}},\quad \frac{\partial\mathbf{F}}{\partial\mathbf{z}}^T\bar{\mathbf{w}},\quad \frac{\partial\mathbf{S}}{\partial\mathbf{z}}^T\bar{\mathbf{u}}, \tag{19.17}$$

must be handled carefully taking into account the interpolation steps.

Table 19.3 Structural sensitivities of OF $J = 0.0590\,\text{m}$ to the vertical loads applied to the five nodes of the tip section (Fig. 19.4) calculated using the FD approach and the ADR method, for a nominal equilibrium condition under the load set of CASE 3

	dJ/dF_{1_z}	Relative Error to FD	dJ/dF_{2_z}	Relative Error to FD
FD	2.282116112499	–	2.034943501002	–
ADR	2.282121475229	2.3499e-04 %	2.034946975076	1.7072e-04 %
	dJ/dF_{3_z}	Relative Error to FD	dJ/dF_{4_z}	Relative Error to FD
FD	2.652881587618	–	2.277763717319	–
ADR	2.652889131720	2.8437e-04 %	2.277769271153	2.4383e-04 %
	dJ/dF_{5_z}	Relative Error to FD		
FD	2.286468140094	–		
ADR	2.286473677823	2.4220e-04 %		

19.4.1 Structural Sensitivities

The structural solver pyBeam has been developed to handle AD in a similar manner as it was done for the aerodynamic solver in SU2 [11] and for the native solid solver in SU2 [13]. Proof of concept of the structural sensitivities evaluation using the AD method is presented here.

AD implementation is demonstrated on the ONERA M6 structural model. For the objective function (OF) J, sensitivity with respect to the chosen design parameter is calculated both with a classic FD approach and AD reverse (ADR) method [13]. The objective function is chosen to be the vertical displacement of the tip node (Node 1 in Fig. 19.4). Sensitivities are calculated for a nominal equilibrium condition under the load set used for validation in CASE 2 presented Sect. 19.3.1.

Comparison of sensitivities with respect to the vertical component of the forces applied to each node is shown for the five nodes of the tip section as for Fig. 19.4 (Table 19.3) and for the respective nodes at a mid-span section (Table 19.4). Comparison with finite differences shows excellent accuracy of the gradients computed with the adjoint method.

19.4.2 FSI Sensitivities

Finally, for the full FSI framework, proof of concept of AD-based coupled sensitivities evaluation is here presented. The calculation of coupled sensitivities (i.e. sensitivities of an aerodynamic objective function with respect to a structural design variable or vice versa) represents a key feature for the framework to be used in the context of aero-structural optimization.

Table 19.4 Structural sensitivities of the OF J to the vertical loads applied to the five nodes of the mid-span section (ordered as in Fig. 19.4) calculated using the FD approach and the ADR method, for a nominal equilibrium condition under the load set of CASE 3 ($J = 0.0590\,\text{m}$)

	dJ/dF_{1_z}	Relative Error to FD	dJ/dF_{2_z}	Relative Error to FD
FD	0.521372402249	–	0.341243158041	–
ADR	0.521372693595	5.5881e-05 %	0.341243293520	3.9702e-05 %
	dJ/dF_{3_z}	Relative Error to FD	dJ/dF_{4_z}	Relative Error to FD
FD	0.791564714791	–	0.518704634402	–
ADR	0.791566479154	2.2290e-04 %	0.5187052129698	1.1154e-04 %
	dJ/dF_{5_z}	Relative Error to FD		
FD	0.524039945244	–		
ADR	0.524040172905	4.3444e-05 %		

Table 19.5 FSI sensitivities of the OF C_D and C_L with respect to the Young Modulus E, calculated using FD approach and the ADR method. Nominal equilibrium condition at $M_\infty = 0.84$, $AoA = 3.06$ and $E = 4.0e + 10\ Pa$

	dC_D/dE	Relative Error to FD	dC_L/dE	Relative Error to FD
FD	7.175227176e-14	–	1.208098392e-12	–
ADR	7.172713849e-14	0.0350 %	1.208552955e-12	0.0376 %

Sensitivities are evaluated for the presented aeroelastic model, at nominal conditions characterized by asymptotic flow at Mach 0.84 with AoA of 3.06° and Young Modulus of the structure E of 4.0e+10 10 Pa. For the lift coefficient C_L and drag coefficient C_D, AD-based sensitivities with respect to E are compared to FD-based ones in Table 19.5. Comparison shows, again, excellent agreement between sensitivities calculated by the two methods.

Figure 19.8 shows the OF C_D and C_L evaluated by the primal FSI solver and the relative sensitivities evaluated by the adjoint fixed-point as a function of the iteration number. Similar convergence trends can be observed which may be linked to the fact that the adjoint fixed-point method inherits the convergence properties of the primal one [26]. The convergence of the coupled FSI adjoint problem within the presented framework will be further studied in coming works.

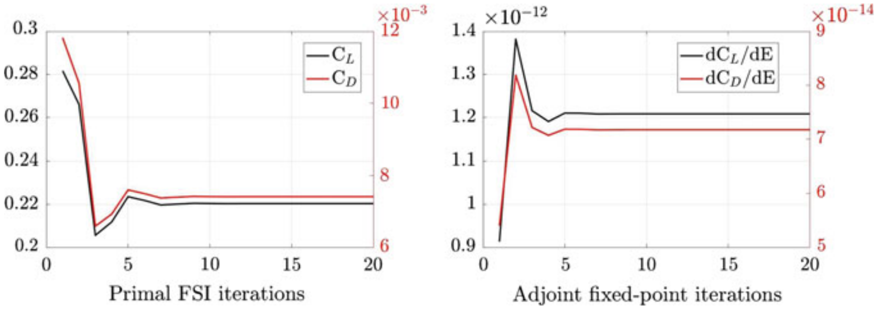


Fig. 19.8 Objective functions C_D and C_L for the primal FSI problem (left) and relative sensitivities for the adjoint fixed-point one (right) as a function of iteration number

19.5 Conclusions and Future Works

An ongoing effort is presented for the development of an open-source framework for the analysis, design and optimization of wing aero-structural problems. This framework is based on the established open-source CFD solver available within the multiphysics suite SU2, and an in-house open-source nonlinear beam solver. A splining algorithm, based on Moving Least Squares, is implemented following the structural solver architecture and provides conservative interpolation over the non-conformal FSI interfaces. All the computational solvers are accessible from python via a SWIG compilation and communicate over the common interface using a fit-for-purpose python framework.

First, the structural solver is validated taking as a reference NASTRAN's non-linear structural solver SOL 106: it is shown how the differences in the predicted displacements of the test-case structure are negligible for two distinct load cases. Secondly, a preliminary run of the primal FSI solver is shown featuring an aeroelastic model of the ONERA M6 wing. It is shown how, with asymptotic conditions of Mach 0.839 with Angles of Attack 2.5 and 3.06°, the FSI system converges to equilibrium configurations different from the rigid one.

This work is part of a bigger effort to build a self-contained tool for rapid analysis and optimization of very flexible wings coupling CFD and nonlinear structural FEM for the structural part. It has been demonstrated, in previous works, the applicability of SU2 to aeroelastic analysis [15] and fully-coupled FSI sensitivity analysis with its native solid mechanics solver [13]. With this aim, a first effort is done in the direction of the evaluation of FSI coupled sensitivities for the presented framework using Algorithmic Differentiation. First, AD based structural sensitivities are validated. For a nominal equilibrium condition of the presented aeroelastic test-case, sensitivities are calculated for a variety of parameters and compared with the same evaluated using Finite Differences. For all the tests, correlation is found to be satisfactory. Finally, validation of FSI sensitivities calculation is sought. For a reference asymptotic flow and nominal structural properties of the presented aeroelastic test case, AD-based sensitivities of coefficients C_L and C_D with respect to the structure's Young Modulus are compared with the same values calculated using Finite Differences. Again, results show excellent accuracy in gradient evaluation.

PyBeam organization on GitHub provides the complete set of test cases discussed above in the repository $Testcases\_Eurogen\_2019$.

References

1. Wright J, Cooper J (2014) Introduction to aircraft aeroelasticity and loads. John Wiley & Sons Ltd
2. Biedron RT, Carlson J-R, Derlaga JM, Gnoffo PA, Hammond DP, Jones WT, Kleb B, Lee-Rausch EM, Nielsen EJ, Park MA, Rumsey CL, Thomas JL, Thompson KB, Wood WA (2019) FUN3D manual: 13.5. https://fun3d.larc.nasa.gov
3. Brezillon J, Ronzheimer A, Haar D, Abu-Zurayk M, Lummer M, Krüger W, Natterer FJ (2012) Development and application of multi-disciplinary optimization capabilities based on high-fidelity methods. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 2012
4. Cambier L, Gazaix M (2002) ElsA: an efficient object-oriented solution to CFD complexity. In: 40th AIAA aerospace sciences meeting and exhibit
5. Economon T, Palacios F, Copeland S, Lukaczyk T, Alonso J (2016) "SU2": an open-source suite for multiphysics simulation and design. AIAA J 54(3):828–846
6. Palacios F, Economon TD, Wendorff AD, Alonso JJ (2015) Large-scale aircraft design using SU2. In: 53rd AIAA aerospace sciences meeting
7. Pini M, Vitale S, Colonna P, Gori G, Guardone A, Economon T, Alonso J, Palacios F (2017) SU2: the open-source software for non-ideal compressible flows. J Phys: Conf Ser 821(1):012013
8. Gori G, Vimercati D, Guardone A (2017) Non-ideal compressible-fluid effects in oblique shock waves. J Phys: Conf Ser 821(1):012003
9. Molina ES, Spode C, Da Silva RGA, Manosalvas-Kjono DE, Nimmagadda S, Economon TD, Alonso JJ, Righi M (2017) Hybrid RANS/LES calculations in SU2. In: 23rd AIAA computational fluid dynamics conference 2017
10. Zhou BY, Albring T, Gauger NR, Ilario C, Economon T, Alonso JJ (2017) Reduction of airframe noise components using a discrete adjoint approach. AIAA 2017-3658
11. Albring T, Sagebaum M, Gauger N (2016) Efficient aerodynamic design using the discrete adjoint method in SU2. In: 17th AIAA/ISSMO multidisciplinary analysis and optimization conference
12. Sanchez R, Palacios R, Economon T, Kline H, Alonso J, Palacios F (2016) Towards a fluid-structure interaction solver for problems with large deformations within the open-source SU2 suite. In: 57th AIAA SDM conference, AIAA SciTech, San Diego, CA, 4–8 January 2016
13. Sanchez R, Albring T, Palacios R, Gauger NR, Economon TD, Alonso JJ (2018) Coupled adjoint-based sensitivities in large-displacement fluid-structure interaction using algorithmic differentiation. Int J Numer Methods Eng 113(7):1081–1107
14. Venkatesan-Crome C, Sanchez R, Palacios R (2018) Aerodynamic optimization using FSI coupled adjoints in SU2. In: 6th European conference on computational mechanics (ECCM 6), Glasgow, UK, p 5
15. Bombardieri R, Cavallaro R, Luis Sáez de Teresa J, Karpel M Nonlinear aeroelasticity: a CFD-based adaptive methodology for flutter prediction, no. AIAA 2019-1866. In: AIAA Scitech 2019 Forum, San Diego, California, 7–11 January 2019
16. Levy R, Spillers W (2003) Analysis of geometrically nonlinear structures, vol 1. Kluwer Academic Publishers, Dordrecht, Netherlands

17. Belytschko T, Liu W, Moran B (2000) Nonlinear finite elements for continua and structures. Wiley
18. Economon TD, Palacios F, Copeland SR, Lukaczyk TW, Alonso JJ (2016) SU2: an open-source suite for multiphysics simulation and design. AIAA J **54**(3), 828–846
19. Wilcox D (1998) Turbulence modeling for CFD. DCW Industries, Inc.
20. White FM (1974) Viscous Fluid Flow. McGraw-Hill, New York
21. Romanelli G, Castellani M, Mantegazza P, Ricci S (2012) Coupled CSD/CFD non-linear aeroelastic trim of free-flying flexible aircraft. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA - Honolulu, Hawaii
22. Cavallaro R, Iannelli A, Demasi L, Razón AM (2015) Phenomenology of nonlinear aeroelastic responses of highly deformable joined wings. Adv Aircr Spacecr Sci 2(2):125–168
23. Quaranta G, Masarati P, Mantegazza P (2005) A conservative mesh-free approach for fluid structure problems in coupled problems. In: International conference for coupled problems in science and engineering, Santorini, Greece, 23–29 May 2005, pp 24–27
24. Cavallaro R, Bombardieri R, Demasi L, Iannelli A (2015) PrandtlPlane joined wing: body freedom flutter, limit cycle oscillation and freeplay studies. J Fluids Struct 59:57–84
25. Dwight R (2009) Robust mesh deformation using the linear elasticity equations. Springer, Berlin, pp 401–406
26. Sanchez R (2018) Coupled adjoint-based sensitivities in large-displacement fluid-structure interaction using algorithmic differentiation. PhD thesis, Imperial College London
27. Beazley DM (1996) SWIG: an easy to use tool for integrating scripting languages with C and C++. In: Proceedings of the 4th conference on USENIX Tcl/Tk workshop, TCLTK'96, USENIX Association, Berkeley, CA, USA, vol 4, pp 15–15
28. Sanchez R, Kline H, Thomas D, Variyar A, Righi M, Economon T, Alonso J, Palacios R, Dimitriadis G, Terrapon V (2016) Assessment of the fluid-structure interaction capabilities for aeronautical applications of the open-source solver SU2. In: VII European congress on computational methods in applied sciences and engineering (ECCOMAS 2016), Crete Island, Greece, 5–10 June 2016
29. Barcelos M, Maute K (2008) Aeroelastic design optimization for laminar and turbulent flows. Comput Methods Appl Mech Eng 197(19–20):1813–1832
30. Schmitt V, Charpin F (1979) Pressure distributions on the ONERA-M6-wing at transonic mach numbers. In: Experimental data base for computer program assessment AGARD AR 138, AGARD, May 1979
31. MSC-Software (2011) MSC Nastran 2012 quick reference guide. MacNeal-Schwendler Corp.
32. Sagebaum M, Albring T, Gauger NR (2017) High-performance derivative computations using CoDiPack. arXiv:1709.07229

Chapter 20
Neuroevolutionary Multiobjective Optimization of Injection Stretch Blow Molding Process in the Blowing Phase

Renê S. Pinto, Hugo M. Silva, Fernando M. Duarte, João P. Nunes, and António Gaspar-Cunha

Abstract Injection stretch blow molding is a very important thermoplastic processing technique producing hollow containers with mechanical performance. One of the main challenges in optimizing this process consists in finding the best thickness profile for each part in order to achieve the desired mechanical properties with less material use. In a previous study, a new methodology based on a neuroevolutionary multiobjective optimization approach was proposed to enhance the entire process, which considers that the process is optimized by phases, starting by the end. In that initial study only the final phase of the process was addressed, where the best thickness profile for an industrial bottle was found in order to satisfy the required mechanical properties with less material use. In the present study, the focus is the second stage of the optimization methodology, concerning the blowing phase of injection blow molding process. The optimal results obtained in the first phase are used as the optimal thickness profile for the bottle with the goal to find the best preform thickness profile which produces the desired bottle. The same procedures are used and the results show that the methodology was successfully applied to its second phase.

Keywords Neuroevolutionary · Multi-objective optimization · Plastics blow-moulding

20.1 Introduction

Injection stretch blow molding is one of the most important processes in the industry to produce hollow plastic containers, such as bottles, jars and several kind of different hollow plastic parts. Basically, this thermoplastic processing technique comprises the following steps: (1) injection of molten raw material into a cavity to produce

R. S. Pinto · H. M. Silva · F. M. Duarte · J. P. Nunes · A. Gaspar-Cunha (✉)
Institute of Polymers and Composites, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: agc@dep.uminho.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_20

the desired shape of the preform; (ii) heating the preform, typically by radiation, so that the material acquires deformation capability; (iii) stretch and blowing the heated preform in order to ensure that the preform reproduces the contours of the mold. The stretch, made mechanically by the action of a plug, and the blowing, using air under pressure, can occur sequentially (stretch followed by blowing) or at the same time; (iv) finally, the part is cooled and removed from the mold.

Since the amount of material used in blow molded products represents a significant share of the total manufacture costs, the minimization of material utilization is required [1]. However, there are several important mechanical properties which are also dependent on this feature. Numerical approaches can be applied to avoid empirical tests to find the process input variables which gives the best tradeoff between the material utilization and the desired mechanical properties. Several studies in the literature present different approaches concerning injection stretch blow molding design process and optimization [1–6]. One of the major challenges in optimizing this process is to define the complete thickness profile and shape of the final part and of the preform in order to achieve desired mechanical properties with less material utilization.

In [1, 2] a global optimization methodology for injection stretch blow molding process is presented and detailed. This methodology uses a neuroevolutionary multi-objective approach and is composed by steps (or phases) that should be performed to optimize the whole process in order to find the best thickness profile of the final part and of the preform. In both studies only the first phase of the optimization, which comprises the final stage of manufacturing process, is addressed. This study focuses on the second phase of the optimization methodology, which comprises the blowing of a stretched preform in the manufacturing process. In the previous study, optimal thickness distributions of the final part were obtained. The main goal of this study is find the best thickness distributions of the preform that will lead to final parts (after the blowing phase) with the optimal thickness profiles found on the previous study.

20.2 Global Optimization Methodology

The proposed optimization methodology for the injection stretch blow molding summarizes the whole process in five main phases: Injection, Stretching, Blowing, Mold opening and Blow-molded part. The injection phase comprises the melting of raw material and its injection into a cavity to form the preform. The stretching phase, that is not always present in the manufacturing process, comprises the stretching of the preform in order to maximize the amount of material at the bottom of the final part. The blowing phase comprises the injection of air under pressure to expand the preform towards the mold, acquiring its shape. Mold opening comprises the phase where the mold is opening and the final part is pulled out. Finally, Blow-molded part comprises the last production phase, where the final part is cooled and becomes ready for packaging.

After summarizing the five main phases in the blow molding process, the optimization methodology establishes four phases (or steps) for the optimization process (O1 to O4). However, the optimization starts by the last production phase, i.e., when the final part is done. The first optimization step (O1) consists in optimizing the thickness profile of the final part, i.e., to find the best thickness profile of the final part which provides the desired mechanical properties with less material utilization. Step O2 consists in optimizing the preform thickness profile after stretching, i.e., to find the thickness profile of the preform which will produce (after blowing) the final part with the optimal profile found in step O1. This study concerns on this phase. The step O3 comprises in the optimization of the preform thickness profile before stretch, i.e., finding the thickness profile of the preform (before stretch) that will produce (after stretch) the preform with the optimal profile found in step O2. Finally, the step O4 also optimizes the preform thickness profile, but injection conditions and cavity geometry are used as decision variables.

20.2.1 Neuroevolutionary Multiobjective Optimization

One of the insights of the proposed methodology is treat a container's thickness distribution as a function of its geometry. In this context, Artificial Neural Networks (ANNs) are built to compute the wall thickness at any location of the part (based on the corresponding 3D coordinate). To allow many evaluations throughout the optimization process, simulations are carried out through finite element models (FEMs). Thus, by using ANNs the search space can be drastically reduced once each FEM model is composed by a 3D mesh with thousands or even millions of points. In the evolutionary algorithm, each solution is represented by an ANN which gives a thickness distribution profile for a given FEM (3D mesh). The attributes of the ANN are evolved to find the networks that give optimal distributions. Figure 20.1 illustrates the ANN representation.

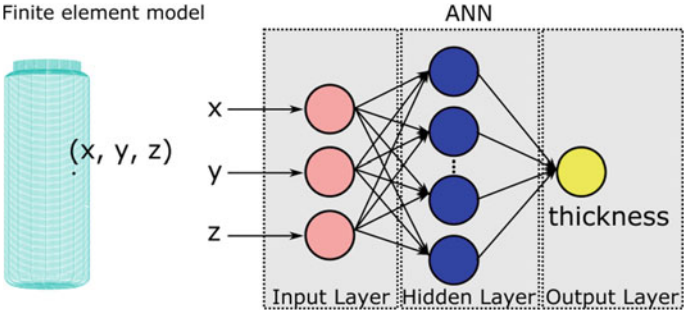


Fig. 20.1 A FEM model (bottle) mesh. Each coordinate of the mesh is an input to the ANN to calculate the thickness in the corresponding point

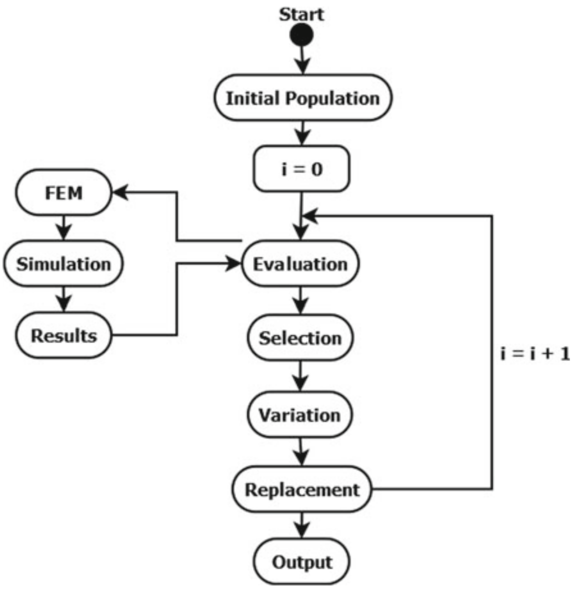


Fig. 20.2 Neuroevolutionary optimization methodology workflow

The multiobjective optimization evolutionary algorithm of the methodology is based on the SMS-EMOA [7]. Figure 20.2 illustrates the basic workflow for the optimization.

Each population is composed of a set of individuals (solutions), each one representing an ANN. The weights and biases of the ANN are encoded in a real number chromosome. Thus, the size of the chromosome depends of the ANN topology instead of the size of FEM model mesh. The initial population is generated randomly.

To evaluate a solution, the coordinates of each point of a given mesh are feed into the ANN to define the thickness in each one of the points, forming the thickness profile that is considered as the input by the simulation process. In the evolutionary algorithm, selection is performed by a uniform distribution and variation is performed by the SBX-Crossover operator, which is designed to work with real number representations. Replacement strategy is based on Pareto front and *hypervolume* [8] measure. As a result of the optimization process, there will be a set of optimal solutions where each of them represents an ANN that gives the wall thickness distribution for the model mesh. All solutions will provide different tradeoffs between the considered objectives, such as mass versus mechanical properties, for instance.

20.3 Experimental Design

In the first phase of the optimization methodology an industrial bottle model was considered in the experiments. The bottle is 45 mm in diameter and 182 mm in

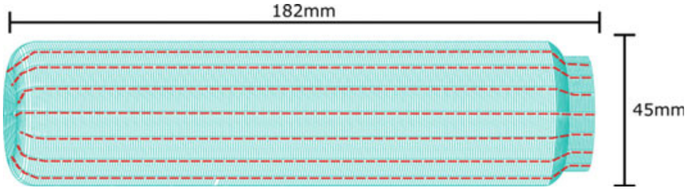


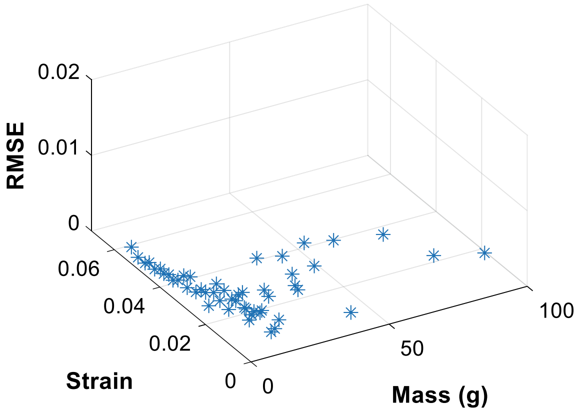
Fig. 20.3 Bottle model with vertical lines of the mesh highlighted

height, composed by plastic material with mass density of 1.15 g/cm^3 and Poison's ratio of 0.4. The applied air (blowing) pressure and Young's module ratio is 0.0027. Figure 20.3 shows the 3D mesh model with dimensions indicated. The vertical lines of the mesh are highlighted to illustrate the points where the wall thickness is calculated.

In the first phase three objective functions were considered to be minimized: f_1, the total mass of the bottle; f_2, the maximum strain suffered by the bottle and f_3, which is the maximum difference between the thickness profile of all vertical lines, measured by RMSE index (root mean square error). This objective measures how uniform is the thickness distribution, since the same thickness profile for all vertical lines, i.e., along the bottle, is desirable.

Figures 20.4 and 20.5 show the Pareto front of the final population for the first phase of the optimization. In Fig. 20.4 it can be seen that all solutions have low value for RMSE error (all below 0.01), which means that the algorithm was able to find uniform distributions. In Fig. 20.5 only the objectives f_1 and f_2 are plotted. All solutions are well distributed along the Pareto curve, providing different tradeoffs between the total mass (f_1) and the maximum strain (f_2). Five optimal solutions (S1 to S5) are highlighted in the curve. Solutions S2 and S3, which are located in the knee area, are considered to give the best (balanced) relationship between f_1 and f_2 objectives. Thus, they were considered as the optimal designs to be achieved by the second optimization phase on this study.

Fig. 20.4 Pareto front for final population for the first phase of the optimization process



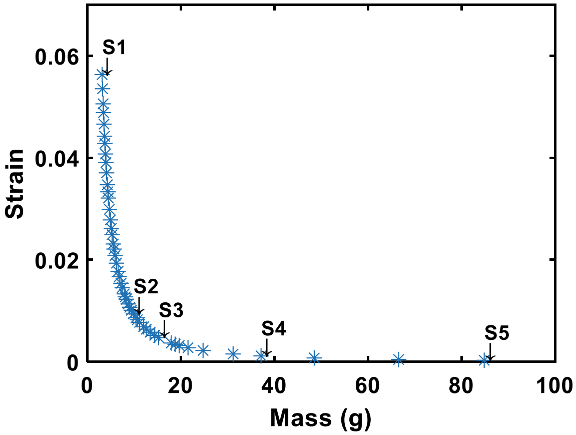


Fig. 20.5 Pareto front of the first phase with only objective functions f_1 and f_2 plotted

Figure 20.6 shows the thickness distribution for the optimal solutions S2 and S3. The x-axis comprises the points located from the bottleneck towards the bottom of the bottle. Each line (distribution) represents the mean thickness values between all vertical lines of the mesh (Fig. 20.3). Both distributions presented the same behavior. Concerning solution S2, all the points presented mean thickness values of 0.30 mm which decreases faster up to 0.03 mm when reaches the bottom of the bottle. Solution S3 had the same behavior, but the mean thickness value was 0.58 mm decreasing up to 0.06 mm at the bottom. From a physical point of view, these results make sense. In S2, the bottle wall is thinner, using less material, but it suffers more strain than solution S3, where the wall is thicker, using more material, but it suffers lower maximum strain. Tab. 1 lists f_1 (total mass) and f_2 (maximum strain) values for both solutions (Table 20.1).

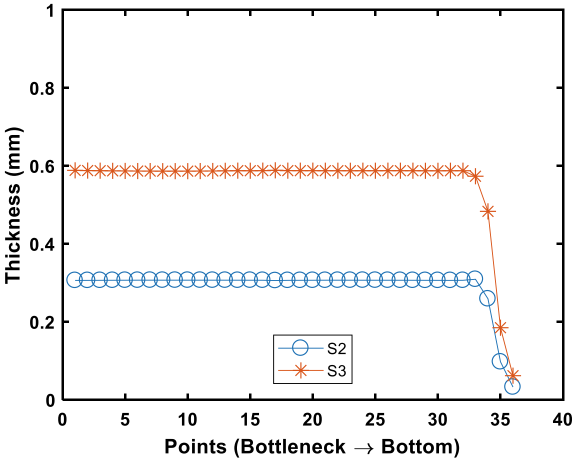


Fig. 20.6 Thickness distribution of optimal solutions S2 and S3

Table 20.1 Total mass and maximum strain values for solutions S2 and S3

Solution	Total Mass (g)	Maximum Strain ($\times 10^{-3}$)
S2	9.8	9.4
S3	15.2	4.8

Once the optimal thickness distributions for the final part (bottle) were obtained in the first phase of the optimization, the second phase comprises in find the best thickness profile of the preform, before the blowing phase, that will produce the final part (with optimal thickness profile) after blowing procedure. To compare the thickness distribution of the final part (after blowing) with an optimal thickness distribution, two objectives were defined for the second phase:

$$f_1 = \frac{1}{M}\sum_{i=1}^{M}\frac{|y_i - \hat{y}_i|}{y_i}$$

$$f_2 = \max_{1\le i\le M}\frac{|y_i - \hat{y}_i|}{y_i}$$

where $y_1, y_2, \ldots, y_M$ comprise the mean thickness value for each point along all vertical lines in the final part (after blowing) and $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_M$ comprise the mean thickness value for the corresponding points in the optimal (or target) distribution, such as in S2 or S3. Thus, f_1 is the mean error between the distributions and f_2 is the maximum error.

A preform 3D mesh model was designed to produce the same bottle model used in the first phase throughout a blow molding simulation using ANSYS Workbench software. An initial population composed by a set of ANNs that provide preform thickness profiles were randomly generated and evolved through the optimization algorithm. The same parameters (number of individuals per population, number of generations and network topology) from the first phase were used.

Two experiments (Exp1 and Exp2) were carried out: in Exp1 the optimal solution S2 (from first phase) was considered as the optimal (target) thickness profile to be reached in the final part (after the blowing procedure). In Exp2, the optimal solution S3 was considered as the optimal profile.

20.4 Results and Discussion

Figure 20.7 shows the evolution of the *hypervolume* on each generation for Exp1 and Exp2 (normalized values). It can be seen that both experiments presented higher *hypervolume* on its final populations, evidencing the evolution of each population throughout the optimization process.

Figures 20.8 and 20.9 emphasize the optimization process by showing the initial and final population for Exp1 and Exp2, respectively.

Fig. 20.7 Evolution of the *hypervolume* on each generation for Exp1 and Exp2

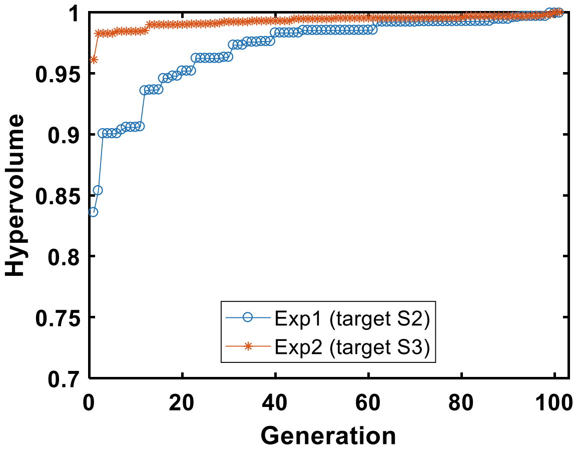


Fig. 20.8 Initial and final population for Exp 1

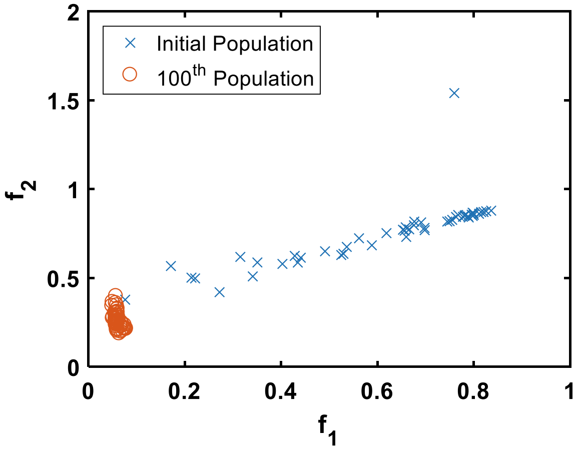
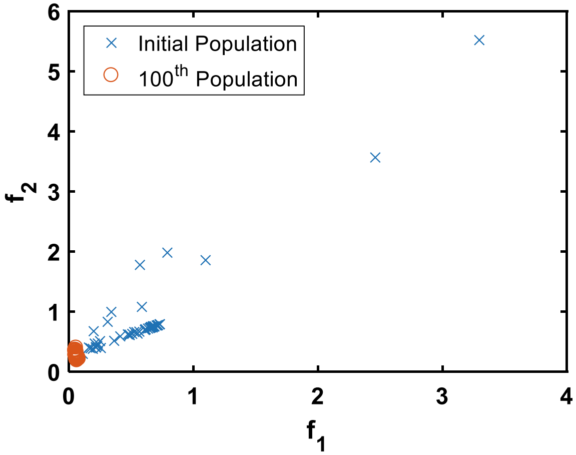


Fig. 20.9 Initial and final population for Exp 2



Figures 20.10 and 20.11 show the Pareto front (all non-dominated solutions) of final population for Exp1 and Exp2. An optimal solution was manually selected in each curve taking into account the most balanced relationship between the two objectives. In Exp1, solutions are spread across the curve while in Exp 2 solutions are concentrate between 0.055 and 0.06 on the x-axis. Also, a lower number of non-dominated solutions were found when comparing with Exp1.

Figures 20.12 and 20.13 show the thickness distribution for the optimal solutions selected from Exp 1 and Exp 2. The corresponding target distribution, i.e., the optimal distribution found in the first phase, is also presented on each graph. Table 20.2 lists the numerical values for objective functions of both solutions.

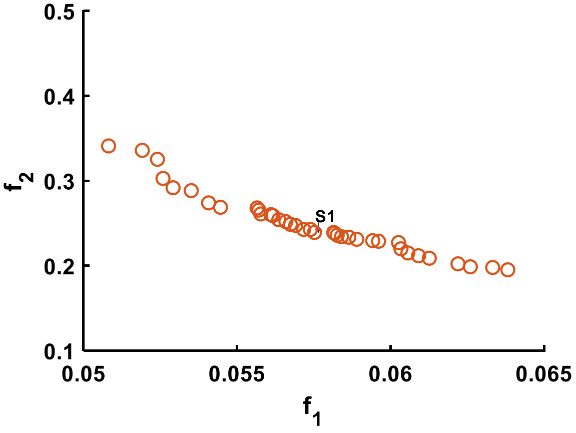


Fig. 20.10 Pareto front of Exp 1. Optimal solution S1 is highlighted

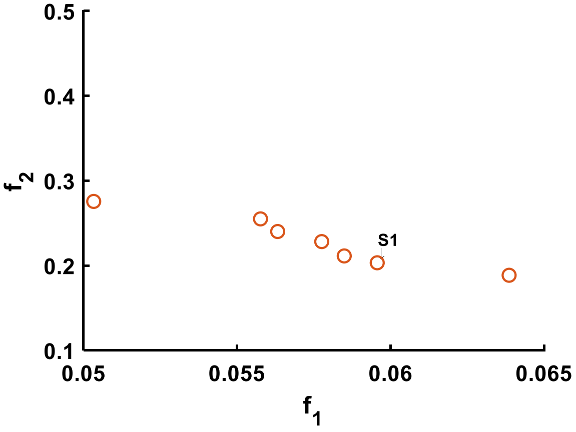


Fig. 20.11 Pareto front of Exp 2. Optimal solution S1 is highlighted

Fig. 20.12 Thickness distribution of solution S1 from Exp1 (target distribution is S2 from the first phase)

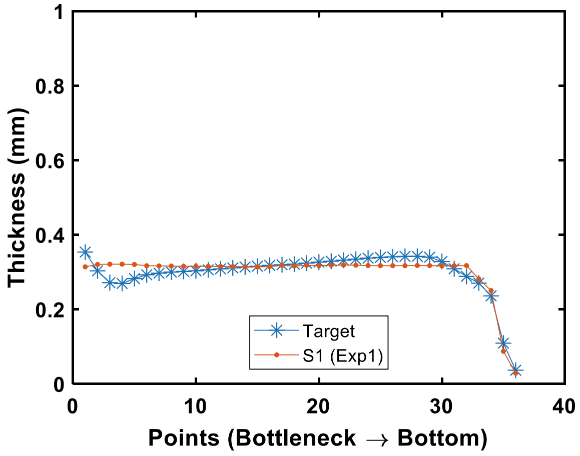


Fig. 20.13 Thickness distribution of solution S1 from Exp2 (target distribution is S3 from the first phase)

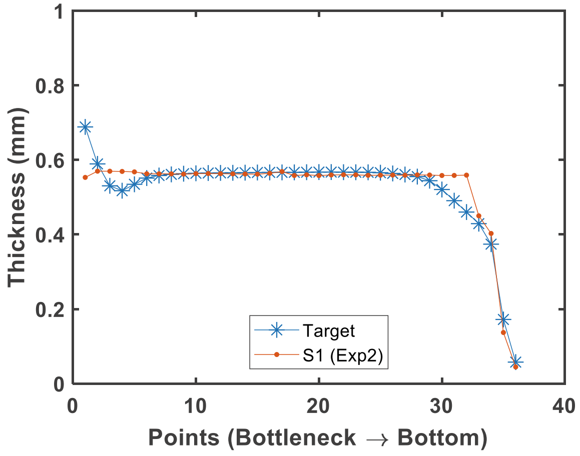


Table 20.2 Objective function values for optimal solutions selected from Exp1 and Exp2

Solution	f_1	f_2
S1—Exp1	0.0575	0.0596
S1—Exp2	0.2386	0.2026

Both solutions presented mean error (f_1) of order 0.05. Concerning the precision generally involved in the manufacturing process, this error is irrelevant, which means that the thickness profiles found for the preform will produce the final bottle with the desired thickness distribution after the blowing process.

Concerning the maximum error (f_2), the values obtained were 0.2386 and 0.2026 for Exp1 and Exp2, respectively. Although it represents around 20% of error, it is important to point out that f_2 is a single point with the highest divergence between the resulted and the target thickness. For a mean thickness distribution of 0.3 mm, 20% represents 0.06 mm, which is also very small concerning the manufacturing process.

20.5 Conclusions

Injection stretch blow molding is a process widely used by the industry to produce hollow plastic parts. The optimization of this process can heavily decrease production costs by finding thickness profiles which give the best tradeoffs between different objectives, such as material utilization and mechanical properties. Previous studies had proposed a neuroevolutionary multiobjective optimization methodology for this process. The methodology is divided in four phases (or steps), but only the first step was previously covered. This study addressed the second phase of the methodology, which corresponds to the optimization of the blowing phase in the manufacturing process.

Using the optimal thickness profile of the final part found in the first phase of the optimization process, the second phase performed the optimization with the goal to find the preform thickness profile that produces the final part with the optimal profile after the blowing process. Two optimal profiles from the first phase were considered and the same procedures for optimization were followed, defining the appropriate objective functions and FEM models for the current phase. The results comprise a set of solutions that provide different thickness distributions for the preform (through ANNs) that will produce the final part with the desired optimal profile with a mean error of 5%, which is irrelevant considering the precision of manufacturing process. Two optimal preform thickness profiles were selected from the Pareto front and can be used in the next phase of the optimization process.

Future works should address the other steps of the optimization methodology. The next phase of the optimization should consider the optimal preform profiles found on this study.

Acknowledgments This work has been supported by the European project MSCA-RISE-2015, NEWEX, with reference 734205.

References

1. Denysiuk R, Gonçalves N, Pinto RS, Silva H, Duarte FM, Nunes JP, Gaspar-Cunha A (2019) Optimization of injection stretch blow molding: Part I - Defining part thickness profile. Int Poly Process XXXIV
2. Pinto R, Silva H, Duarte F, Nunes J, Gaspar-Cunha A (2019) neuroevolutionary multiobjective methodology for the optimization of the injection blow molding process. In: International conference on evolutionary multi-criterion optimization

3. Denysiuk R, Duarte FM, Nunes JP, Gaspar-Cunha A (2017) Evolving neural networks to optimize material usage in blow molded containers. EUROGEN - International conference on evolutionary and deterministic methods for design optimization and control with applications to industrial and societal problems
4. Biglione J, Béreaux Y, Charmeau J-Y, Balcaen J, Chhay S (2016) Numerical simulation and optimization of the injection blow molding of polypropylene bottles-a single stage process. Int J Mater Form 9:471–487
5. Hopmann C, Rasche S, Windeck C (2015) Simulative design and process optimization of the two-stage stretch-blow molding process. In: AIP Conference proceedings
6. Huang G-Q, Huang H-X (2007) Optimizing parison thickness for extrusion blow molding by hybrid method. J Mater Process Technol 182:512–518
7. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: Multiobjective selection based on dominated hypervolume. Eur J Oper Res 181:1653–1669
8. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: international conference on parallel problem solving from nature

Chapter 21
Simulation of Vacuum Assisted Resin Infusion (VARI) Process for the Production of Composite Material Parts

Joana M. Malheiro and J. P. Nunes

Abstract The Vacuum Assisted Resin Infusion (VARI) is manufacturing process used worldwide to produce composite parts having great diversity of dimensions (from small to very large ones) and geometrical complexity. This manufacturing process is particularly versatile, to produce small series of high performance structural parts. In these cases, the simulations of the VARI process is a very useful tool to define the infusion strategy and to plan and predict the resin flow progress in order to reduce the material waste and manufacturing cycle time and obtaining lighter structures, having lower void fraction and higher fibre content and mechanical performance. The numerical simulation of the VARI process implies the modelling of different complex phenomena, such as flow in porous media, mechanical deformation, heat exchange and chemical reaction. Therefore, a finite element software was used to solve a combination of governing equations based on a combination of pre-defined theoretical assumptions, by considering a moving mesh and appropriated boundary conditions. In this work, results obtained from simulations of VARI process were used to define the best strategy to be applied in the production of composite parts with different geometries, sizes and materials and predict the possibility of defects occur. In order to validate the accuracy of simulations, the numerical results were compared with those experimental ones obtained from the production of different composite parts where the best processing strategies were implemented. After analysing and discussing the theoretical and experimental obtained results, changes were applied to the numerical model to improve simulation accuracy.

J. M. Malheiro (✉)
Composite Research Group, PIEP – Innovation in Polymer Engineering, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: joana.malheiro@piep.pt

J. P. Nunes
Institute of Polymers and Composites, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: jpn@dep.uminho.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_21

Keywords Composite materials · VARI · Composite material manufacturing simulation · Finite element · Porous medium flow · Material characterization

21.1 Introduction

The increase of production rates and from quality of manufactured parts is implying a growing interest by composite manufacturing process simulation and by its stronger predictive accuracy [1]. Among the industrial manufacturing processes used, resin infusion appears more and more as the best economical alternative to manufacture large and/or high mechanical performance parts, with large fibre fractions (wind turbine blades, structural components, wings of aircrafts, etc.). Composite manufacturing processes by resin infusion have been developed for years to bring a cheaper solution to big parts production. Those processes allow a significant cost reduction in raw materials storage and mould fabrication, shorter cycle times, less void formation and avoid the use of trial and error time-consuming procedures to optimise the process parameters [3]. In recent years, the interest in using out-of-autoclave (OOA) processing techniques, such as resin infusion ones, has also been renewed because of the very expensive initial investment required by the traditional methods in autoclave, particularly when the composite parts and structures to be produce become larger and more complex.

Furthermore, the possibility of simulating these liquid resin infusion processes allows to predict potential defects on the final parts and reduce significantly the time for setting and optimising their processing parameters [2]. In fact, without using computer simulations, the success of these resin infusion methods become highly dependent upon operator skill and experience, particularly in the development of new manufacturing strategies for complex parts. Process modelling, as a predictive computational tool, aims to address and improving the reliability and waste issues that usually result from traditional trial-and-error approaches. Basic modelling attempts generally focus on simulating fluid flow through an isotropic porous reinforcement material. Currently, the more recently developed advanced numerical algorithms are also beginning to take into account the multiscale and multidisciplinary complexity of the reinforcement materials, increasing the accuracy of the simulations [4]. In the case of resin infusion manufacturing with textile reinforcement materials, the physical draping of the fabric and the subsequent resin flow through the material are the key stages of the process [4].

In this paper, the accuracy of the results obtained by the simulation of Vacuum Assisted Resin Infusion (VARI) manufacturing process is assessed. To accomplish that, the simulation of the infusion of composite parts with different geometries (from a planar plate to a hull of a boat) was performed and compared to the experimental results, obtained from the actual production of the same components. The main limitations of the numerical model are pointed out as well as the considerations and assumptions necessary to obtain more accurate numerical results.

21.2 Vacuum Assisted Resin Infusion (VARI)

The Liquid Composite Moulding (LCM) is a generic term for a family of related processes in composites manufacturing, in which continuous fibres used as reinforcement are first placed in a mould, then a polymer matrix (usually thermosetting) is injected as liquid resin in the cavity [5]. The Vacuum Assisted Resin Infusion (VARI), in particular, consists in impregnating a dry preform placed onto a rigid half-mould and under a vacuum bag [1]. Then, the pressure differential between a vent pipe connected to a vacuum pump (aprox. at 0 bar) and the injection line (at atmospheric pressure) induces the infusion of the resin along and across the reinforcement. After complete impregnation, the part is subjected to a curing process, usually out of autoclave. For the infusion, several injection ports, injection lines or a tree of injection channels can be used. It is necessary to select a good configuration of injection ports and vents to avoid dry spots and minimize filling time. The VARI process eliminates the costs associated with matched metal tooling, reduces volatiles emission and allows the use of lower resin injection pressures. Also enables the use of low-cost tooling while still producing high quality composite parts with complex geometries [6]. Despite the versatility of the VARI process, the reliability and repeatability issues still is, however, a concern for the widespread adoption of this manufacturing process by the advanced industry, when well-validated simulations are not used. The efforts to simulate the resin infusion manufacturing process aim to address the reliability and repeatability concerns in a cost-effective manner [4]. Numerical simulations of mould filling can be of great help to avoid problems such as resin rich areas, air bubbles, dry spots, zones of high porosity, as well as the formation of cracks following cure shrinkage. It is also advantageous to determine the optimal infusion strategy [5, 6].

21.3 Governing Equations

The resin infusion process is particularly complex to model. In general, the manufacturing process is divided into four main phenomena [4, 5]: the physical accommodation of the reinforcement material lay-up to the mould (draping); the flow of the resin through the reinforcement material (infusion); the exothermal reaction of the resin (curing), consequent thermal analysis of heat exchange between the part, mould and environment and the influence of all these factors on the resin viscosity [4].

Up until a few years ago, many flow models that were still used by industry lacked enough precision, because they relied on the assumptions of a homogenous, continuum-based approximation of the preform domain and neglect through-thickness effects, saturation, compaction, and heat transfer. Recently, finite-element based methods have been developed with increasing sophistication, to take into account not only the interdependence of the different phenomena that influences the infusion process but also almost all factors that affect resin flow behaviour. These

Table 21.1 Governing phonomena and mathematical models used in infusion process simulations [5]

	Phenomena	Mathematical model
Rheological	Resin flow in a porous medium Variation of viscosity	Darcy's law Constitutive law
Thermal	Mould: conduction, loss in surface Part: conduction, convection, generation of heat, superficial heat loss	Heat equation, transfer coefficient (convection-radiance) Equation of convection-diffusion with source term, model with one temperature
Chemical	Transport of chemical species, diffusion, polymerization	Equation of convection-diffusion with source term, kinetic model (Kamal-Sourour)
Mechanical	Mould deformation	Newtonian's law Empirical models

last factors are: permeability, pressure, viscosity, temperature and heat exchange, variability and susceptibility to handling and cutting of reinforcement, presence of passive apparatus (such as inlets, outlets, flow enhancing materials, etc.), through-thickness effects (effect negligible in thin composite parts), deformation-dependent permeability properties, saturation, tool compaction (because the process uses a flexible film semi-tooling, which deforms simultaneously under the internal mould depression and in result of resin infiltration), void formation, among others [5].

The main phenomena and respective mathematical models usually considered in infusion process simulations are summarized in Table 21.1.

21.3.1 Flow in Porous Media

In the VARI process, the resin flows through a fibrous reinforcement, which can be considered as a porous medium. In this case, the flow of resin is governed by Darcy's Law, which states that the flow rate of resin per unit area is proportional to the pressure gradient and inversely proportional to the viscosity of the resin. The constant of proportionality is the permeability of the porous medium. It is independent from the fluid, but it depends on the direction of the fibres in each layer of reinforcement. Also capillary forces of attraction or repulsion, which depend on the resin surface tension and its ability to adhere to the surface of fibres and that may also affecting the forehead of flow, by either reducing or increasing the effective pressure at the resin front. However, these latter effects are generally considered too small and, therefore, neglected by almost all numerical models. So, assuming that the resin is an incompressible fluid (generalized Newtonian fluid) that travels at low velocity trough a porous medium and the permeability of the porous media is 10^{-3} m^2 or less, the Darcy's Law may be written as [4, 7]:

$$\vec{V} = -\frac{K}{\mu}\vec{\nabla}P \tag{21.1}$$

where, K is the permeability tensor, μ is the viscosity of the resin, V is the Darcy's velocity and P is the pressure (overall pressure gradient through the system) [4].

The permeability characterizes the relative facility that a viscous liquid has in flowing through a porous medium in order to impregnate it. This physical property of the porous medium (cloth, fabric, fibre mat, etc.) depends on the fibre volume fraction (degree of compaction), orientation and configuration of fibres and draping of plies. The permeability of the reinforcement in their principal directions may be determined experimentally.

21.3.2 Draping (Mechanical Properties)

From a mechanical perspective, draping behaviour has proven to be difficult to replicate accurately. Woven warp and weft yarns exhibit considerable tensile strength and stiffness but are highly susceptible to reorientation under shear and bending modes. Therefore, any attempt to model draping must accurately account for the yarn reorientation that result from shear loading [4]. In order to replicate the mechanical behaviour the characterization of the reinforcement tensile, shear, and bending properties is mandatory.

21.3.3 Thermal Phenomena

The final impregnated part that will lie in the cavity of the mould, consists of reinforcements and resin, which first fills the mould and then becomes progressively polymerized. Heat transfer phenomena significantly affect mould filling and resin curing. Indeed, the temperature of the resin governs the reactivity of the polymerization reaction. Temperature also has an influence on mould filling, since the viscosity of the resin depends on temperature. Thermal simulation are therefore delicate to conduct because of all the related phenomena. Firstly, heat is transferred by conduction between the fibres and the resin. Secondly, a convective transport of heat occurs during the filling of the cavity by the resin. Finally, heat is produced by the exothermic chemical reaction of resin polymerization. Some heat is also created by the viscous dissipation during the resin flow, but in lower degree than the heat originated by the chemical reaction of cure. The temperature field is governed by the general equation:

$$\rho C_p \frac{\partial T}{\partial t} + \rho_r c_{pr} \vec{V} \cdot \nabla T = \vec{\nabla} \cdot \{k \cdot \nabla T\} - p_r \Delta h \frac{D\alpha}{Dt} \tag{21.2}$$

where T denotes temperature, t is the time, ρ is the density, C_p is the specific heat, k is the heat conduction coefficient tensor, the subscript r designates the resin, Δh is the total enthalpy of the cure reaction of the resin, α is the degree of resin cure conversion. This general equation enables to treat the steps of pre-heating, filling and curing.

21.3.4 Viscosity of the Resin

The viscosity of the resin depends highly on the temperature and degree of cure conversion, by assuming that viscosity will be infinite when the resin reaches gelation. The dependence of viscosity on these factors can be modelled by a range of different assumption and respective constitutive laws, such as: constant viscosity (Newtonian fluid); predefined law considering the viscosity dependence on temperature; predefined law considering the viscosity as function of temperature and resin curing rate; predefined law considering the viscosity as a function of temperature and resin strain and curing rate.

21.3.5 Kinetics of Resin Polymerization

The kinetics of polymerization of the resin is usually simulated by the model of Kamal-Sourour, and is essential to describe the curing reaction of the resin [5]. In this study, the effect of the resin polymerization will be neglected because the gel time of the resin is assumed to be sufficiently long for not affecting the resin viscosity, which is made constant, and that the curing reaction will take place long time after the infusion process is finished.

21.4 Numerical Method

In this study, the PAM-RTM® finite element software from ESI was used to simulate the infusion process. It is based on the coupling between the resin flow, governed by Darcy's law, and the preform behaviour considered as porous medium undergoing deformations accordingly to the Terzaghi's principle. The numerical algorithm also considers the changing thickness of the laminate and compaction as a function of the fibre volume content during the infusion [1, 5, 7]. For that, the software decomposes space and time, being the system divided in three zones in space [2]: Stokes zone (fast flow zone constituted by the distribution medium and the resin); Darcy zone (incompressible flow of the resin in the preforms submitted to finite deformations); and dry preforms zone (zone constituted of non-impregnated preforms submitted to finite strains). On the other hand, time is divided in four periods that correspond to the following changes in boundary conditions or physical problem [2]: pre-filling (initial compaction of the preforms due to the vacuuming of the system); filling; post-filling (re-compaction or "rest period" ending by the mechanical equilibrium mandatory to the dimensional quality of the final part); and curing. The model also take into account the porous medium deformation during the temperature and pressure cycles, and deals with the influence of the preform deformation on permeability, and therefore on pressure distribution. Moreover, a thermo-chemical model describes viscosity changes during the infusion [7]. More details of the algorithm used can be found in the work of Celle et al. [7] and Dereims et al. [8]. The software allows Dirichlet or Neuman boundary conditions, and takes into account the effect of gravity, which is important in large structures and negligible in small parts [5].

21.5 Results and Discussions

The results of the infusion process to manufacture different parts in composite materials are presented below, where the simulation results are compared to the experimental results. For simulations, it was necessary to characterize properly both resin and reinforcement materials.

The permeability of the reinforcement materials, and its variation of the combining effects of orientation and configuration of fibres, draping of plies, compression, etc., is difficult to measure accurately, but its determination is paramount in the simulation of the VARI process. To overcome this problem, a methodology to determine and validate numerically this parameter is presented. After numerically validate the experimental parameters, the simulation of different composite parts are performed and experimentally verified. To accomplish that, simple geometries with simple laminates are firstly validated, then the same is done for geometries and laminates increasingly complex. In the process different assumptions and simplifications are admitted without compromising the accuracy of the numerical results.

21.5.1 Flat Square Plates

A Brookfield viscometer was used to measure the viscosity of a the polyester resin Distriton 3501S with 1.5% of hardener (NOROX MCP) along time. Table 21.2 summarises the results obtained from those tests. The resin behaves as a Newtonian fluid, with constant viscosity of 469 mPa. s, and allows, approximately, 150 min of working time. The long gel time allows to perform the infusion without significant variations in viscosity, and ensuring that the curing process takes place after the complete impregnation of the laminate.

The reinforcement properties are presented in Table 21.3. A glass fibre unidirectional stitched fabric was used as reinforcement of a laminate of 300 × 300 mm, which had only one reinforced layer. The permeability along the two main directions

Table 21.2 Resin Properties—Polyester resin Distitron 3501 S

Density	Viscosity	Gel Time	Curing time	Exothermic Peak
(g/cm^3)	(mPa . s)	(min)	(min)	(°C)
1.12	469	≈150	22–32	140–170
Curing Cycle				
24 h a 23 °C + 2 h a 100 °C + 1 h a 100 °C				

Table 21.3 Reinforcement Properties–Glass fibre fabric

Density (kg/m^3)	Structure	Weight (g/m^2)	Orientation (°)	Thickness (mm)
2600	Unidirectional, stitched	300	0°	1.0

of the glass fibre fabric is, respectively, $K_1 = 1.090\text{E}{-}08\ \text{m}^2$ and $K_2 = 1.250\text{E}{-}10\ \text{m}^2$, assuming that gravity and thickness of the laminate have no effect.

Figures 21.1 and 21.2 compare, at the same moment in time, the experimental and numerical results obtained when two different types of arrangements were used for the resin inlet and outlet (with and without runners). As may be seen, good agreement between the numerical and experimental results is observed. As Fig. 21.3 shows, the resin flow front and the filling time depends on the type of resin inlet. As this last figure depicts, the filling time is lower when runners are used as resin inlet (t = 1515 s) than without runners (t = 6520 s), while experimentally the infusion took the similar values of, approximately, 1476 and 6180 s, respectively. For both situations, simulations predicted that a volume of resin of approximately 4.60E-05 m^3 will be used in the infusion, while experimentally a volume around 5.00E-05 m^3 was used.

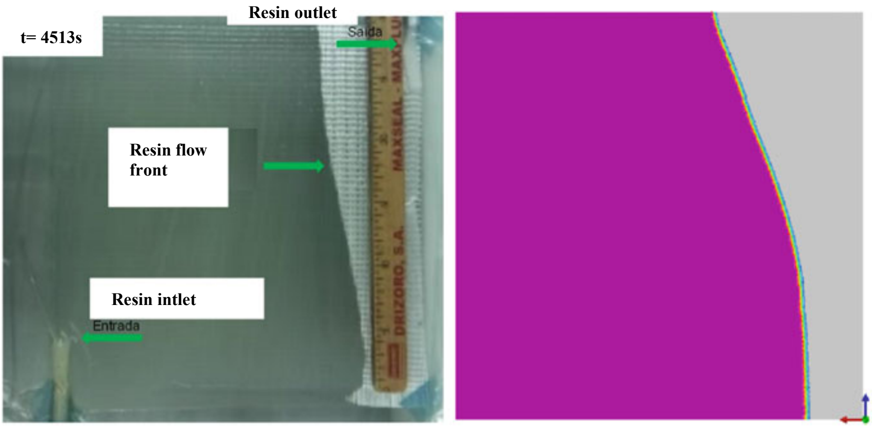


Fig. 21.1 Infusion of a plane laminate without runners at t = 4513 s: experimental (left); simulation (right)

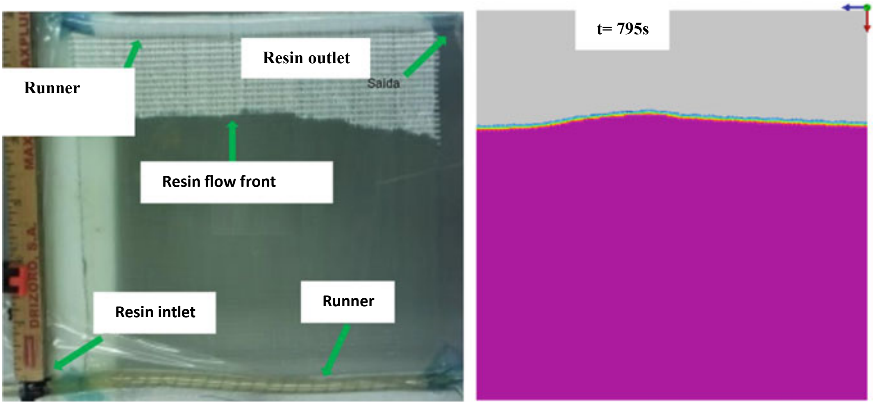


Fig. 21.2 Infusion of a plane laminate with runners at t = 795 s: experimental (left); simulation (right)

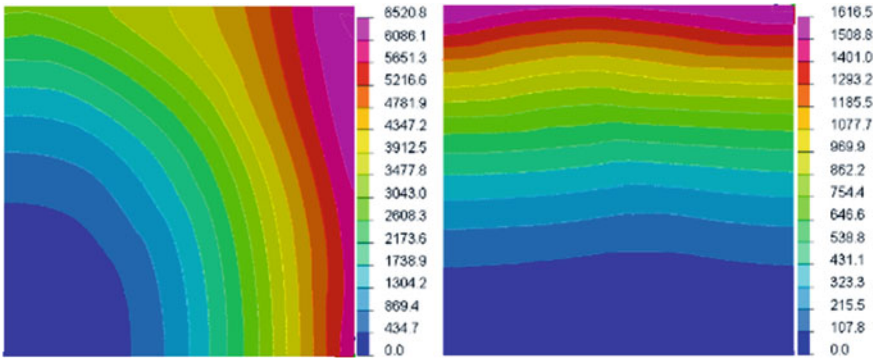


Fig. 21.3 Filling time for infusion with (right) an without (left) runners

21.5.2 Guitar Plate

The same kind of study was performed for a plate with a complex geometry, namely, a pickguard of a guitar. In this case, the laminate had five layers of carbon fibre fabric with the following properties: plain (0°/90°); weight per unit area: 195 g/m^2; density: 1770 kg/m^3; thickness: 0.30 mm. The single layer of reinforcement presents the following permeability along its main directions: K_1= K_2= 8.304E-11 m^2. As matrix was used a polyester resin, with the following properties: viscosity: 0.300 Pa. s; density: 1200 kg/m^3. The properties of the resin were considered constant in time.

Although, this part is actually a flat plate, it presents a complex boundary geometry, with different curves and cuts. These characteristics will affect the resin flow front during impregnation because resin flows faster along the reinforcement boundaries. This was observed independently of the type of arrangement used for the resin inlet

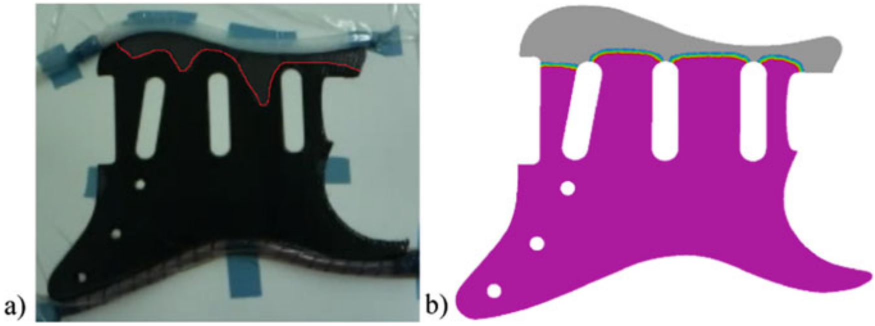


Fig. 21.4 Flow front at t = 2040 s: experimental (**a**), simulation (**b**)

and outlet, and partially explains the difference between the experimental and numerical results (Figs. 21.4 and 21.5 a, b). Thus, by changing the model and assuming that the permeability was higher at the reinforcement boundaries, both experimental and numerical results start to approximate as it may be seen in Fig. 21.5a, c. Still, differences between numerical and experimental resin front outline obtained (Fig. 21.5) were significant. This is easily explained by the permeability admitted in the simulation, which was determined experimentally for one single layer (following the same methodology of the previous example) while the laminate is a stack of five layers, i.e., effects, for example, of draping and compression were not taken into account in the global permeability of the laminate.

21.5.3 Car Seat Part

The assumptions admitted previously were applied in the production of a component from a car seat (Fig. 21.7a). The laminate was manufactured by using the same carbon fibre fabric, but twice as many layers (10 layers) than those used in the guitar plate. Since a different laminate is used, it will present necessarily a distinct permeability. Knowing, from the previous example, that the software does not allow to get an accurate simulation of the real resin flow front advance by using the permeability of the single elemental layer used, the permeability of the ten layer laminate was determined experimentally and validated numerically as previously described in point 5.1 and Fig. 21.2. In order to quantify the new permeability, the validation of the infusion of the laminate was performed (Fig. 21.6), using an epoxy resin (density:1140 kg/m^3) which behaves as a Newtonian fluid during the infusion, with constant viscosity equal to 0.170 Pa.s. A permeability of K_1= K_2= 6.827E-12 m^2 was determined for the laminate by using this procedure (Fig. 21.6a). Such permeability is considered as the global permeability of the laminate, which means the effect of compressibility and draping between layers, that are difficult to quantify and mimic in the simulation, are accounted for. Thus, the following simplifications

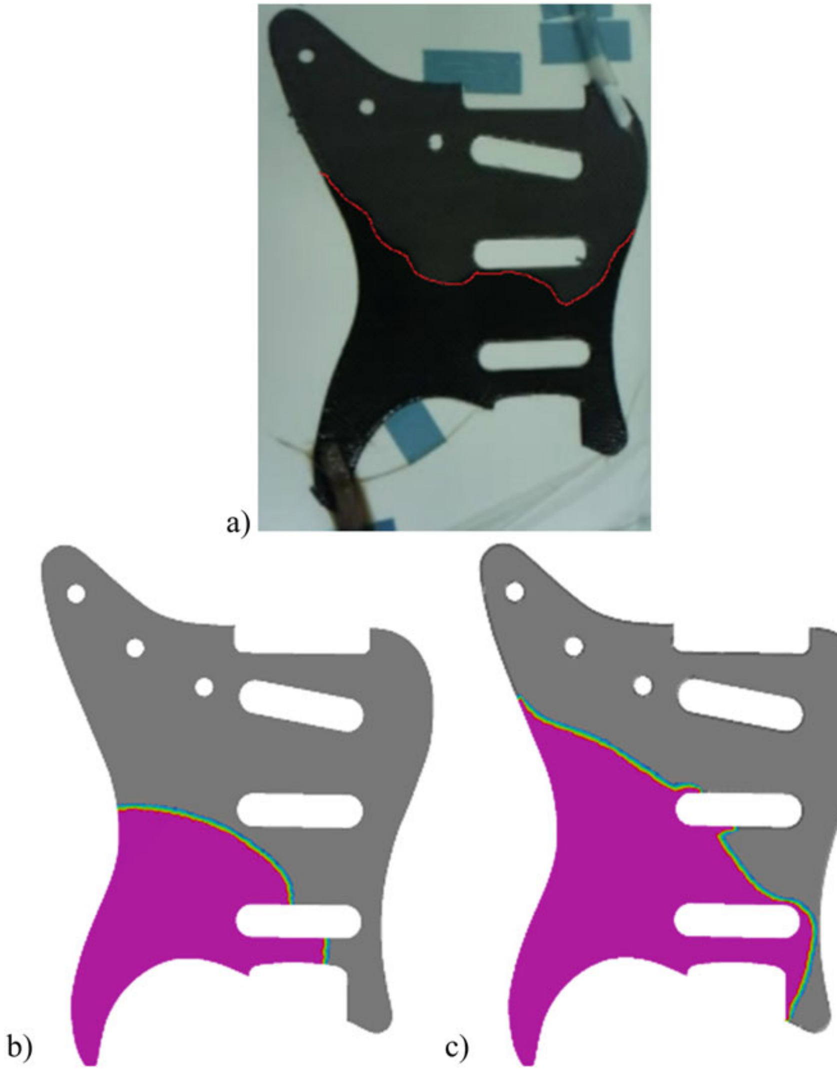


Fig. 21.5 Resin flow front at t = 1840 s: experimental (**a**), simulation (**b**), simulation with different *K* at boundaries (**c**)

were assumed to validate the permeability in the simulation: (i) the geometry of the laminate is a single layer (surface) of 2D triangular elements and, (ii) it presents the global permeability determined for whole 10-layer laminate. As Fig. 21.6 shows, a good agreement was found between the experimental and numerical results: the infusion took ≈1700 s experimentally while the simulation predicted 1783 s; at ≈ 883 s the resin flow front advanced approximately the same distance (≈330 mm);

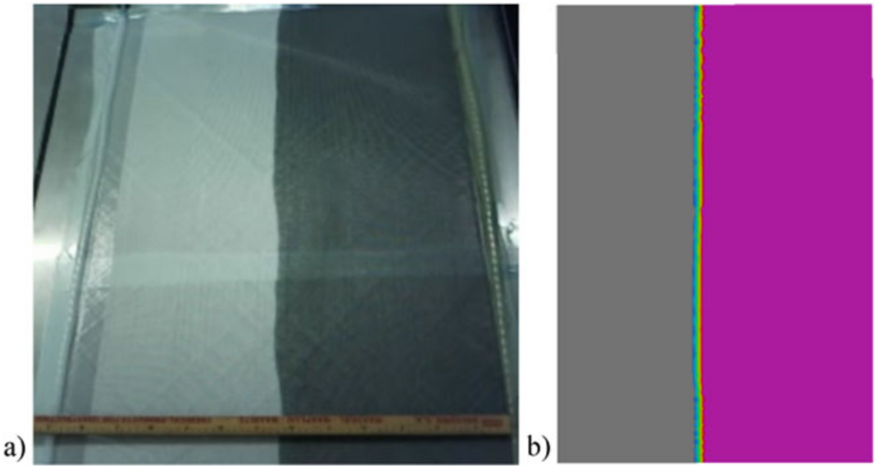


Fig. 21.6 Resin front: experimental (**a**) and simulation (**b**) at t = 883 s

and the quantity of resin estimated in the simulation was 723 g, while experimentally ≈1520 g of resin were necessary, due to passive accessories, for a fibre volume fraction of 50%.

After validation of laminate properties, the data obtained were used to simulate the vacuum assisted infusion of a component of a car seat (Fig. 21.7a). The simulation results are presented in Fig. 21.7. During infusion, the resin flows from the region of maximum pressure, at the entry runner, (1 bar, Fig. 21.7b) toward the region where pressure is minimum (0 bar, Fig. 21.7b). The distribution of velocity (Fig. 21.7d) shows that the flow is faster at the concave corners of the geometry than in plane regions and convex corners, as expected and observed experimentally, due to the formation of channels in these regions as a consequence of the reinforcement draping on the geometry. The selected entry and exit ports leaded to a steady progression of the resin flow front along the laminate, and the total impregnation of the laminate is observed, taking 2133.3 s and ≈2220 s to be numerically (Fig. 21.8c) and experimentally completed, respectively. The good agreement found between the simulations and experimental results shows that the adjustments applied, so far, to the numerical model resulted in a very realistic representation of the infusion of composite parts.

21.5.4 Sandwich Laminates

Sandwich laminates are used to produce composites requiring both lightweight and high mechanical performances. In general, they are constituted by a core protected by a skin on, at least, two sides of its structure (Fig. 21.8). The adhesion between core and skin is, usually, achieved by the matrix resin, which impregnate both skin

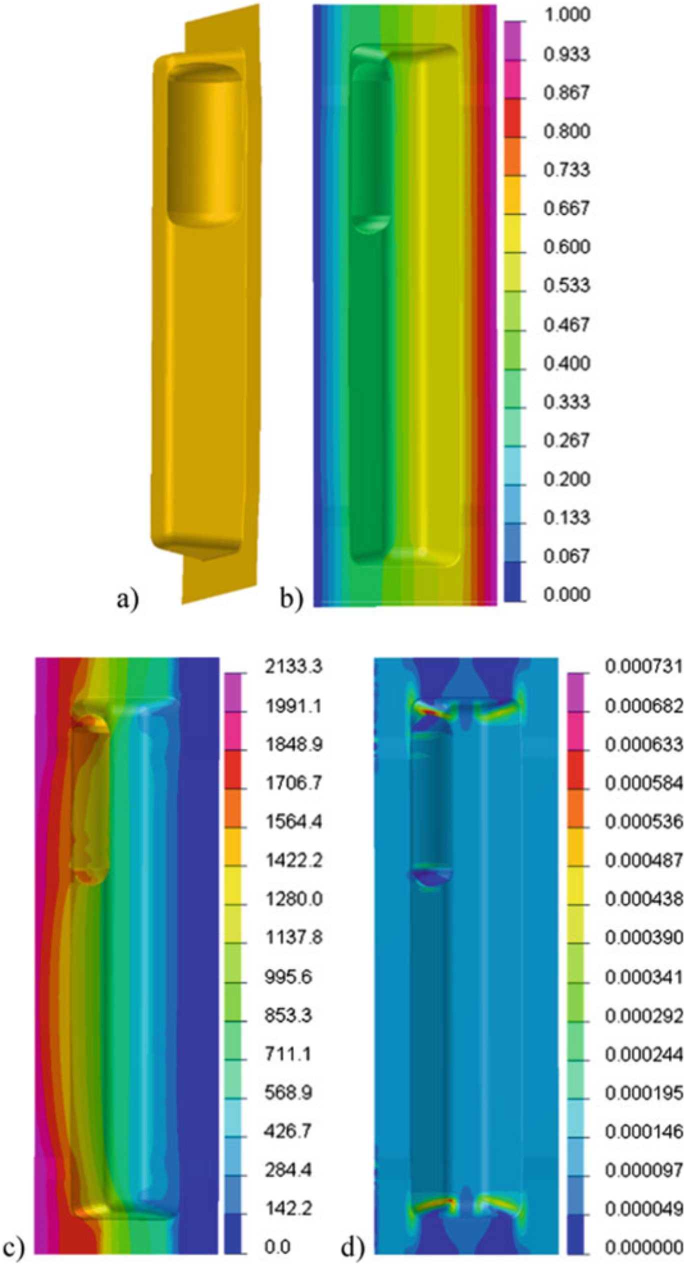


Fig. 21.7 Simulation results of the infusion process of the part of a car seat (**a**): pressure distribution [bar] (**b**), filling time [s] (**c**), flow velocity [m/s] (**d**)

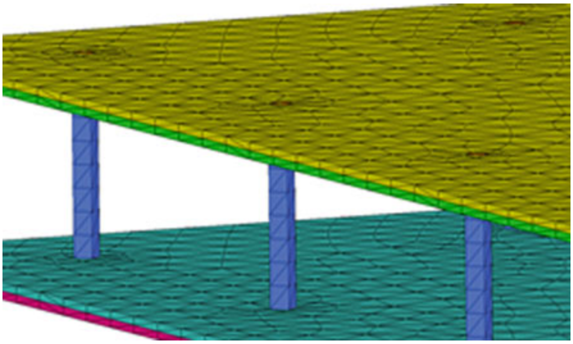


Fig. 21.8 Sandwich laminate structure and mesh

and core together. However, in order to get a lightweight composite, the core should not absorb resin within its structure. The adhesion is obtained by particular mechanical finishing, such as, perforations, grooves, grid-scores, etc., which guarantee the desired mechanical adhesion (generally designated as biding points).

Thus, along with the simplifications described in the examples above, to simulate the VARI process of a sandwich laminate other assumptions were considered. By way of example, the sandwich laminate depicted in Fig. 21.8 is considered: it uses a perforated PVC foam (core), two plies of biaxial glass fibre fabric on each side (skins) and a polyester resin. The properties of the reinforcements and resin are summarized in Tables 21.4 and 21.2, respectively.

The main simplifications used to build the mesh were: (i) the volume of resin deposited in the surface of the core was neglected and, (ii) it was assumed that only the binding points are filled with resin. Thus, instead of building a mesh throughout the volume of the core it was built only at the binding points (in the example, perforations), which guarantee the connection between the core and skins as Fig. 21.8 illustrates. It is admitted that the perforations of the core have maximum permeability in all directions ($K_1 = K_2 = 1.257\text{E-05 m}^2$), while the biaxial fabric has $K_1 = K_2 = 9.913\text{E-11 m}^2$, which was determined experimentally as previously described in point 5.1 and Fig. 21.2 by considering a laminate with two biaxial plies.

Table 21.4 Sandwich Laminate—Reinforcement Properties

Reinforcements		
Material	Glass fibre	PVC
Density (kg/m^3)	2600	100
Structure	Biaxial, stitched	Foam, perforated
Weight (g/m^2)	300/300 (612)	–
Orientation (°)	0°/90°	Random
Thickness (mm)	1.00	10.0

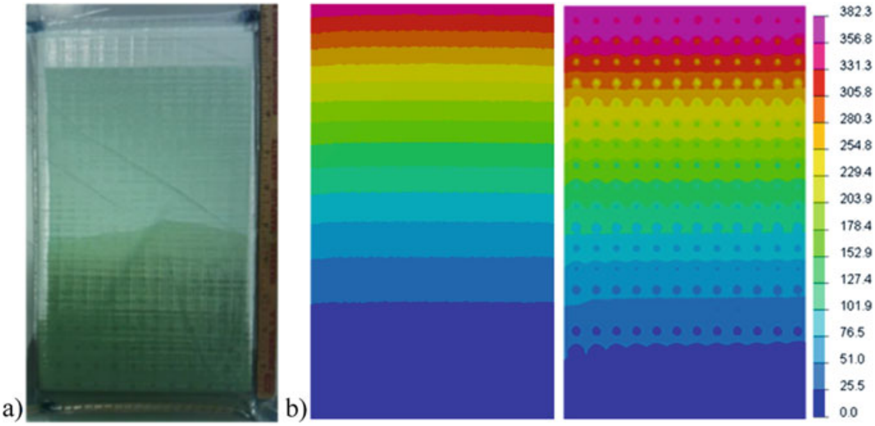


Fig. 21.9 Experimental results: resin front at t = 60 s (**a**); and numerical results: filling time (**b**)

Following the same previously mentioned method, the material characterization was validated by comparing the numerical and experimental results (Fig. 21.9). Illustrated in Fig. 21.9 is the numerical filling time (of both skins) and the experimental advance of the resin front, at the top skin, in the sandwich laminate at t = 60 s, where good agreement is observed between numerical and experimental results: in both cases, the resin front travelled a distance of ≈245 mm at the top skin, but at the bottom skin the resin front had a delay of ≈50 mm; while the actual infusion time was ≈405 s, numerically was 382 s; and advance of the resin front in the sandwich structure registered was very similar.

21.5.5 Hull of a Boat

In the production of composite parts by VARI, in addition to an accurate material characterization, a good definition of the infusion strategy (distribution of resin ports of entry and exit) is mandatory, especially in complex geometries. It is in the definition of the infusion strategy that the simulation of the infusion process has a significant role. Thus, as an example, the steps for the production of a hull boat prototype (Fig. 21.10), with 3 m length and 1 m width, are described.

Before simulation, some simplifications were assumed for assuring that an adequate approximation is achieved and, at the same time, considerably reducing the simulation time. The first approximation was applied to the geometry used to build the mesh: the thickness of the laminate was neglected and only surfaces were allowed. However, the properties of the material were defined according to the three-dimensional material. This simplification allows reducing considerably the number of elements (which are 2D) in the mesh.

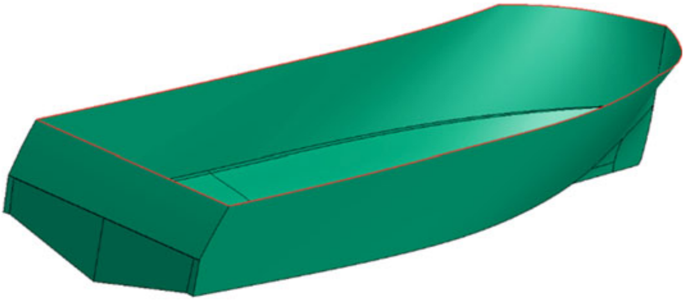


Fig. 21.10 Geometry of the hull of the vessel prototype

The matrix was considered to be a polyester resin (Table 21.2), with constant viscosity during infusion, i.e., it has a long gel time and the curing cycle begins only after the complete impregnation of the laminate. The reinforcement is constituted by a sandwich laminate similar to the one depicted in Fig. 21.8, with reinforcement properties described in Table 21.4. However, instead of a three-dimensional structure (such as the one in Fig. 21.8 and Fig. 21.9), the laminate was considered as an homogeneous material with a global permeability ($K_1= K_2=$ 3.744–09 m^2), as in Figs. 21.6 and 21.7.

After defining the materials parameters, the next step was to define the distribution lines and the entry and exit ports of resin. First, it was stablished that the flow of resin would occur from the bottom of the geometry (keel) to the top (flange) and, so a main exit line was considered along the entire length of the flange. Then, different ports of entry and distribution lines were defined, as Fig. 21.11 shows.

In general, the different infusion strategies revealed not to be adequate to the impregnation of the geometry (Fig. 21.12): regions with high probability of formation of dry zones were detected and not every laminate ended completely impregnated by the resin. Although these observations apply to all cases, significant differences were observed, depending on the infusion strategy: using resin entry ports only (Fig. 21.11c, d), the advance of the resin front is uniform and the filling of the laminate is homogeneous (Fig. 21.12c, d); using distribution lines (Fig. 21.11a, b), decreases significantly infusion time but are more likely to form dried regions, particularly at the bow (Fig. 21.12a, b); the addition of perpendicular distribution lines (Fig. 21.11e, f), results in a decrease of infusion time, but does not decrease the risk of void/dry zones formation (Fig. 21.12e, f).

Based on these results, the infusion strategy depicted in Fig. 21.14a was simulated: two distribution lines, ranging from bow to stern, located at the bottom of the hull, arranged parallel to and close to the keel. The distribution lines are opened simultaneously, resulting in the advance of the resin front depicted in Fig. 21.14. The sequence of pictures (Fig. 21.14) show that the laminate was completely impregnated, by a steady resin front. In addition, the flange of the hull is the last region to be impregnated (where the exit line is placed), indicating that the probability to occur voids and dry zones is small. Furthermore, the total infusion time was 6121 s (about

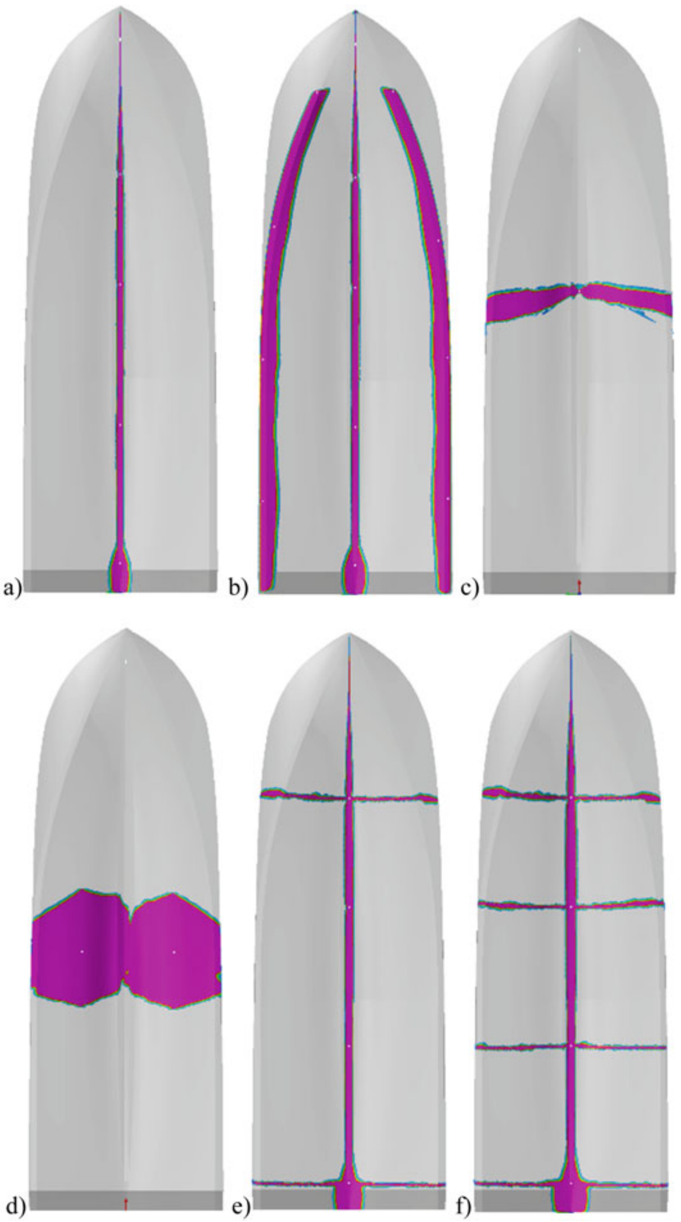


Fig. 21.11 Infusion strategies: **a** one central line along the keel; **b** one central line along the keel and laterla lines on the bottom; **c** one central entry port on the keel; **d** two entry ports on the bottom; **e** three lines: one along the keel and two perpendicular at the bow and stern; **f** similar to **e** but with two more lines at the centre

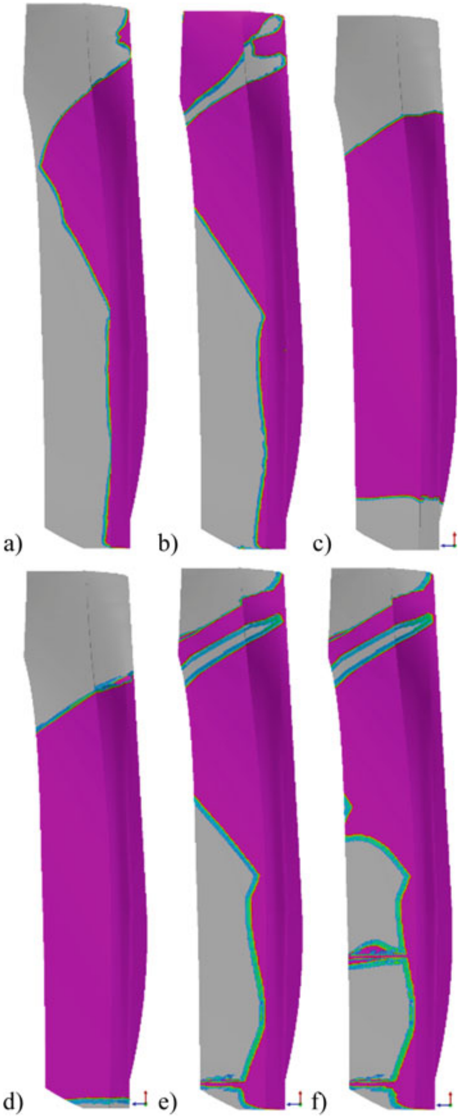


Fig. 21.12 Simulation results: resin front advance for each different infusion strategies illustrated in Fig. 21.11

102 min), which ensures that the infusion takes place within the working time of the resin (Table 21.2). In the end, the infusion strategy illustrated in Fig. 21.14 revealed to be a good strategy to be applied in the production of the hull.

Using the infusion strategy optimized (Fig. 21.14) the hull of the prototype vessel was produced (Fig. 21.14). Comparing the sequence of pictures from both Figs. 21.13 and 21.14, it may be observed that the predicted path followed by the resin front mimics accurately the resin flow front advance observed experimentally. Moreover, experimentally, the infusion took about 90 min. This difference is justified by the

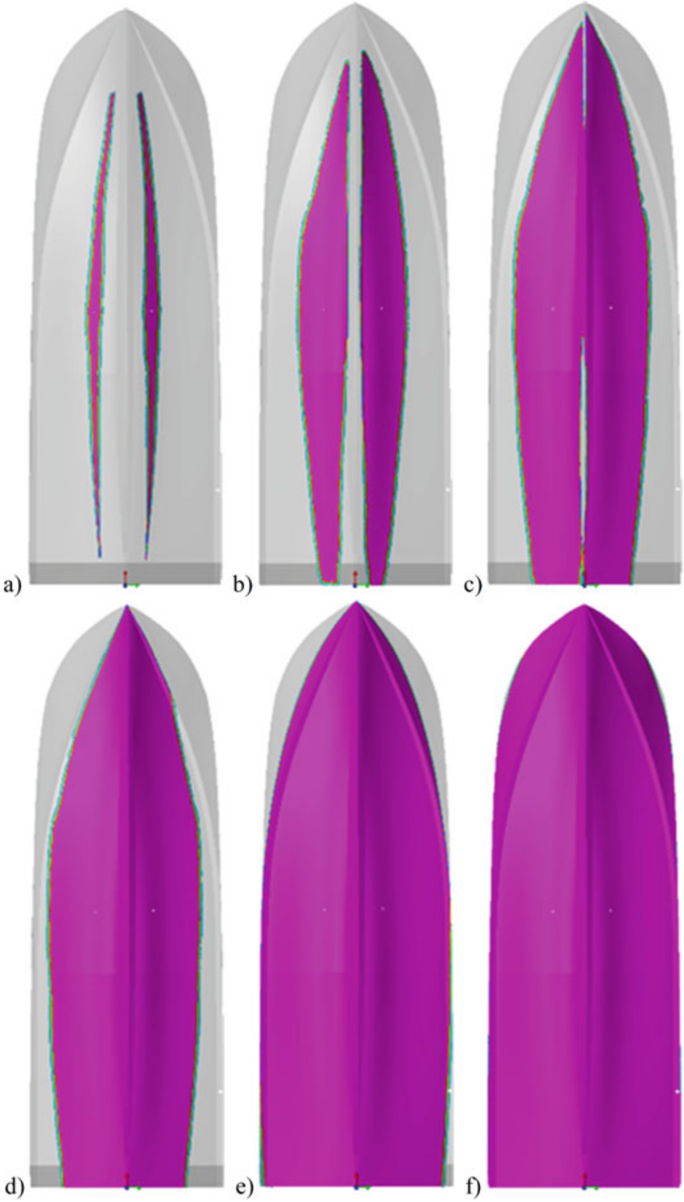


Fig. 21.13 Optimized infusion strategy: resin front advance (filling factor)

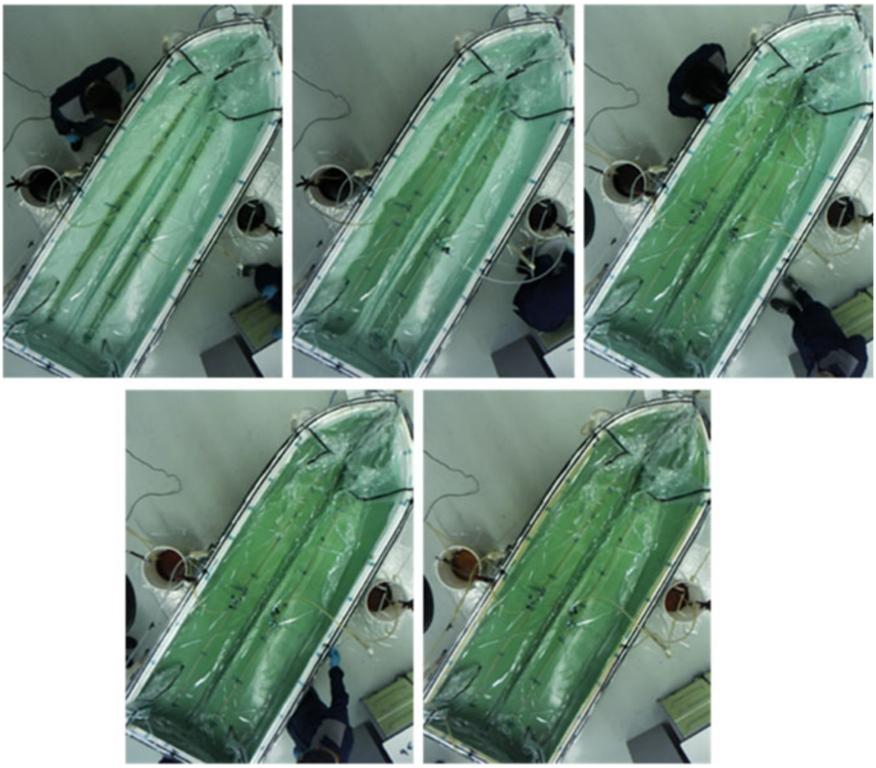


Fig. 21.14 Production of the hull using the infusion strategy illustrated in Fig. 21.13

formation of wrinkles, channels and bridges, in the assembly of the laminate and in the sealing of the vacuum bag, where the resin moves at a higher speed, compared to the displacement velocity in the laminate. These effects are not considered in the simulation where the laminate arrangement is perfect and the resin only moves faster along boundaries (as in the guitar pickguard) and corners (as in the part of a car seat). Despite the differences, the simulation results reveled to be accurate.

21.6 Conclusions

The simulation program PAM-RTM® from ESI was used in this work to simulate the production of composite parts by using the Vacuum Assisted Resin Infusion (VARI) process. It has shown to be an important tool for predicting the advance of the resin flow front and possible problems and defects that can result from the infusion process.

The accuracy of results are considerably dependent on the material characterisation, particularly, the permeability of the laminate. In order to better compromise the accuracy of results with a reduced simulation time, a global permeability of the laminate should be determined, which considers the combined effect of draping, compression, orientation, etc., and which are difficult to measure. To accomplish that, infusion of the laminate (instead a single layer) to experimentally determine its permeability is, firstly, performed. Then, a simulation of the laminate may be done (admitting that it is a single layer, with the properties of the stack of plies), and the numerical and, finally, the experimental results compared to validate the material characterization. After validation, the infusion process can be simulated for complex geometries with the specific laminate, in order to define the best infusion strategy, predict potential problems and defects, forecast the infusion time, resin quantity, etc. Lastly, the infusion strategy is implemented in the production of a part, so the numerical results can be validated. This method was applied to different laminates (with 1, 5 and 10 plies) and sandwich structures, and used to produce a wide range of parts: from a guitar pickguard to a hull of a vessel.

In addition to the material characterization, several assumptions and simplifications were considered in the simulations in order to obtain more realistic results. It was assumed that the resin behaves like a Newtonian fluid with constant viscosity; presents a sufficiently long gel time and, that the curing process only starts after complete impregnation of the laminate. The mesh is built on the geometry admitting, in most of cases, a 3D tetrahedral elements for the laminate characterization, 2D triangular elements to validate the material properties and to simulate the production of complex geometries; the boundaries of the geometry, regions with sharp concave surfaces or corners, laminate transitions, foam cuts and perforations, etc., are regions with higher velocity flow, compared to that of a laminate. In sandwich structures, it is assumed that the foam does not absorb resin on its surface and within its core, and the resin only flows and fills the biding points.

In general, the simulations were in very good agreement with the experimental results, and the assumptions and simplifications, as in the production of the hull of the boat, helped to increase the accuracy of the results in one hand and, simplified and decreased the simulation time on the other hand.

References

1. Dereims A, Chatel S, Marquette P, Dufort L (2017) Accurate liquid resin infusion simulation through a fluid-solid couples approach. SAMPE 2017, Seatle, Washington, USA
2. Dereims A, Troian R, Drapier S, Bergheau J-M, de Luca P (2012) Simulation of liquid resin infusion process by finite element method. ECCM15-15th, 24–28 June 2012, Venice, Italy (2012)
3. Dereims A, Zhao S, Yu H, Pasupuleti P, Doroudian M, Rogers W, Aitharaju V (2016) Compression resin transfer molding (C-RTM) simulation using a coupled Fluid–Solid approach. American society for composites 32nd technical conference, Indiana, USA (2016)
4. Pierce RS, Falzon BG (2017) Simulating Resin Infusion through textile reinforcement materials for the manufacture of complex composite structures. Engineering 3:596–607
5. PAM-RTM (2013) User's guide & tutorials. Esi Group

6. Zhao C, Zhang G, Wu Y (2012) Resin flow behaviour simulation of grooved foam sandwich composites with vacuum assisted resin infusion (VARI) moulding process. Materials 5:1285–1296
7. Celle P, Drapier S, Bergheau J-M (2008) Numerical modelling of liquid infusion into fibrous media undergoing compaction. Eur J Mec A/Sol 27:647–667
8. Dereims A, Drapier S, Bergheau J-M, de Luca (2014) 3D robust iterative coupling of Stokes, Darcy and solid mechanics for low permeability media undergoing finite strains. ESI-Technical Paper, 1–50, November (2014)

Chapter 22
Towards CAD-Based Shape Optimization of Aircraft Engine Nozzles

Simon Bagy, Bijan Mohammadi, Michaël Mèheut, Mathieu Lallia, and Pascal Coat

Abstract Shape optimization is a powerful method to design efficient aerodynamic shapes for aircraft and engine configurations at a limited cost. However, performing an optimization on "real-world" problems, including industrial tools and processes, remains challenging. In this paper, an original approach is presented, aiming at integrating expert knowledge and reducing the dimension of the optimization search space. Thanks to this method, it becomes possible to perform gradient-free or gradient-based optimization with an industrial design workflow comprising CAD. When applied to a nozzle shape optimization problem, this approach leads to encouraging performance improvements in both inviscid and viscous cases, for a given level of fuel consumption. Moreover, the reduced number of parameters enables the use of response surfaces and a better understanding of the design space.

Keywords Optimization · Aerodynamics · Computer-Aided Design

S. Bagy (✉) · M. Lallia · P. Coat
Safran Aircraft Engines, Rond Point René Ravaud, 77550 Moissy-Cramayel, France
e-mail: simon.bagy@safrangroup.com

M. Lallia
e-mail: mathieu.lallia@safrangroup.com

P. Coat
e-mail: pascal.coat@safrangroup.com

B. Mohammadi
Institut Montpellièrain Alexander Grothendieck, Universitède Montpellier,
34090 Place Eugène Bataillon, Montpellier, France
e-mail: bijan.mohammadi@umontpellier.fr

M. Mèheut
Office National d'Etudes et de Recherches Aérospatiales, 8 rue des Vertugadins,
92190 Meudon, France
e-mail: michael.meheut@onera.fr

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_22

22.1 Introduction

Reducing fuel consumption is one of the main challenges tackled by aircraft and engine manufacturers to keep lowering the environmental footprint of air transport. In order to achieve the efficiency needed for greener configurations, designers tend to include more and more innovative technologies in their processes. In this context, shape optimization is a powerful tool to improve aerodynamic performance and has already proven its efficiency on cases of growing complexity [1]. Applying such methods to industrial design processes requires to take into account all software involved (see Fig. 22.1). In particular, Computer Aided Design (CAD) is mandatory to manage geometrical models of industrial complexity. But the integration of these models in an optimization workflow remains a major challenge, due to the great number of design parameters involved, as well as the fact that most of the CAD software is used as "black box". Consequently, this represents a limitation for the use of optimization in design phases of industrial systems.

To deal with this problem, a first approach consists in using gradient-free optimization methods. However, the cost of these methods quickly increases with the number of design parameters and becomes prohibitive for several hundreds of variables.

The second possible approach is the use of gradient-based methods. Several solutions have been developed to tackle the issue of gradient computation while including a CAD software inside the optimization loop. For instance, Banovic et al. [2] performed automatic differentiation on a CAD kernel. A second solution has been investigated by Dannenhoffer et al. [3], who has differentiated the analytic shapes obtained with a CAD software. Yet, without access to the source code, these methods cannot be considered. Danenhoffer et al. [3] and Robinson et al. [4] have studied an alternative solution, using finite-differences to compute the sensitivities of the CAD model with respect to the design parameters. They have shown that this method can give accurate sensitivities, but have also highlighted that finite-differences must be applied carefully on complex geometries.

This paper proposes an innovative method that takes expert-based knowledge into account and facilitates the use of industrial CAD software and meshing tools in optimization. At first, this method is described in details with its main advantages and drawbacks. Then, the approach is assessed on a simple nozzle shape optimization problem. First results are presented with inviscid flow computations. Then, viscous computations are performed with Reynolds Averaged Navier-Stokes (RANS) equations.

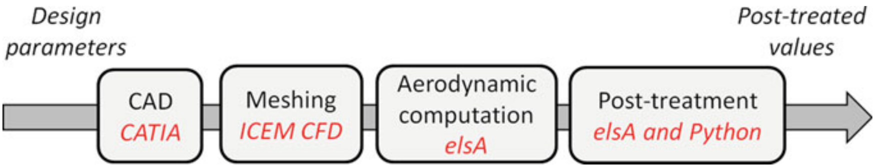


Fig. 22.1 Example of an industrial aerodynamic design workflow

22.2 Reduction of the Search Space

22.2.1 *Method of Convex Combination Based on Expert Knowledge*

Considering a case where the design tool is a CAD modeler, the number of parameters associated to the geometrical model is $N \sim 10^2$. With this modeler, n target shapes $X_{\{i=1,\dots,n\}}$ are defined in the admissible domain, with $n \ll N$ (usually, n is comprised between 2 and 10). These shapes are generated using for instance the designer's background knowledge, previous shapes database, litterature, or by taking physical constraints into account. Therefore, they will be referred as "expert configurations". Their number depends on previous optimizations, although it is recommended to start with a low number (e.g. 3) and to increase their number afterwards.

Then, the convex set is defined as a simplex $\mathcal{S}$ based on these configurations:

$$\mathcal{S} = \left\{ \sum_{i=1}^{n} \lambda_i X_i \;\middle|\; \sum_{i=1}^{n} \lambda_i = 1 \text{ and } \lambda_i \geq 0 \;\; \forall i \right\}$$

where $\lambda_i \in [0, 1]$ are the barycentric coordinates.

In this framework, optimizations are not performed on the whole design space, but only on the convex set. Thus, the dimension of the problem is reduced to n and the global design variables have been replaced with $\Lambda = \{\lambda_1, \dots, \lambda_n\} \in [0, 1]^n$.

22.2.2 *Main Advantages and Drawbacks*

This approach is particularly interesting for preliminary design: using the expert configurations to define the design space strongly reduces the risk of getting industrially unfeasible designs. Moreover, the complexity of the shapes defined with a high number of parameters is not reduced; only the way to drive the exploration has changed. Mathematically, this method proposes an innovative way to explore a high dimensional design space. Because the approach is low-dimensional, functional sensitivities can be evaluated using finite differences. It also reduces the computational cost of gradient-free techniques and enables their use. Finally, thanks to the projection theorem, the solution of this optimization is supposed to be the projection of the global solution on the "expert-based" simplex. Starting from this first solution, a second optimization can be run over the entire search space, with a method adapted for higher dimension.

The major drawback of this method is that generated shapes are limited by the expert configurations given for the combination. Further improvements are studied in order to reduce this limitation and make the method more "exploratory".

22.3 Application to the Optimization of Nozzle Shapes

The first application of the convex combination method is a simplified aircraft nacelle case, inspired from the work of Toubin et al. [5]. The goal of this study is to confirm the ability of the proposed approach to design innovative nozzle shapes while reducing the dimension of the optimization problem. This simple test case also enables to validate the proposed methodology by comparing the results with nozzle theory. For this first validation, the use of a CAD modeler in the workflow is not mandatory. It can be replaced with a relevant parameterization able to produce shapes that are consistent with the nozzle theory. Consequently, the choice has been made to keep a similar workflow as Toubin et al. [5] and to use the same parameterization tools to design the nozzle shapes.

22.3.1 Geometry and Setup

The geometry considered is based on an experimental through-flow nacelle (DLR-F6 [6]) and has an axisymmetric nozzle without central body. Two planes have been added to define the engine inflow and the injection planes of the nozzle. The resulting configuration is depicted in Fig. 22.2. Cruise flow conditions are defined at the far-field borders of the computational domain, for a Mach number of 0.82 and an altitude of 35,000 feet with zero angle of attack. The boundary condition in the injection plane is defined so that the nozzle is sonic at the throat, with a stagnation pressure $p_{i\,injection} = 62,739Pa$. Defining the nozzle pressure ratio (NPR) as the stagnation pressure at the injection plane of the nozzle $p_{i\,injection}$ divided by the static pressure at the upstream infinity $p_{s\,\infty} = 23,849Pa$, this gives $NPR = 2.63$.

22.3.2 Formulation of the Optimization Problem and Methods

The goal of this work is to increase the efficiency of the nozzle. This problem can be considered in several ways, but the choice has been made to maximize thrust for

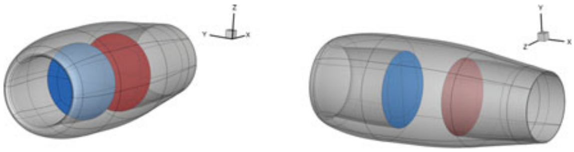


Fig. 22.2 Front and rear view of the axisymmetric nozzle, with the engine inflow and nozzle injection planes (blue and red respectively)

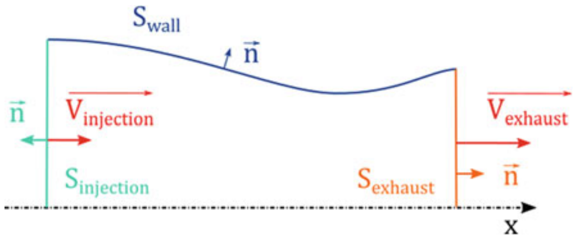


Fig. 22.3 Schematic drawing of the axisymmetric nozzle for momentum conservation

a fixed fuel consumption. Let ρ and **V** be the density and the velocity respectively, the momentum conservation in the nozzle (see Fig. 22.3) gives for a steady flow:

$$\int_S \rho \mathbf{V}(\mathbf{V}.\mathbf{n})dS = \int_S \overline{\overline{\sigma}}.\mathbf{n}dS \tag{22.1}$$

with $S = S_{injection} + S_{wall} + S_{exhaust}$ and $\overline{\overline{\sigma}}$ defined as the sum of the pressure and viscous stresses $\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}}$.

The surface integrals are developed and the equation becomes:

$$\int_{S_{injection}} \left[\rho \mathbf{V}(\mathbf{V}.\mathbf{n}) - \overline{\overline{\sigma}}\mathbf{n}\right] dS + \int_{S_{exhaust}} \left[\rho \mathbf{V}(\mathbf{V}.\mathbf{n}) - \overline{\overline{\sigma}}\mathbf{n}\right] dS = \int_{S_{wall}} \overline{\overline{\sigma}}.\mathbf{n}dS \tag{22.2}$$

The left-hand side terms of Eq. 22.2 can be associated to the “impulsion” $\mathcal{F}$ defined by Candel [7] (p. 218) expressed in the viscous case. Fixing the fuel consumption implies that the first term, $\mathcal{F}_{injection}$, remains constant. The second term, $\mathcal{F}_{exhaust}$, participates significantly to the thrust of the engine. To maximize $\mathcal{F}_{exhaust}$, it is necessary to maximize the right-hand side of the equation.

Finally, the optimization problem is defined as the minimization of a function J:

$$J = \left[\int_{S_{wall}} -\overline{\overline{\sigma}}.\mathbf{n}dS\right].\mathbf{x} \tag{22.3}$$

In the inviscid case, this term reduces to the integral of pressure stresses at the walls of the nozzle.

Considering Eq. 22.2, it also appears that increasing the mass flow rate $\dot{m} = \int_{S_{exhaust}} \rho \mathbf{V}.\mathbf{n}dS$ is favorable to thrust. In this study, the target is to maximize thrust at a given value of $\dot{m}$ and an upper constraint $\dot{m} < \dot{m}_{target}$ is defined for the optimizer.

The optimization is performed with Dakota [8], using DOT’s modified method of feasible descent [9] to explore the constrained design space. A workflow dedicated to aerodynamic design has been used to perform the optimizations (see Fig. 22.4).

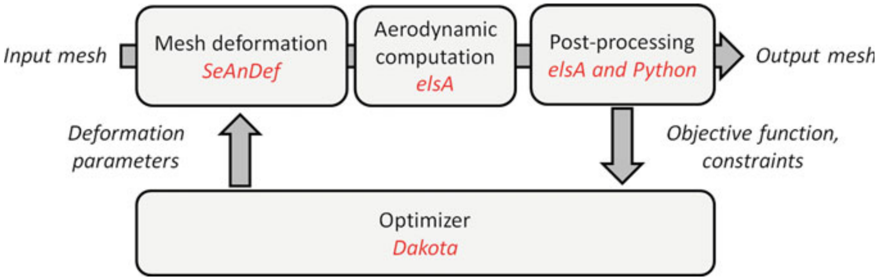


Fig. 22.4 Optimization workflow steps and associated software

22.3.3 Parameters and Mesh Deformation

The parameterization of this nacelle is based on the deformation of a reference mesh using *SeAnDef* (Sequential Analytical Deformation). This in-house code developed at ONERA applies spline deformations based on control points and has been used in previous optimization workflows [10].

This study focuses on the optimization of the nozzle shape, located downstream the injection plane. Consequently, seven control points are defined on the internal wall, as shown in Fig. 22.5. The point distribution is refined in the vicinity of the exhaust, because this region is expected to have a critical influence on the flow. The inlet is considered fixed and the external wall of the nacelle is only influenced by the radial position of the internal trailing edge point. At each point, the parameter driving the deformation in the radial direction becomes a design variable, resulting in 7 degrees of freedom for the optimization.

22.3.4 Mesh and Numerical Setup

The elsA solver [11] (ONERA-Airbus-SAFRAN property) is used for the Computational Fluid Dynamics (CFD) simulations of the airflow around the nacelle. The computations take the axial symmetry into account, and 2D planar meshes are used. In order to meet the requirements for inviscid and viscous computations, two dif-

Fig. 22.5 Setup of control points (in green) defined for the mesh deformation

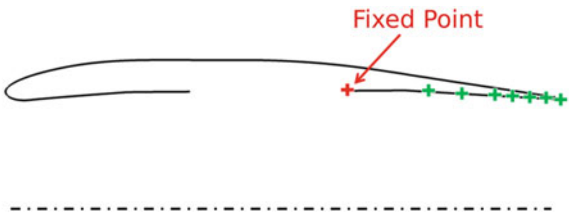


Table 22.1 Numerical parameters for inviscid and viscous flow computations

Flow model	Turbulence model	Spatial scheme	Pseudo-time stepping scheme
Euler	/	Roe	Backward Euler and LUSSOR implicit phase scalar
RANS	Spalart-Allmaras	Jameson	Backward Euler and LUSSOR implicit phase scalar

ferent meshes are generated, of $264 \cdot 10^3$ and $106 \cdot 10^3$ cells respectively. In Euler computations, strong local deformations of the nozzle shape can lead to numerical issues in the absence of viscous boundary layer. In order to avoid such phenomenon, the mesh is refined in the nozzle, leading to a higher overall number of cells than for RANS computations. Finally, a similarly refined RANS mesh has been created and has shown that the values and the gradients of the quantities of interest were not affected by the refinement difference.

The main numerical parameters can be found in Table 22.1.

22.3.5 Post-processing

After aerodynamic computations, the solver provides integrated values at the boundary conditions, used to compute the quantities of interest (mass flow rate, pressure and friction stresses at the wall for instance).

22.4 Inviscid Optimization of Nozzle Shapes

22.4.1 Nozzle Theory and Expectations

In order to get a first and simple understanding on nozzle flows, an isentropic flow hypothesis can be considered.

The nozzle is assumed to be sonic at its throat. As explained in Candel [7] (pp. 270–273), several kinds of flow regimes are possible in this situation, depending on the NPR. In particular, when the static pressure at the exhaust is equal to the external static pressure $p_{s\,\infty}$, the nozzle flow is called “adapted”. Regarding nozzle efficiency, this adapted regime represents the ideal case. For a given NPR, i.e. a given level of stagnation pressure at the entry of the nozzle, the shape has an effect on the static pressure level in the exhaust plane. In order to reach an adapted nozzle flow, it can be necessary to have a sonic throat followed by a diverging shape to accelerate the flow at a supersonic speed before the exhaust. Such shapes are called "convergent-divergent" nozzles.

Knowing the stagnation pressure injected at the entry of the nozzle, the section ratio $\frac{S_{exhaust}}{S_{throat}}$ necessary for nozzle adaptation case can be computed. By definition of the adaptation regime, $p_{s\,exhaust} = p_{s\,\infty}$. In the absence of shocks, the isentropic flow induces that the stagnation pressure is conserved and $p_{i\,injection} = p_{i\,exhaust}$. Then, using the isentropic relations:

$$M_{exhaust} = \left(\frac{2}{\gamma - 1} \left[\left(\frac{p_{s\,exhaust}}{p_{i\,exhaust}} \right)^{\frac{1-\gamma}{\gamma}} - 1 \right] \right)^{\frac{1}{2}} \tag{22.4}$$

With the Mach number at the exhaust, the section ratio between the exhaust and the sonic throat can be obtained:

$$\frac{S_{exhaust}}{S_{throat}} = \left(\frac{\gamma + 1}{2} \right)^{-\frac{\gamma+1}{2(\gamma-1)}} \frac{\left(1 + \frac{\gamma - 1}{2} M_{exhaust}{}^2 \right)^{\frac{\gamma+1}{2(\gamma-1)}}}{M_{exhaust}} \tag{22.5}$$

For $NPR = 2.63$ as defined in Sect. 22.3.1, a Mach number $(M_{exhaust})_{ideal} = 1.262$ is necessary to reach the adaptation. Only a convergent-divergent nozzle is capable of accelerating the flow to supersonic speed. Therefore, it is expected to generate such shapes to reach the best nozzle efficiency. Moreover, the associated section ratio is $\left(\frac{S_{exhaust}}{S_{throat}} \right)_{ideal} = 1.051$.

22.4.2 Optimal Shape on the Entire Design Space

A first optimization is launched over the entire design space, with a feasible descent method and inviscid flow computations (Euler equations). The algorithm starts from a geometry with the characteristics of the DLR-F6 nacelle (see Fig. 22.2). This configuration has a simply convergent nozzle and is defined as reference for this study.

After 5 gradient iterations and 89 evaluations, the optimizer leads to an optimal convergent-divergent shape that satisfies the mass flow constraint. The evolution of the area and the Mach number in this nozzle are depicted in Fig. 22.7 (blue). This shape has an area ratio $\frac{S_{exhaust}}{S_{throat}} = 1.012$, which appears to be smaller than the ratio calculated a priori with isentropic computations (see comparison in Table 22.2). Moreover, the throat is located at 95% of the nozzle length.

This first result validates the workflow and the ability of the optimizer to manage the constrained minimization problem. It also demonstrates that the process can create convergent-divergent nozzle, starting from a simply convergent shape. In addition, it shows that the 2D Euler computations and the objective function are in agreement with the isentropic nozzle theory.

22.4.3 Optimal Shape on Reduced Design Space

In a second step, the same workflow is used to perform an optimization on the design space of reduced dimension, using convex combination.

Three convergent-divergent nozzle shapes are generated as "expert" configurations, which characteristics are presented in Table 22.2. The area evolution of these nozzle shapes can be observed in Fig. 22.6.

Two have a section ratio of 1.051 as expected by the isentropic theory. The third is inspired from the optimal shape obtained on the entire design space and has larger throat, i.e. a smaller section ratio. The axial position of the throat is also expected to have a significant effect on the performance. Hence, two throat positions are defined: upstream at 73% of the nozzle length and downstream at 86%.

It must be noted that these expert configurations have different throat areas. For sonic flows, the throat drives the value of the mass flow rate. Thus, the three configurations do not have a mass flow value of $\dot{m}_{target}$; they do not verify the mass flow constraint.

The optimizer performs a feasible descent on the convex space generated by these shapes. Starting from the barycenter at $\Lambda = \{0.333; 0.333; 0.333\}$, it converges after 10 gradient iterations and 66 evaluations. The resulting optimal shape is obtained for $\Lambda = \{0.814; 0.008; 0.178\}$ and can be observed in Fig. 22.7 (green). It has a sonic throat located at 76% of the nozzle length, near the upstream position. This shape

Table 22.2 Main geometrical characteristics of the nozzle configurations

Configuration	Section ratio	Throat location (% of nozzle length)
Isentropic theory	1.051	/
Reference	1.000	100
Expert shape X_1	1.051	73
Expert shape X_2	1.051	86
Expert shape X_3	1.044	86
Full domain research optimum (Euler)	1.012	95
Convex combination optimum (Euler)	1.048	76

Fig. 22.6 Evolution of the area through nozzle shapes defined as “expert” configurations

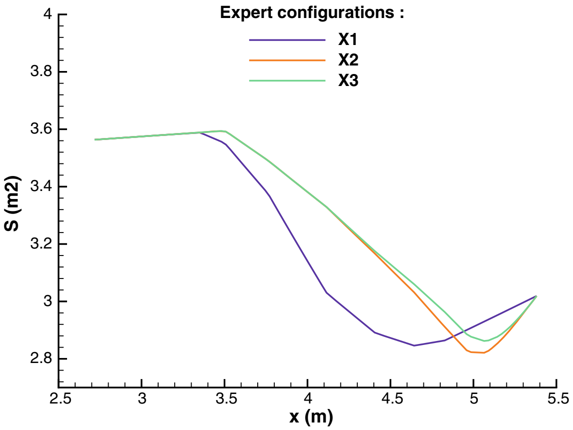
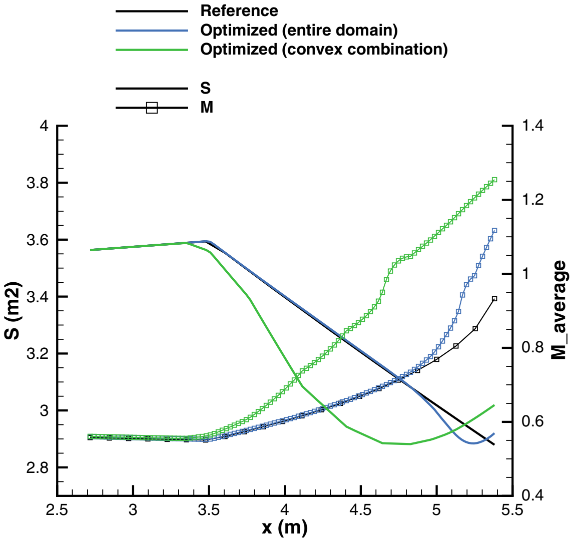


Fig. 22.7 Evolution of the area and the Mach number through nozzle shapes computed with an inviscid flow model



allows a progressive increase of the Mach number up to the exhaust. Moreover, the optimal shape verifies the mass flow constraint with $\dot{m} = \dot{m}_{target}$, in contrast with the configurations used for combination. This demonstrates the capability of the proposed method to reach shapes that validate the constraint, even by combining configurations that do not.

Finally, in comparison with the previous optimum (see Table 22.3), this configuration shows better performance. As both shapes belong to the design space of dimension 7, this suggests the possible presence of local minima. In this case, and even more so for targeted high-dimensional applications, the research of a global optimum is irrelevant due to costly computations and the number of design variables. However, it appears that the convex combination method can enable to find

Table 22.3 Comparison of nozzle shapes performance, obtained with Euler computations

Configuration	Objective function J/J_{ref}	Constraint $\dot{m}/\dot{m}_{target}$
Reference	1.0000	0.9995
Expert shape X_1	1.0441	0.9887
Expert shape X_2	1.1852	0.9738
Expert shape X_3	1.0517	0.9940
Full domain research optimum	0.9932	1.0005
Convex combination optimum	0.9775	1.0015

a better optimal solution than local exploration of the full design space. Thereby, it confirms the interest of using a method that includes "expert" knowledge.

22.4.4 Numerical Design of Experiment and Interpolation

In order to have a full understanding of the exploration performed on the domain defined with convex combination, a numerical design of experiment (DOE) is carried out. The cost of such methods is prohibitive in large dimensions; in this case, it is made possible by the reduction of the search space dimension.

A convex space generated with three expert configurations can be represented as a triangular surface, with the vertices of the triangle being the generating shapes. An uniform seeding of this space is generated with Wootton, Sergent Phan-Tan-Luu's (WSP) algorithm [12]. A resulting set of 68 designs is evaluated through the workflow and gives a discrete representation of the functions of interest (see Fig. 22.8). Despite its discrete character, a zone of interest can already be identified near the bottom left hand corner of the triangle.

The post-processed values are then interpolated on a response surface with a Kriging method [13] and plotted as a continuous response surface. The objective function map and the constraint isocontours on the reduced search space are depicted on Fig. 22.8. This interpolated response surface shows the correlation between the mass flow constraint and the objective function: the constraint isolines are strongly related to the objective function levels. This also enables to check *a posteriori* if the solution found with feasible descent algorithm is in good agreement with the response map. According to the map, the best feasible solution is obtained with $\Lambda = \{0.833; 0.000; 0.167\}$. These coordinates appear to be close to the coordinates of the optimal solution reached with gradient-based methods. Therefore, this validates the capability of the gradient-based algorithm to explore the reduced search space under constraint.

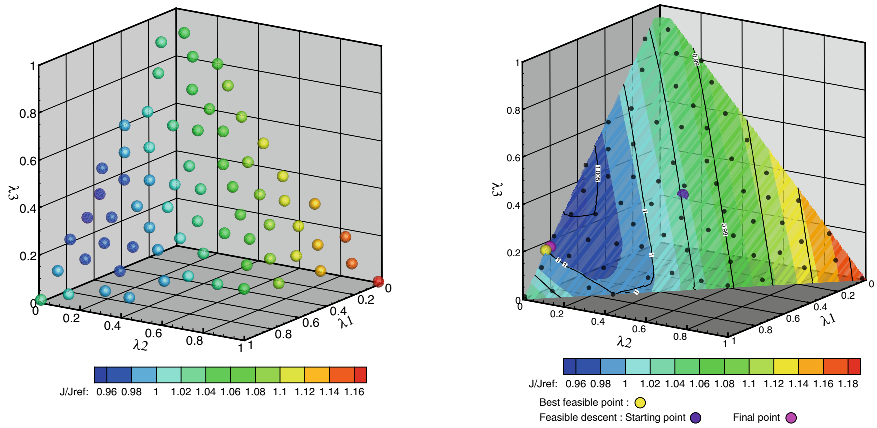


Fig. 22.8 Set of points (left) and interpolated response surface (right) obtained with a numerical DOE on inviscid computations

22.5 Viscous Optimization of Nozzle Shapes

Fluid viscosity affects the flow in the near-wall region and mixing layers, both of which are important parts of nozzle aerodynamics. As a consequence, viscosity can have a significant influence on nozzle performance and should not be neglected in design. To take this phenomenon into account, the inviscid flow model is replaced by RANS flow computations. The objective function comprises viscous stresses accordingly and all the terms of Eq. 22.3 are considered non-zero.

22.5.1 Optimal Shape on the Entire Design Space

An optimization is performed with a feasible descent algorithm on the full search domain of dimension 7. Starting from the reference shape, the algorithm reaches convergence after 5 gradient iterations, totalizing 64 evaluations.

The area and Mach number evolution through the resulting optimal shape are depicted in Fig. 22.10 (blue) and its geometric characteristics are presented in Table 22.4. It appears that the nozzle cross-sectional area is widely opened along the nozzle and reduced at the exhaust, presumably in order to respect the mass flow criterion. This shape enables a reduction of the wall friction and a good conservation of total pressure along the nozzle, which explains the gain in performance predicted by the optimizer (see Table 22.5). However, it also appears counter-intuitive regarding expert knowledge for several reasons. First of all, the nozzle is not convergent-divergent, on the contrary of nozzle theory prediction and inviscid optimization results. Then, a closer look to the design parameters shows that they mostly converge towards the lower boundary of their definition interval. This explains the strong deformation of the shape, and denotes an unexpected behavior of the optimizer. In addition, the obtained nozzle geometry has irregularities between $x = 3.5m$ and $x = 4.5m$, that can lead to poor quality flows (and flow separation in the worst case), which are highly unlikely to be favorable to nozzle performance.

This situation could be due to several reasons, including without being exhaustive:

- an optimization problem that is not well defined to reflect the improvement expected by the designer
- an inappropriate (too narrow or too large) variation range for the design variables.

At this point, it remains unclear to the authors which of these aspects may be responsible for this behavior. As this work is mainly dedicated to method assessment, it has been chosen to follow expert intuition and to dismiss this shape. In this situation, convex combination enables to perform optimization although the optimum found on the full design space is "unfeasible" from the designer's point of view. Moreover, it helps to correct his possible shortcomings in the optimization problem definition. By considering a well-chosen set of "expert" configurations, it excludes the direction of undesirable shapes and defines an industrially feasible design space.

22.5.2 Optimal Shape on the Reduced Search Space Defined with 3 "Expert" Configurations

In order to "remove" undesirable shapes from the nozzle design space, the shapes considered in Sect. 22.4.3 are re-used for convex combination. The feasible descent algorithm starts again from the barycenter and converges after 11 gradient computations and 59 evaluations.

The optimal shape is obtained with $\Lambda = \{0.731; 0.000; 0.269\}$ and the nozzle area and Mach evolution can be observed in Fig. 22.10 (green). This nozzle has a throat located at 79% of its length, with an area ratio of 1.044.

Compared to the optimum obtained with inviscid computations and convex-combination, the throat appears to be located downstream (see characteristics in Table 22.4). This position minimizes the losses due to wall friction, because it enables to keep a subsonic flow as long as possible. Therefore, it limits the higher wall friction induced by the supersonic flow in the divergent. Moreover, the throat area is larger than in the inviscid case. Since all "expert" configurations have the same exhaust area (as depicted in Fig. 22.6), the section ratio is smaller on this case. This is an effect of the viscous boundary layer; the reduced flow speed in the wall region implies that a greater throat area is needed to pass the mass flow rate $\dot{m}_{target}$.

Table 22.5 indicates that the optimized shape on the reduced space achieves a significant gain on the objective function compared to the "expert" configurations. However, this improvement is insufficient to reach a better performance than the reference nozzle. At this point, the DOE is expected to give information about how to continue the optimization process.

22.5.3 Numerical DOE

The DOE methods are applied to the viscous case. Since the three configurations used for combination have remained the same, comparison with inviscid flow is possible.

At first sight, the response map displayed in Fig. 22.9 is similar to the inviscid case of Fig. 22.8. However, the plotted objective function J is different, because of the additional viscous term in Eq. 22.3. This term appears to have a "smoothing" effect on the performance of the nozzles, but does not change the global trend of the response surface. In particular, the zone of interest is located in the same area of the map for both inviscid and viscous flows. Although viscosity modifies the objective function implicitly, it also affects the mass flow constraint. The location of the feasible-mass flow isoline ($\dot{m}/\dot{m}_{target} = 1$) has changed. Again, a strong correlation is observed between the constraint and the objective function.

According to the response map, the feasible design that minimizes J is found for $\Lambda = \{0.722; 0.003; 0.274\}$. These coordinates are similar to the coordinates obtained at convergence of the feasible descent algorithm, and confirm that the optimizer is able to converge in the vicinity of the global optimum of the reduced design space. This also indicates to the designer that the set of "expert" configurations is not sufficient to improve significantly the objective function. Consequently, an enrichment of the "expert database" is suggested to continue the optimization process.

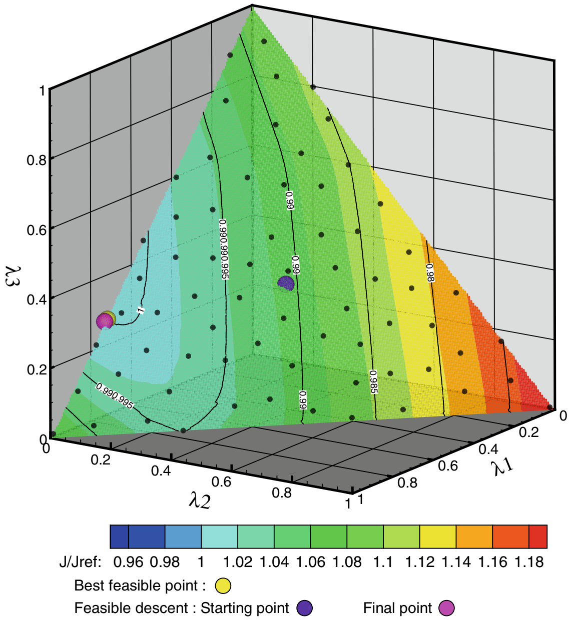


Fig. 22.9 Interpolated response surface obtained with a numerical DOE on RANS computations

22.5.4 *Optimal Shape on the Reduced Search Space Defined with 4 Expert Configurations*

A new configuration is added to the existing set of 3 configurations. In order to introduce a convergent nozzle, the reference shape is chosen as X_4.

Then, the feasible descent algorithm is launched on the space generated by the convex combination of these 4 configurations. The optimal shape found in Sect. 22.5.2 is defined as starting point for the feasible descent. After 2 gradient computations and a total of 39 evaluations, the algorithm returns an optimal set of coordinates $\Lambda = \{0.855; 0.000; 0.000; 0.145\}$.

First of all, these coordinates indicate that the optimizer takes advantage of the newly introduced expert configuration. The resulting shape is a convergent-divergent nozzle, with a throat located at 73% of the nozzle length and an area ratio of 1.038 (see Table 22.4). The area and Mach number evolution in this nozzle are depicted in Fig. 22.10 (red). The higher velocity along the nozzle implies that wall friction stresses are more important in this case than in the previous optimal nozzle. However, this effect is balanced by a favorable pressure resulting force in the divergent, that is extended. Thanks to this, this nozzle has an improved performance compared to the optimum obtained with 3 configurations and the reference. This improvement is smaller than the one obtained with full design space exploration, which is a possible consequence of the search on a subspace of the design space. However, in a context where the search of a global optimum is not affordable, the convex-combination method has the advantage of generating an understood and well-defined optimal shape.

Finally, the viscous case highlights new aspects of the convex combination method. By generating shapes as a combination of well-defined designs, it enables to avoid the generation of undesirable or erratic shapes. Then, it shows that if a set of

Table 22.4 Main geometrical characteristics of the nozzle configurations

Configuration	Section ratio	Throat location (% of nozzle length)
Isentropic theory	1.051	/
Reference (X_4)	1.000	100
Expert shape X_1	1.051	73
Expert shape X_2	1.051	86
Expert shape X_3	1.044	86
Full domain research optimum (RANS)	1.000	100
Convex combination optimum (RANS, 3 expert shapes)	1.044	79
Convex combination optimum (RANS, 4 expert shapes)	1.038	73

Fig. 22.10 Evolution of the area and the Mach number through nozzle shapes computed with a viscous flow model

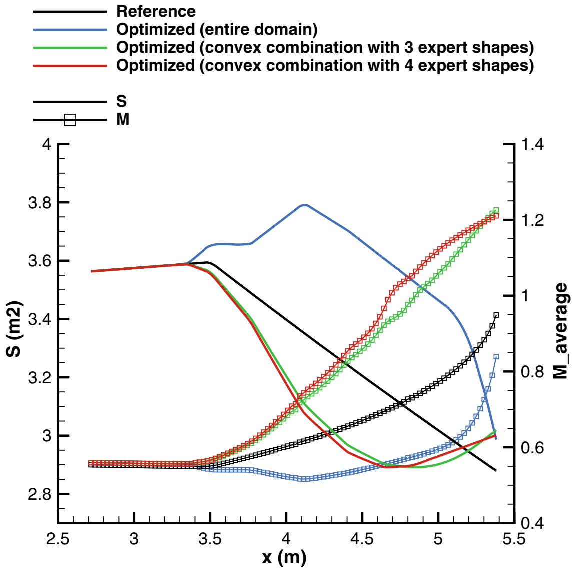


Table 22.5 Comparison of nozzle shapes performance, obtained with RANS computations

Configuration	Objective function J/J_{ref}	Constraint $\dot{m}/\dot{m}_{target}$
Reference	1.0000	0.9997
Expert shape X_1	1.0900	0.9881
Expert shape X_2	1.1981	0.9716
Expert shape X_3	1.0830	0.9848
Full domain research optimum (RANS)	0.9598	1.0000
Convex combination optimum (RANS, 3 expert shapes)	1.0047	1.0000
Convex combination optimum (RANS, 4 expert shapes)	0.9977	1.0000

expert configurations is insufficient to reach significant performance improvement, it can be enriched by new configurations. Moreover, this case demonstrates again the ability of the method to find optimal shapes in a design space where the objective function and the constraint have similar evolution.

22.6 Physical Analysis of the Nozzle Optimization Problem

The main part of this paper focuses on optimization processes and in order to perform numerical analyses, tolerances have been defined on the mass flow constraint. However, discrepancies in the mass flow rate values can alter the predicted performance level. Consequently, from a physical point of view, nozzles should only be compared at same mass flow rate.

To compensate the mass flow differences and quantify the associated performance variations, a corrective term can be defined. This term is based on the sensitivity of the objective function with respect to the design constraint, that is computed from the gradients of the functions of interest. Consequently, an estimate of the performance value at same mass flow, called $J/Jref@m_{target}$, is computed for the optimized shapes and presented in Table. 22.6.

It appears that small mass flow perturbations can significantly alter the predicted performance, in comparison with the improvements achieved with optimization. Again, this reveals the tight link between mass flow and thrust, which are the two values driving nozzle performance. In order to consider this link and enable efficient optimizations, the *a posteriori* analysis of this work advocates for the use of objective functions integrating the mass flow constraint and penalizing the configurations that do not respect it.

Table 22.6 Performances of the optimized nozzle shapes, including mass flow correction estimate

Configuration	Objective function J/J_{ref}	Constraint $\dot{m}/\dot{m}_{target}$	$J/J_{ref}@\dot{m}_{target}$
Reference (Euler)	1.0000	0.9995	1.0000
Full domain research optimum (Euler)	0.9932	1.0005	0.9995
Convex combination optimum (Euler with 3 expert shapes)	0.9775	1.0015	0.9880
Reference (RANS)	1.0000	0.9997	1.0000
Full domain research optimum (RANS)	0.9598	1.000	0.9615
Convex combination optimum (RANS with 4 expert shapes)	0.9977	1.000	0.9994

22.7 Conclusions

This paper proposes an original approach specially developed to use an industrial workflow and CAD methods within optimization processes. The method described, based on the idea to generate designs as a combination of reference shapes appears to have several advantages. By reducing the dimension of the search space, it enables the use of gradient-free or finite-difference gradient methods. It also places the industrial expert knowledge at the heart of the optimization process and reduces the risk of producing unfeasible shapes.

When applied to the design of a simple nozzle with inviscid flow computations, this approach shows that an improvement is possible by taking into account expert knowledge. In addition, reducing the complexity of the search by a reduction of the dimension of the search space has permitted to find a more efficient shape than by exploring the full domain. Hence, the convex combination method can help reducing the risk of being captured by local minima, whose number increases with the dimension of the space, especially when involving a CAD environment and trigonometric manipulations. With three expert configurations, it also gives the opportunity to draw response surfaces for the quantities of interest. These figures improve the understanding of the optimizer behavior and more generally of the nozzle shape problem.

Introducing flow viscosity leads the optimization on the full design space to a non-acceptable shape for industrial designers. In this case, convex combination with an adapted choice of expert configurations enables to perform optimization while remaining in an acceptable design space. After an optimization and a DOE, the first set of 3 expert configurations appears insufficient to reach an optimum with significant performance improvements. Consequently, the expert database is enriched with a new configuration, and thanks to this the optimizer succeeds in finding a new optimal shape with better efficiency.

In the end, this approach introduces new aspects of design, by mixing expert knowledge and algorithmic exploration and enabling a further understanding of the design space. Moreover, this method is adapted to deal with complex industrial cases. By enriching the database of expert configurations, a wider design space can be explored, while keeping a low number of variables. Therefore, it is expected to show interesting results for geometries of higher complexity and number of parameters.

Acknowledgments The studies presented in this article have used the ONERA-Airbus-SAFRAN elsA software whose development is partially funded by Airbus, Safran and ONERA, which are co-owners of this software.

References

1. Mohammadi B, Pironneau O (2010) Applied shape optimization for fluids. Oxford University Press, Oxford
2. Banović M, Mykhaskiv O, Auriemma S, Walther A, Legrand H, Müller JD (2018) Optim Methods Softw 33(4–6):813
3. Dannenhoffer JF, Haimes R (2015) In:53rd AIAA aerospace aerospace sciences meeting
4. Robinson TT, Armstrong CG, Chua HS, Othmer C, Grahs T (2009) In: 8th world congress on structural and multidisciplinary optimization
5. Toubin H, Salah El Din I, Meheut M (2014) In: 52nd AIAA aerospace sciences meeting. AIAA SciTech
6. Rossow CC, Godard JL, Hoheisel H, Schmitt V (1994) J Aircr 31(5):1022
7. Candel S (1990) Mécanique des fluides. Dunod
8. Adams B, Bohnhoff W, Dalbey K, Eddy J, Eldred M, Gay D, Haskell K, Hough P, Swiler L (2009) Sandia National Laboratories, Technical report SAND2010-2183
9. Vanderplaats Research and Development (1995)
10. Meheut M, Arntz A, Carrier G (2012) In: 30th AIAA applied aerodynamics conference, p 3122
11. Cambier L, Heib S, Plot S (2013) Mech Ind 14(3):159
12. Santiago J, Claeys-Bruno M, Sergent M (2012) Chemometr Intell Lab Syst 113:26
13. Stein, ML (2012) Interpolation of spatial data: some theory for kriging. Springer Science & Business Media

Chapter 23
A Two-Phase Heuristic Coupled DIRECT Method for Bound Constrained Global Optimization

M. Fernanda P. Costa, Edite M. G. P. Fernandes, and Ana Maria A. C. Rocha

Abstract In this paper, we investigate the use of a simple heuristic in the DIRECT method context, aiming to select a set of the hyperrectangles that have the lowest function values in each *size* group. For solving bound constrained global optimization problems, the proposed heuristic divides the region where the hyperrectangles with the lowest function values in each *size* group lie into three subregions. From each subregion, different numbers of hyperrectangles are selected depending on the subregion they lie. Subsequently, from those selected hyperrectangles, the potentially optimal ones are identified for further division. Furthermore, the two-phase strategy aims to firstly encourage the global search and secondly enhance the local search. Global and local phases differ on the number of selected hyperrectangles from each subregion. The process is repeated until convergence. Numerical experiments carried out until now show that the proposed two-phase heuristic coupled DIRECT method is effective in converging to the optimal solution.

Keywords Global Optimization · DIRECT Method · Heuristics

M. F. P. Costa (✉)
Centre of Mathematics, Department of Mathematics, University of Minho,
Campus de Gualtar, 4710-057, Braga, Portugal
e-mail: mfc@math.uminho.pt

E. M. G. P. Fernandes
ALGORITMI Center, University of Minho,
Campus de Gualtar, 4710-057, Braga, Portugal
e-mail: emgpf@dps.uminho.pt

A. M. A. C. Rocha
ALGORITMI Center, Department of Production and Systems, University of Minho,
Campus de Gualtar, 4710-057, Braga, Portugal
e-mail: arocha@dps.uminho.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_23

23.1 Introduction

This paper addresses the use of a DIRECT-type method that coupled with a simple heuristic and a two-phase strategy aims to globally solve non-smooth and non-convex bound constrained optimization problems. The bound constrained global optimization (BCGO) problem can be stated as:

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \ , \tag{23.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear function and $\Omega = \{\mathbf{x} \in \mathbb{R}^n : -\infty < l_i \leq x_i \leq u_i < \infty, i = 1, \ldots, n\}$ is a bounded feasible region. We assume that the optimal set $\mathbf{X}^*$ of the problem (23.1) is nonempty and bounded, $\mathbf{x}^*$ is a global minimizer and f^* represents the global optimal value.

When the function f is non-smooth, or its evaluation requires different simulations, and those simulations may add noise to the problem, analytical or numerical gradient-based methods may fail to solve the problem (23.1). Derivative-free methods, like the DIRECT method [1, 2], can solve it. The main idea in the DIRECT method is the partition of the feasible region into an increasing number of each time smaller hyperrectangles. At each iteration, a set of the most promising hyperrectangles are identified for further division. DIRECT needs to store all the information about all the generated hyperrectangles. This means that for larger dimensional problems, computational requirements may prevent DIRECT to find a high quality solution. DIRECT has strong convergence properties and produces a good coverage of the feasible region [3]. For the hyperrectangle division, DIRECT uses two criteria: the *size* of the hyperrectangle to favor the global search feature of the algorithm and the *value* of the hyperrectangle, translated by the objective function value at the center point of the hyperrectangle, to give preference to its local search feature. DIRECT-type algorithms that are more biased toward local search are proposed in [4, 5]. They are mostly suitable for small problems with one global minimizer and a few local minimizers. In [3], the deterministic partition strategy of the DIRECT method is used, in a multi-start context, to perform local minimizations starting from the center points of the most promising hyperrectangles. Globally biased searches are also reinforced in DIRECT by making use of a new technique for selecting the hyperrectangles to be divided [6–8].

For further details on the original DIRECT and other recent interesting modifications, we refer the reader to [6–10].

This paper investigates the use of a DIRECT-type method coupled with a heuristic aiming to potentiate the exploration of the most promising regions in the DIRECT method context. The heuristic categorizes the hyperrectangles with the lowest function values in each *size* group into three subregions for further sampling and division. Additionally, a two-phase strategy aims to cyclically encourage the global search phase (first phase) and enhance the local search one (second phase). Our proposal reinforces the global search capabilities of the DIRECT by avoiding the selection of the hyperrectangles that were mostly divided and choosing all the hyperrectangles with largest sizes (first phase). Conversely, when the new algorithm enters the second phase, the hyperrectangles with largest sizes are mostly prevented from being selected and the ones with smallest sizes are all included in the selection.

The paper is organized as follows. Section 23.2 briefly presents the main ideas of the DIRECT method and Sect. 23.3 describes the heuristic and the two-phase strategy in the DIRECT method context. Finally, Sect. 23.4 contains the results of our preliminary numerical experiments and we conclude the paper with the Sect. 23.5.

23.2 DIRECT Method

The DIRECT (DIviding RECTangles) algorithm has been originally proposed to solve BCGO problems like (23.1) where f is assumed to be a continuous function, by producing finer and finer partitions of the hyperrectangles generated from Ω [1]. The algorithm is a modification of the standard Lipschitzian approach, in which f must satisfy the Lipschitz condition

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq K\|\mathbf{x}_1 - \mathbf{x}_2\| \text{ for all } \mathbf{x}_1, \mathbf{x}_2 \in \Omega\ ,$$

where $K > 0$ is the Lipschitz constant. DIRECT is a derivative-free and deterministic global optimizer since it is able to explore potentially optimal regions in order to converge to the global optimum solution, thus avoiding to be trapped in a local optimum solution. It does not require any derivative information or the value of the Lipschitz constant [2]. DIRECT views the Lipschitz constant as a weighting parameter that balances global and local search. These searches are carried out by exploring some of the hyperrectangles in the current partition of Ω, in order to divide them further [5, 11]. First, the method organizes hyperrectangles by groups of the same *size* and considers dividing in each group the hyperrectangles that have the lowest value of the objective function—herein denoted by *candidate* hyperrectangles. However, not all of these *candidate* hyperrectangles are divided. The selection falls on the hyperrectangles that satisfy the following two criteria that define a potentially optimal hyperrectangle (POH):

Definition 1 Given the partition $\{P^i : i \in I\}$ of Ω, let ε be a positive constant and let $f_{\min}$ be the current best function value. A hyperrectangle j is said to be potentially optimal if there exists some rate-of-change constant $\hat{K}_j > 0$ such that

$$f(\mathbf{c}_j) - \frac{\hat{K}_j}{2}\|\mathbf{u}^j - \mathbf{l}^j\| \leq f(\mathbf{c}_i) - \frac{\hat{K}_j}{2}\|\mathbf{u}^i - \mathbf{l}^i\|,\ \forall i \in I \tag{23.2}$$

$$f(\mathbf{c}_j) - \frac{\hat{K}_j}{2}\|\mathbf{u}^j - \mathbf{l}^j\| \leq f_{\min} - \varepsilon|f_{\min}|\ , \tag{23.3}$$

where $\mathbf{c}_j$ is the center and $\|\mathbf{u}^j - \mathbf{l}^j\|/2$ is a measure of the *size* of hyperrectangle j.

The use of $\hat{K}_j$ intends to show that it is not the Lipschitz constant but it is just a rate-of-change constant [1]. Condition in (23.2) aims to check if the lower bound on the minimum of f on the hyperrectangle j is lower than the lower bounds on the minima of the other hyperrectangles of the partition P^i (for the hyperrectangle j to be potentially optimal). Condition (23.3) aims to balance the local and global search and prevents the algorithm from searching locally a region where very small improvements are obtained. The parameter ε aims to ensure that a sufficient improvement of f for the hyperrectangle j will be potentially found based on the current $f_{\min}$ [12, 13]. The value of $f_{\min} - \varepsilon|f_{\min}|$ (in contrast to $f_{\min}$) prevents the hyperrectangle with the smallest objective function value from being a POH.

DIRECT can be briefly described by Algorithm 1 [1].

```
Input: $f$, $\Omega$.
Output: $(\mathbf{x}_{\min}, f_{\min})$.
Normalize $\Omega$ to be the unit hypercube and compute $f(\mathbf{c})$ where $\mathbf{c}$ is the center; Set $k = 0$,
$f_{\min} = f(\mathbf{c})$, $\mathbf{x}_{\min} = \mathbf{c}$;
while Stopping condition is not satisfied do
    Define the set $I_k$ of the candidate hyperrectangles; Identify the set $O_k \subseteq I_k$ of POH;
    while $O_k \neq \emptyset$ do
        Select a hyperrectangle $j \in O_k$; Identify the set $L_j$ of dimensions with maximum
        size $\delta_{max}$; Set $\delta = (1/3)\delta_{max}$;
        for all $i \in L_j$ do
            Sample $f$ at $\mathbf{c}_j \pm \delta\mathbf{e}_i$; Divide hyperrectangle $j$ into thirds along the dimensions
            in $L_j$ starting with the dimension with lowest
            $w_i = \min\{f(\mathbf{c}_j + \delta\mathbf{e}_i), f(\mathbf{c}_j - \delta\mathbf{e}_i)\}$ and continue until the dimension with
            highest $w_i$;
        end
        Set $O_k = O_k \setminus \{j\}$;
    end
    Update $f_{min} = \min_{i \in I_k} f(\mathbf{c}_i)$; Set $\mathbf{x}_{\min} = \arg\min_{i \in I_k} f(\mathbf{c}_i)$; Set $k = k + 1$;
end
```

Algorithm 1: DIRECT algorithm

Identifying the set of POH can be regarded as a problem of finding the extreme points on the lower right convex hull of a set of points in the plane [1]. A 2D-plot can be used to identify the set of POH. The horizontal axis corresponds to the *size* of the hyperrectangle and the vertical axis corresponds to the f value at the center of the hyperrectangle. Figures 23.1a, b show the center points of the hyperrectangles (marked with ‘red’ ‘+’ in the plots) generated up to iteration 4 (after 47 function evaluations) and iteration 7 (after 159 function evaluations) respectively, of DIRECT when solving the problem:

$$\min_{\mathbf{x} \in \Omega} \sum_{i=1}^{4} |x_i| + 1 \ , \tag{23.4}$$

where $\Omega = \{\mathbf{x} \in \mathbb{R}^4 : -2 \leq x_i \leq 3\}$ [14]. The mark that identifies a *candidate* hyperrectangle is a ‘magenta’ *diamond* and the mark to identify a POH is a ‘black’ *square*.

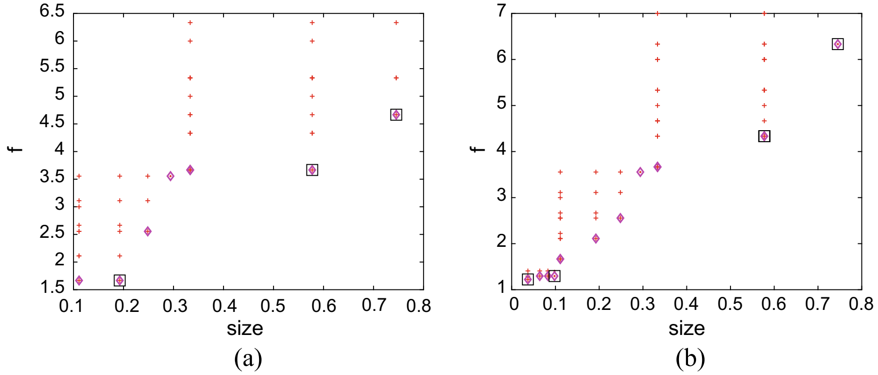


Fig. 23.1 Points representing hyperrectangles, *candidate* hyperrectangles and POH, when solving the problem (23.4) by DIRECT. **a** Iteration 4. **b** Iteration 7

The identified POH at iteration 4 were divided and generated smaller hyperrectangles. They are no longer hyperrectangles of that size at iteration 7, although other hyperrectangles with the same sizes and higher function values are identified as POH.

23.3 Two-Phase Heuristic Coupled DIRECT Method

In this section, we reveal how the DIRECT algorithm is modified to incorporate a heuristic that aims to divide a promising search region into three subregions. The implementation of the two-phase strategy aims to drastically reduce the selection of the mostly divided hyperrectangles and, in contrast, select all the hyperrectangles that have the lowest function values in each group of the largest sizes, when a global search phase seems convenient. Conversely, for the local search phase, all the hyperrectangles that have the lowest function values in each group of the smallest sizes are selected and, at the same time, the selection of the largest hyperrectangles are greatly reduced.

23.3.1 Heuristic

POH either have center points with low function values or are large enough to provide good and unexplored regions for the global search [14]. Hyperrectangles with the smallest *sizes* are the ones that were mostly divided so far. On the other hand, hyperrectangles with large *sizes* were the least divided. Avoiding the identification of POH that were mostly divided can enhance the global search capabilities of DIRECT [7]. Conversely, identifying POH that are close to the hyperrectangle which corresponds to $f_{\min}$ may improve the local search process in DIRECT. Thus, at any iteration k, the present heuristic incorporated into the DIRECT method aims to divide the region of the *candidate* hyperrectangles (the ones with least function values at each *size* group) into three subregions. The leftmost subregion includes hyperrectangles with indices based on *size* that are larger than $i_l = \lfloor 2/3 i_{\min} \rfloor$ (denoting the set by I_k^3), where $i_{\min}$ is the index based on the *size* of the hyperrectangle that corresponds to $f_{\min}$. The rightmost subregion contains the hyperrectangles with indices that are smaller than $i_u = \lfloor 1/3 i_{\min} \rfloor$ (denoting the set by I_k^1). The middle subregion contains hyperrectangles with indices i that satisfy $i_l \leq i \leq i_u$ (denoting the set by I_k^2). (We remark that the larger the *size*, the smaller is the index based on *size*.)

We present in Algorithm 2 the main steps of the proposed heuristic to be integrated into the DIRECT method, coupled with the two-phase strategy (see details in the next subsection).

23.3.2 Two-Phase Strategy

Since the balance between global and local information must be provided with caution so that convergence to the global solution is guaranteed and stagnation in a local solution is avoided, the two-phase strategy performs a cycling process between a globally biased set of iterations and locally biased iterations. The first phase (identified in the algorithm by 'phase = global') runs for $G_{\max}$ iterations and aims to potentiate the exploration of the hyperrectangles with the largest *sizes*. Here, all *candidate* hyperrectangles with indices based on *size* in I_k^1 are selected. From the middle region, 50% of the indices in the set I_k^2 are randomly selected and the corresponding *candidate* hyperrectangles are used in the selection. From the leftmost subregion, 10% of the indices in the set I_k^3 are randomly selected and the corresponding *candidate* hyperrectangles are selected. Thereafter, the set of POH are identified (following Definition 1) from all these selected hyperrectangles.

The second phase runs for $L_{\max}$ iterations. Now, all *candidate* hyperrectangles that have indices in the set I_k^3 are selected, 50% of randomly selected indices from I_k^2 are used to choose the corresponding *candidate* hyperrectangles, and 10% of randomly selected indices from I_k^1 are used to pick the corresponding *candidate* hyperrectangles. Then, based on all these selected hyperrectangles, the set of POH are identified. This process is repeated until convergence.

Figures 23.2a, b show the centers of the hyperretangles generated by Algorithm 2 up to iteration 4 (after 43 function evaluations) and iteration 7 (after 79 function evaluations) respectively, when solving the problem (23.4). In each plot, the 'green' *circles* correspond to the selected *candidate* hyperrectangles from the set I^3, the 'magenta' *diamonds* correspond to the selected *candidate* hyperrectangles from I^2, and the 'blue' '*' correspond to the selected *candidate* hyperrectangles from I^1. The identified POH are marked with the 'black' *squares*. Comparing with the previous Fig. 23.1a, b obtained from DIRECT, it may be concluded that the heuristic and the two-phase strategy have reduced the number of selected *candidate* hyperrectan-

```
Input: $f$, $\Omega$, $G_{\max}$, $L_{\max}$.
Output: $(\mathbf{x}_{\min}, f_{\min})$.
Normalize $\Omega$ to be the unit hypercube and compute $f(\mathbf{c})$ where $\mathbf{c}$ is the center;
Set $k = 0$, $f_{\min} = f(\mathbf{c})$, $\mathbf{x}_{\min} = \mathbf{c}$; phase=global; $k_G = 0$, $k_L = 0$
while Stopping condition is not satisfied do
    Identify the indices based on size $i_l = \lfloor 2/3 i_{\min} \rfloor$ and $i_u = \lfloor 1/3 i_{\min} \rfloor$ and define the sets
    of indices $I_k^1$, $I_k^2$, $I_k^3$ of candidate hyperrectangles;
    if phase=global then
        Set $H_k^1 = I_k^1$; Randomly select 50% of indices in $I_k^2$ to define $H_k^2$; Randomly select
        10% of indices in $I_k^3$ to define $H_k^3$; Set $k_G = k_G + 1$;
    else
        Set $H_k^3 = I_k^3$; Randomly select 50% of indices in $I_k^2$ to define $H_k^2$; Randomly select
        10% of indices in $I_k^1$ to define $H_k^1$; Set $k_L = k_L + 1$;
    end
    Set $H_k = H_k^3 \cup H_k^2 \cup H_k^1$; Identify the set $O_k \subseteq H_k$ of POH;
    while $O_k \neq \emptyset$ do
        Select a hyperrectangle $j \in O_k$; Identify the set $L_j$ of dimensions with maximum
        size $\delta_{max}$; Set $\delta = (1/3)\delta_{max}$;
        for all $i \in L_j$ do
            Sample $f$ at $\mathbf{c}_j \pm \delta \mathbf{e}_i$; Divide hyperrectangle $j$ into thirds along the dimensions
            in $L_j$ starting with the dimension with lowest
            $w_i = \min\{f(\mathbf{c}_j + \delta \mathbf{e}_i), f(\mathbf{c}_j - \delta \mathbf{e}_i)\}$ and continue until the dimension with
            highest $w_i$;
        end
        Set $O_k = O_k \setminus \{j\}$
    end
    Update $f_{min} = \min_{i \in H_k} f(\mathbf{c}_i)$; Set $\mathbf{x}_{\min} = \arg\min_{i \in H_k} f(\mathbf{c}_i)$;
    if phase=global and $k_G \geq G_{\max}$ then
        Set phase=local; $k_G = 0$;
    else
        if phase=local and $k_L \geq L_{\max}$ then
            Set phase=global; $k_L = 0$;
        end
    end
    Set $k = k + 1$;
end
```

Algorithm 2: Two-phase heuristic coupled DIRECT algorithm

gles from which POH are identified, without affecting the convergence to a global solution.

23.4 Numerical Experiments

Numerical experiments were carried out to analyze the performance of the presented two-phase heuristic coupled DIRECT method, when compared with other DIRECT-type methods. The MATLAB® (MATLAB is a registered trademark of the MathWorks, Inc.) programming language is used to code the algorithm and the tested problems. The parameter ε is set to $1E-04$. Because there are some elements of randomness in the algorithm, each problem was solved 20 times by the algorithm.

23.4.1 Termination Based on a Budget

First, we want to analyze what would be the most favorable set of $G_{\max}$ and $L_{\max}$ to be used in the Algorithm 2. The following three sets are tested:

- $G_{\max} = 10$ and $L_{\max} = 10$ giving the Variant V_1;
- $G_{\max} = 10$ and $L_{\max} = 5$ giving the Variant V_2;
- $G_{\max} = 5$ and $L_{\max} = 10$ giving the Variant V_3.

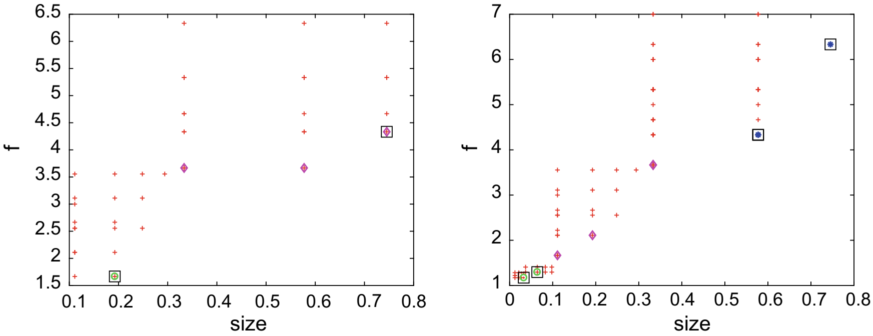
The algorithm runs for a budget of 100 function evaluations. This type of stopping condition is what would be used in practice [4].

The well-known Jones set of benchmark problems [1, 4, 8–11, 14–16] is used to compare the above defined three variants of the Algorithm 2. The Jones set contains nine problems: Shekel 5 (S5) with $n = 4$, Shekel 7 (S7) with $n = 4$, Shekel 10 (S10) with $n = 4$, Hartman 3 (H3) with $n = 3$, Hartaman 6 (H6) with $n = 6$, Branin (BR) with $n = 2$, Goldstein and Price (GP) with $n = 2$, Six-Hump Camel (C6) with $n = 2$, Schubert (SHU) with $n = 2$.

Table 23.1 shows the *perror* value given by

$$perror \equiv \frac{(f_{\min} - f^*)}{|f^*|} \,, \tag{23.5}$$

where $f_{\min}$ is the best obtained function value and f^* is the best known global minimum. Our results are compared to those reported in [4]. The *perror* value reported from our algorithm is obtained by using the average value of the solutions



(a) (b)

Fig. 23.2 Points representing hyperrectangles, selected *candidate* hyperrectangles and POH, when solving the problem (23.4) by Algorithm 2. **a** Iteration 4. **b** Iteration 7

Table 23.1 Achieved $perror$ for 100 function evaluations, using three variants of Algorithm 2

	Variant V_1	Variant V_2	Variant V_3	DIRECT-l[a]
Problem	$perror$	$perror$	$perror$	$perror$
S5	$0.12E+00$	$0.17E+00$	$0.21E+00$	$0.59E-02$
S7	$0.58E-02$	$0.58E-02$	$0.62E-01$	$0.58E-02$
S10	$0.57E-02$	$0.57E-02$	$0.81E-01$	$0.41E-02$
H3	$0.66E-03$	$0.62E-03$	$0.77E-03$	$0.85E-04$
H6	$0.13E+00$	$0.13E+00$	$0.13E+00$	$0.23E-01$
BR	$0.16E-03$	$0.19E-03$	$0.20E-03$	$0.39E-03$
GP	$0.27E-03$	$0.27E-03$	$0.14E-02$	$0.27E-03$
C6	$0.10E-01$	$0.11E-01$	$0.63E-02$	$0.16E-01$
SHU	$0.83E+00$	$0.83E+00$	$0.83E+00$	$0.82E+00$

[a] Results (locally-biased form) reported in [4]

$f_{\min}$ obtained over the 20 runs. Although the differences in the performance of the Variants V_1 and V_2 are almost negligible, Variant V_1 is slightly superior, and both outperform the Variant V_3. We may conclude that adopting a larger maximum number of global search iterations gives a better advance in the convergence issue. The comparison with the results in [4] is slightly favorable to the therein locally-biased form of the DIRECT algorithm since it finds slightly better solutions for S5, H3 and H6. However, the results for the remaining six test problems are almost identical to our results.

23.4.2 Termination Based on the Known Global Minimum

We now test the Algorithm 2 with a stopping condition that uses the knowledge of the global minimum f^*. The algorithm aims to guarantee a solution as close as possible to the f^*. Thus, the algorithm stops when

$$perror \leq \tau \ , \tag{23.6}$$

where $perror$ has been defined in (23.5) and τ is a positive small tolerance. It is assumed that $f^* \neq 0$. However, if condition (23.6) is not satisfied, the algorithm runs until a specified number of function evaluations is reached. When $f^* = 0$, the $perror$ becomes $f_{\min}$.

Based on the previous results, we compare Variant V_1 and Variant V_2 of Algorithm 2 with other DIRECT-type algorithms and some stochastic heuristics. The nine problems of the Jones set are used. Table 23.2 shows the number of function evaluations required to achieve a solution with accuracy given by $\tau = 1E-04$ and $\tau = 1E-06$, in the context of the stopping condition (23.6). The results reported from the two variants of Algorithm 2 correspond to the average value of the required

Table 23.2 Number of function evaluations required by the algorithms, with τ as shown in each row

Algorithm	τ	S5	S7	S10	H3	H6	BR	GP	C6	SHU
Variant V_1	$1E-04$	256	173	171	141	488	145	129	190	2093
	$1E-06$	329	538	580	1140	6908	258	208	362	2684
Variant V_2	$1E-04$	201	170	171	137	454	147	127	179	2409
	$1E-06$	704	430	480	1027	5587	246	209	317	2567
RDIRECT-b[a]	$1E-04$	159	157	157	173	559	181	175	115	3501
	$1E-06$	251	325	325	853	1209	287	373	115	4259
DIRECT[b]	$1E-04$	155	145	145	199	571	195	191	145[c]	2967
	$1E-06$	255	4879	4939	751	182623	377	305	211	3867
DIRECT-GL[d]	$1E-04$	1227	1141	1151	379	4793	333	223	–	425
mDIRECT[e]	$1E-04$	155	145	145	199	571	259	191	285	3663
DISIMPL-V[f]	$1E-04$	2454	723	750	261	6799	242	17	337	4509
DISIMPL-C[f]	$1E-04$	90948	(fail)	(fail)	334	25334	292	180	308	518
DTS$_{APS}$[g]	$1E-04$	819	812	828	438	1787	212	230	–	274
(% succ)		(75)	(65)	(52)	(100)	(83)	(100)	(100)	–	(92)
m-AFSA[h]	$1E-04$	1183	1103	1586	1891	2580	475	417	247	–
St-Coord_D[i]	$1E-04$	–	–	–	–	–	239	1564	512	–

[a] Results reported in [9]; [b] Results reported in [9], for both values of τ;
[c] Different from result in [1] (285) for $\tau = 1E-04$; [d] Results in [8]; – Not available;
[e] Results in [14] (with a modified update to (23.3)); [f] Results reported in [10];
[g] Results reported in [15]; [h] Results reported in [16]; [i] Results reported in [17]

number of function evaluations of the 20 runs. The results from the other DIRECT-type algorithms are taken from their original papers [1, 8–10, 14], unless otherwise stated. The maximum number of function evaluations is set to $1E+05$.

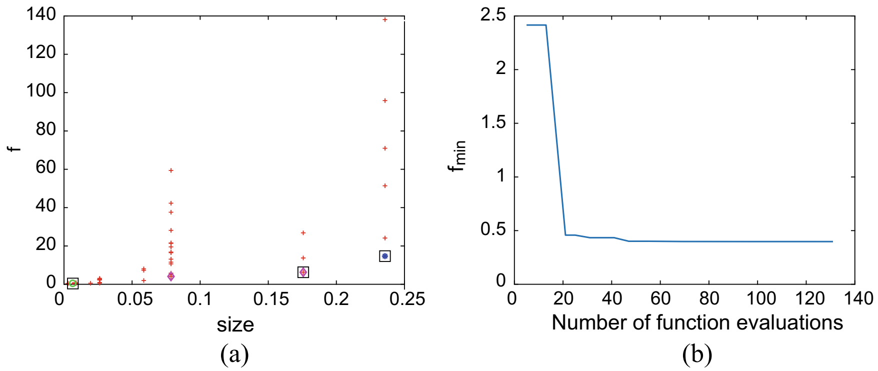
Firstly, we note that using the stopping condition (23.6) with a higher accuracy demand (0.01% and 0.0001%), the results favor Variant V_2. (This conclusion is different from what would be expected after the comparisons in Table 23.1.) In fact, when $\tau = 1E-04$, Variant V_2 is better, i.e., reaches the required accuracy with fewer function evaluations than Variant V_1 on 6 problems (out of 9) and is a tie in one problem. When a higher accuracy is demanded ($\tau = 1E-06$), Variant V_2 is still better on 7 problems.

When we compare the results of both variants of Algorithm 2 with DIRECT [1] and the solver RDIRECT-b [9], we may conclude that the results for a 0.01% accuracy is favorable to [9] on four problems, but is favorable to our algorithm on five problems. On the other hand, for a higher accuracy demand (0.0001%), the overall balance is six against three. From the comparison with the original DIRECT, we conclude that our algorithm wins (requires less function evaluations) for a 0.01% accuracy solution on five problems and wins for a 0.0001% accuracy solution on six problems. RDIRECT-b is a robust (insensitive to linear scaling of f) version of DIRECT with a bilevel strategy to accelerate convergence to a higher accurate result. The table also shows the results obtained by DIRECT-GL [8], that includes new strategies for the identification of an extended set of POH, a modified DIRECT version that uses an update to the condition (23.3) [14], and those reported in [10] of the two versions DISIMPL-V and DISIMPL-C of a DIRECT-like method that uses simplices instead of hyperrectangles. The first evaluates f at 2^n vertices and divides a simplex into two new simplices, the second evaluates f at $n!$ centroid points and divides a simplex into three new simplices. For a 0.01% accuracy solution, we may conclude that our algorithm outperforms DIRECT-GL [8] on seven (out of eight common problems), the modified DIRECT [14] on six (out of nine problems), the DISIMPL-V [10] on eight (out of nine problems), and the DISIMPL-C [10] also on eight problems.

Finally, we compare our results with three stochastic algorithms. In the directed tabu search with the adaptive pattern search in the intensification phase (DTS$_{APS}$) [15], the average number of function evaluations therein reported are related only to successful trials. For completeness, we also report the corresponding success rates (shown in the table as "% succ"). The other stochastic algorithm used in the comparison is the mutation-based artificial fish swarm algorithm (m-AFSA) [16]. It is a population based algorithm that uses a local search procedure to refine the search around the best point found so far. Another population-based algorithm is selected for the comparison. It uses a stochastic version of the coordinate descent method (St-Coord_D) [17] and the results are from the variant "hscore_w" with success rates of 100%. We may conclude that both variants of the Algorithm 2 outperform the three selected algorithms. Only for the problem SHU, DTS$_{APS}$ reaches the solution with the required accuracy in fewer function evaluations than our variants.

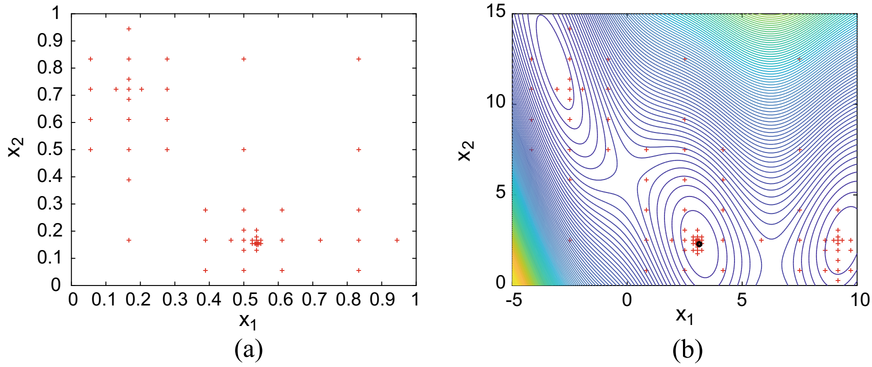
With Fig. 23.3a we aim to illustrate the influence of the heuristic coupled DIRECT on the selected *candidate* hyperrectangles and the POH, at iteration 8 of the global phase, when solving the problem BR, a two-dimensional problem with three global minima. As previously reported the 'green' *circles* correspond to the selected *candidate* hyperrectangles from the set I^3, the 'magenta' *diamonds* are from I^2, and the 'blue' '*' are from I^1. The 'black' *squares* mark the identified POH. Figure 23.3b displays the progress of $f_{\min}$ as the number of function evaluations increases, when solving the problem BR by Algorithm 2 with $G_{\max} = 10$ and $L_{\max} = 10$. The value of $f_{\min}$ rapidly drops (after 20 function evaluations) to a value near the global minimum (0.398).

Figure 23.4a shows the center points of the hyperrectangles generated at iteration 9 when Algorithm 2 uses $G_{\max} = 10$ and $L_{\max} = 10$ (corresponding to the Variant V_1) to solve the problem BR. Figure 23.4b shows the center points at the final iteration where the reported solution is within 0.01% of the global minimum (shown by a 'black' *full circle*). Similar information is shown in Fig. 23.5a, b, but now $G_{\max} = 10$ and $L_{\max} = 5$ (Variant V_2) are used instead. Finally, Fig. 23.6a, b show the center points of the generated hyperrectangles when $G_{\max} = 5$ and $L_{\max} = 10$ (Variant V_3). It can be seen that the points cluster around the three global solutions, being Variant V_2 the one that concentrates the search the most. After exploring the feasible region looking for promising regions, the Variant V_2 gathers around one of the global solutions instead of jumping and gathering around the other global optima.



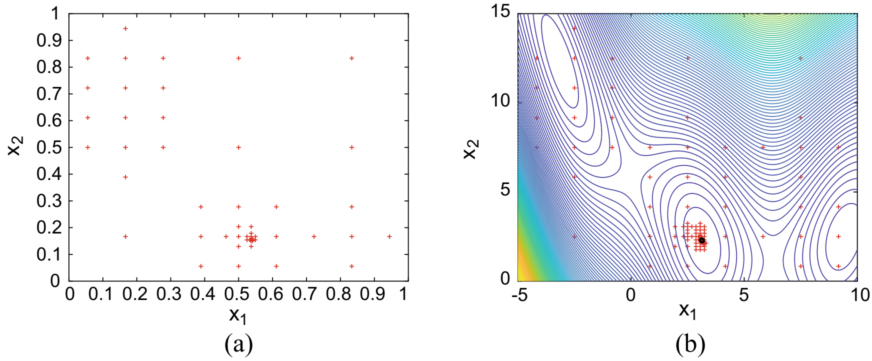
(a) (b)

Fig. 23.3 Solving the problem BR by Algorithm 2. **a** Center points of generated hyperrectangles, selected *candidate* hyperrectangles and identified POH. **b** Progress of $f_{\min}$



(a) (b)

Fig. 23.4 Generated hyperrectangles when solving the problem BR by Variant V_1 of Algorithm 2. **a** Iteration 9 (55 function evaluations). **b** Final iteration (131 function evaluations)



(a) (b)

Fig. 23.5 Generated hyperrectangles when solving the problem BR by Variant V_2 of Algorithm 2. **a** Iteration 9 (51 function evaluations). **b** Final iteration (137 function evaluations)

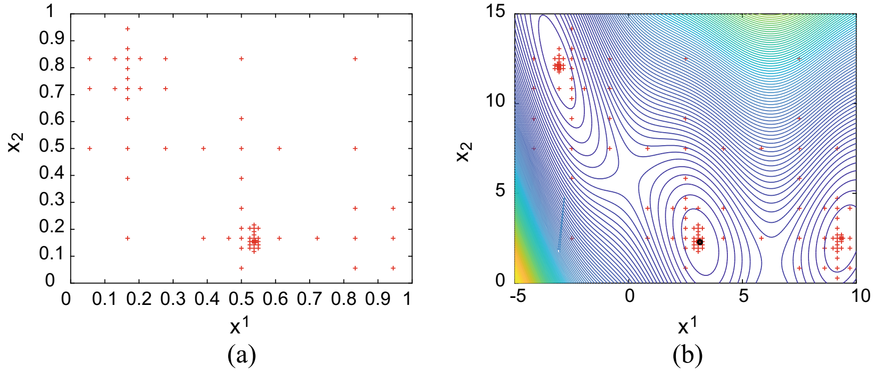


Fig. 23.6 Generated hyperrectangles when solving the problem BR by Variant V_3 of Algorithm 2. **a** Iteration 9 (69 function evaluations). **b** Final iteration (163 function evaluations)

23.4.3 Experiments with Larger Dimensional Problems

Another set of six larger dimensional benchmark problems from the Hedar test set [18] is also used for comparative purposes: Griewank (GW), Levy (L), Rastrigin (RG), Sphere (S), Sum squares (SS), Trid (T) (also available in [19]). We note that the search domain (S. Domain) was modified for some problems in order to avoid that the global minimum lies in the centroid of the feasible region [8, 9].

First, the problem SS is used to analyze the performance of the Variants V_1 and V_2 of the Algorithm 2, when compared to other DIRECT-type methods, as the number of variables increases. The maximum number of function evaluations is now set to $1E + 06$ and $\tau = 1E - 04$ in the stopping condition (23.6). See Table 23.3. The results are compared to those reported in [8], DIRECT, DIRECT-G (DIRECT with a strategy that globally enhances the set of POH), DIRECT-GL (DIRECT with strategies that globally and locally enhance the set of POH). Since numerical data for this problem are not available in [9], a direct comparison is not possible (marked as '–' in the table). (The authors use performance profiles to compare four DIRECT-type methods.) Between Variants V_1 and V_2, the latter is more efficient and from the results it can be concluded that the S. Domain affects the performance of the algorithm, in particular for the largest problem.

The number of function evaluations achieved by Variants V_1 and V_2 of Algorithm 2 when solving the problems GW, L, RG, S and T for $n = 10$ are shown in Table 23.4. Between the two tested variants, V_2 outperforms V_1 since it solves the largest problems in general with less function evaluations.

Table 23.3 Number of function evaluations required by Variants V_1 and V_2 to solve problem SS

Problem	S. Domain	Variant V_1	Variant V_2	DIRECT[a]	DIRECT-G[a]	DIRECT-GL[a]
SS $n = 2$	$[-10, 15]^n$	84	86	107	143	191
SS $n = 5$		2546	1765	833	1951	2919
SS $n = 10$		86122	29861	7795	16523	24763
SS $n = 2$	$[-8, 12.5]^n$	136	133	–	–	–
SS $n = 5$		3209	3135	–	–	–
SS $n = 10$		7695	5710	–	–	–

[a]Results reported in [8]

Table 23.4 Number of function evaluations of Variants V_1 and V_2 (problems with $n = 10$)

Problem	S. Domain	Variant V_1	Variant V_2	DIRECT[a]	DIRECT-G[a]	DIRECT-GL[a]
GW	$[-480, 750]^{10}$	14475	10389	–	–	–
L	$[-10, 10]^{10}$	70437	34067	5589	11149	16179
RG	$[-4.1, 6.4]^{10}$	524921	605391	–	–	–
S	$[-4.1, 6.4]^{10}$	192140	63155	–	–	–
T	$[-100, 100]^{10}$	77653	27075	$> 1E + 06$	$> 1E + 06$	115073

[a]Results reported in [8]; '–' Not available

23.5 Conclusions

The DIRECT method is coupled with a heuristic aiming to divide the region of promising hyperrectangles into three subregions for a discerned selection of a reduced number of hyperrectangles. Furthermore, a two-phase strategy that aims to cyclically encourage the global search capabilities (first phase) and enhance the local search (second phase) is implemented.

During the first phase, the heuristic DIRECT avoids the selection of the hyperrectangles that were mostly divided and chooses all the hyperrectangles with largest sizes. Conversely, during the second phase, the hyperrectangles with largest sizes are mostly avoided and the ones with smallest sizes are all included in the selection. The numerical experiments carried out until now show that a cycle of a global search phase of ten iterations and a local search phase of five iterations provides in general a more efficient process even when solving the largest dimensional problems.

Acknowledgments This work has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 734205—H2020-MSCA-RISE-2017 and by national funds through FCT Fundação para a Ciência e Tecnologia within the Projects Scope: UIDB/00319/2020 and UIDB/00013/2020 and UIDP/00013/2020 of CMAT-UM.

References

1. Jones DR, Perttunen CD, Stuckman BE (1993) Lipschitzian optimization without the Lipschitz constant. J Optim Theory Appl 79(1):157–181
2. Jones DR (2008) Direct global optimization algorithm. In: Floudas C, Pardalos P (eds) Encyclopedia of optimization. Springer, Boston MA, pp 431–440
3. Liuzzi G, Lucidi S, Piccialli V (2010) A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. Comput Optim Appl 45(1):353–375
4. Gablonsky JM, Kelley CT (2001) A locally-biased form of the DIRECT algorithm. J Global Optim 21(1):27–37
5. Liu Q, Zeng J (2015) Global optimization by multilevel partition. J Global Optim 61(1):47–69
6. Sergeyev YD, Kvasov DE (2006) Global search based on efficient diagonal partitions and a set of Lipschitz constants. SIAM J Optim 16(3):910–937
7. Paulavičius R, Sergeyev YD, Kvasov DE et al (2014) Globally-biased DISIMPL algorithm for expensive global optimization. J Global Optim 59(2–3):545–567
8. Stripinis L, Paulavičius R, Žilinskas J (2018) Improved scheme for selection of potentially optimal hyper-rectangles in DIRECT. Optim Lett 12(7):1699–1712
9. Liu Q, Cheng W (2014) A modified DIRECT algorithm with bilevel partition. J Global Optim 60(3):483–499
10. Paulavičius R, Žilinskas J (2014) Simplicial Lipschitz optimization without the Lipschitz constant. J Global Optim 59(1):23–40
11. Liu Q (2013) Linear scaling and the DIRECT algorithm. J Global Optim 56(3):1233–1245
12. Liu Q, Zeng J, Yang G (2015) MrDIRECT: a multilevel robust DIRECT algorithm for global optimization problems. J Global Optim 62(2):205–227
13. Liu H, Xu S, Chen X et al (2017) Constrained global optimization via a DIRECT-type constraint-handling technique and an adaptive metamodeling strategy. Struct Multi Optim 55(1):155–177
14. Finkel DE, Kelley CT (2006) Additive scaling and the DIRECT algorithm. J Global Optim 36(4):597–608
15. Hedar A-R, Fukushima M (2006) Tabu search directed by direct search methods for nonlinear global optimization. Eur J Oper Res 170(2):329–349
16. Rocha AMAC, Costa MFP, Fernandes EMGP (2011) Mutation-based artificial fish swarm algorithm for bound constrained global optimization. AIP Conf Proc 1389:751–754
17. Rocha AMAC, Costa MFP, Fernandes EMGP (2020) A population-based stochastic coordinate descent method. In: Le Thi H, Le H, Pham Dinh T (eds) Advances in intelligent systems and computing vol 991, Optimization of complex systems: theory, models, algorithms and applications. Springer, Berlin, pp 16–25
18. Hedar A-R http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm
19. Ali MM, Khompatraporn C, Zabinsky ZB (2005) A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J Global Optim 31(4):635–672

Chapter 24
A Multiple Shooting Descent-Based Filter Method for Optimal Control Problems

Gisela C. V. Ramadas, Edite M. G. P. Fernandes, Ana Maria A. C. Rocha, and M. Fernanda P. Costa

Abstract A direct multiple shooting (MS) method is implemented to solve optimal control problems (OCP) in the *Mayer form*. The use of an MS method gives rise to the so-called 'continuity conditions' that must be satisfied together with general algebraic equality and inequality constraints. The resulting finite nonlinear optimization problem is solved by a first-order descent method based on the filter methodology. In the equivalent tri-objective problem, the descent method aims to minimize the objective function, the violation of the 'continuity conditions' and the violation of the algebraic constraints simultaneously. The numerical experiments carried out with different types of benchmark OCP are encouraging.

Keywords Optimal control · Direct multiple shooting · Filter method · Descent directions

G. C. V. Ramadas (✉)
Research Center of Mechanical Engineering (CIDEM), School of Engineering of Porto (ISEP), Polytechnic of Porto, 4200-072 Porto, Portugal
e-mail: gcv@isep.ipp.pt

E. M. G. P. Fernandes
ALGORITMI Center, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
e-mail: emgpf@dps.uminho.pt

A. M. A. C. Rocha
ALGORITMI Center, Department of Production and Systems, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
e-mail: arocha@dps.uminho.pt

M. F. P. Costa
Centre of Mathematics, Department of Mathematics, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
e-mail: mfc@math.uminho.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_24

24.1 Introduction

An optimal control problem (OCP) is a constrained optimization problem that has a set of dynamic equations as constraints. Application domains of OCP are varied [1]. There are three types of OCP that differ in the formulation of the functional to be optimized. For example, an OCP of the *Lagrange form* has the objective functional in its pure integral form as shown

$$\begin{aligned} J^* = \min_{\mathbf{u}(t)\in U} \; & J(\mathbf{y}(t), \mathbf{u}(t)) \equiv \int_0^T f_2(t, \mathbf{y}(t), \mathbf{u}(t))\, dt \\ \text{s.t. } & \mathbf{y}'(t) = \mathbf{f}_1(t, \mathbf{y}(t), \mathbf{u}(t)), \quad \text{for } \; t \in [0, T] \\ & \mathbf{y}(0) = \mathbf{y}_0, \;\; \mathbf{y}(T) = \mathbf{y}_T \;, \end{aligned} \tag{24.1}$$

where $\mathbf{y} \in \mathbb{R}^{\bar{s}}$ is the vector of state variables of the dynamic system, $\mathbf{u} \in U \subset \mathbb{R}^c$ is the vector of control or input variables and U represents a class of functions (in particular functions of class C^1 and piecewise constant) and usually contains limitations to the control [2]. To convert problem (24.1) into the *Mayer form*, a new variable is added to the states vector $\mathbf{y}$, such that $y_s'(t) = f_2(t, \mathbf{y}(t), \mathbf{u}(t))$ with the initial condition $y_s(0) = 0$, where $s = \bar{s} + 1$ represents the total number of state variables. Thus, problem (24.1) becomes:

$$\begin{aligned} \min_{\mathbf{u}(t)\in U} \; & J(\mathbf{y}(t), \mathbf{u}(t)) \equiv y_s(T) \\ \text{s.t. } & \mathbf{y}'(t) = \mathbf{f}_1(t, \mathbf{y}(t), \mathbf{u}(t)) \\ & y_s'(t) = f_2(t, \mathbf{y}(t), \mathbf{u}(t)), \quad \text{for } \; t \in [0, T] \\ & \mathbf{y}(0) = \mathbf{y}_0, \;\; y_s(0) = 0, \;\; \mathbf{y}(T) = \mathbf{y}_T \;. \end{aligned} \tag{24.2}$$

In the OCP we want to find $\mathbf{u}$ that minimizes the objective functional J subject to the dynamic system of ordinary differential equations (ODE). The problem may have other more complex 'terminal constraints' $H(T, \mathbf{y}(T), \mathbf{u}(T)) = 0$. States $\mathbf{y}$ and control $\mathbf{u}$ may also be constrained by algebraic equation constraints $h_e(t, \mathbf{y}(t), \mathbf{u}(t)) = 0$, $e \in E$ and 'path constraints' $g_j(t, \mathbf{y}(t), \mathbf{u}(t)) \leq 0$, $j \in F$, where $E = \{1, 2, \ldots, m\}$ and $F = \{1, 2, \ldots, l\}$.

Methods for solving OCP like (24.2) can be classified into indirect and direct methods. Indirect methods use the first-order necessary conditions from Pontryagin's maximum principle to reformulate the original problem into a boundary value problem. On the other hand, direct methods solve the OCP directly [3] transforming the infinite-dimensional OCP into a finite-dimensional optimization problem that can be solved by effective and well-established nonlinear programming (NLP) algorithms. All direct methods discretize the control variables but differ in the way they treat the state variables [4]. They are also classified as *Discretize then Optimize* strategies in contrast to the *Optimize then Discretize* strategies of the indirect methods [1].

This paper explores the use of a first-order descent method based on the filter methodology [5, 6] to solve the NLP problem, within a direct method for solving an OCP in the *Mayer form*. The use of a direct multiple shooting (MS) method gives rise to the so-called ‘continuity conditions’ that must be satisfied. The novelty here is that a filter methodology is used to minimize the objective function, the violation of the ‘continuity conditions’ and the violation of algebraic constraints simultaneously. The NLP problem is a tri-objective problem and the first-order descent method generates a search direction that is either the negative gradient of one of the functions to be minimized or a convex combination of negative gradients of two functions. To overcome the drawback of computing first derivatives, the gradients are approximated by finite differences.

The paper is organized as follows. Section 24.2 briefly describes the direct MS algorithm for solving the OCP in the *Mayer form*. The herein proposed first-order descent filter algorithm is discussed in Sect. 24.3, the numerical experiments are shown in Sect. 24.4 and we conclude the paper with Sect. 24.5.

24.2 Direct Multiple Shooting Method

In a direct single shooting (SS) method, only the controls are discretized in the NLP problem [3]. The dynamic system is solved by an ODE solver to get the state values for the optimization. Thus, simulation and optimization are carried out sequentially. On a specific grid defined by $0 = t_1 < t_2 < \cdots < t_{N-1} < t_N = T$, where $N - 1$ is the total number of subintervals, the control $\mathbf{u}(t)$ is discretized, namely using piecewise polynomial approximations. The simplest of all is a piecewise constant, $\mathbf{u}(t) = \mathbf{q}^i, \quad \text{for} \quad t \in [t_i, t_{i+1}]$ and $i = 1, \ldots, N - 1$ so that $\mathbf{u}(t)$ only depends on the control parameters $\mathbf{q} = (\mathbf{q}^1, \mathbf{q}^2, \ldots, \mathbf{q}^{N-1})$ and $\mathbf{u}(t) = \mathbf{u}(t, \mathbf{q})$. When the horizon length T is not fixed, the control parameter vector also includes T to define the optimization variables. The dynamic system is solved by (forward numerical integration) an ODE solver and the state variables $\mathbf{y}(t)$ are considered as dependent variables $\mathbf{y}(t, \mathbf{q})$. The main advantage of a direct SS method is the reduced number of decision variables (control parameters) in the NLP even for very large dynamic systems. However, unstable systems may be difficult to handle.

In a direct MS method, discretized controls and state values at the start nodes of the grid (grid points)—$\mathbf{x}^i \in \mathbb{R}^s, i = 1, 2, \ldots, N - 1$, known as MS node variables—are the decision variables for the NLP solver [7]. After the discretization of the controls, the ODE system is solved on each shooting subinterval $[t_i, t_{i+1}]$ independently, but they need to be linked by the auxiliary variables $\mathbf{x}^i, i = 1, 2, \ldots, N - 1$. They are the initial values for the state variables for the $N - 1$ independent initial value problems on the subintervals $[t_i, t_{i+1}]$:

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{q}^i) \equiv \begin{cases} \mathbf{f}_1(t, \mathbf{y}(t), \mathbf{q}^i) \\ f_2(t, \mathbf{y}(t), \mathbf{q}^i) \end{cases} \text{ with } \mathbf{y}(t_i) = \mathbf{x}^i, \text{ for } t \in [t_i, t_{i+1}] \; ,$$

where $\mathbf{y} \in \mathbb{R}^s$. Trajectories $\mathbf{y}^i(t; \mathbf{x}^i, \mathbf{q}^i)$ are obtained where the notation “$(t; \mathbf{x}^i, \mathbf{q}^i)$”, for the argument, means that they are dependent on t as well as on the specified values for the node variables $\mathbf{x}^i$ and control parameters. The initial state values $\mathbf{x}^i$ should satisfy the 'continuity conditions'

$$\mathbf{y}^i(t_{i+1}; \mathbf{x}^i, \mathbf{q}^i) = \mathbf{x}^{i+1}, \; i = 1, \ldots, N-1 \; , \tag{24.3}$$

(ensuring continuity of the solution trajectory), the initial value $\mathbf{x}^1 = \mathbf{y}_0$ and the final state constraints $\mathbf{x}^N = \mathbf{y}_T$ [4, 8].

We choose to implement a direct MS method since it can cope with differential and algebraic equations that show unstable dynamical behavior [7]. The main steps of the direct MS algorithm are shown in Algorithm 1.

```
Input: $T$, $N$, $\mathbf{f}(t, \mathbf{y}, \mathbf{u})$, $\mathbf{y}_0$, $\mathbf{y}_T$, constraint functions.
Output: Optimal control and state variables.
Define the grid points in the interval $[0, T]$: $0 = t_1 < \cdots < t_{N-1} < t_N = T$.
Discretize the control: $\mathbf{u}(t) = \mathbf{q}^i$ for $t \in [t_i, t_{i+1}]$, $i = 1, \ldots, N-1$.
Define the starting values for the state vector $\mathbf{x}^i$ for each $[t_i, t_{i+1}]$, $i = 1, \ldots, N-1$, and $\mathbf{x}^N$.
(Invoke the NLP algorithm)
while Stopping conditions are not satisfied do
    With $\mathbf{q}^i, i = 1, \ldots, N-1$, $\mathbf{x}^i, i = 1, \ldots, N$, use an ODE solver to evaluate the state
      trajectories in $[t_i, t_{i+1}]$, $i = 1, \ldots, N-1$:
    for $\mathbf{y}^i(t_i) = \mathbf{x}^i$, $(\mathbf{y}^i)'(t) = \mathbf{f}(t, \mathbf{y}^i(t), \mathbf{q}^i)$;
    Evaluate the 'continuity conditions' $\mathbf{y}^i(t_{i+1}; \mathbf{x}^i, \mathbf{q}^i) = \mathbf{x}^{i+1}$, $i = 1, \ldots, N-1$, as well as
      $\mathbf{x}^1 = \mathbf{y}_0$ and $\mathbf{x}^N = \mathbf{y}_T$;
    Evaluate algebraic equality and inequality constraints for $t \in [t_i, t_{i+1}]$, $i = 1, \ldots, N-1$;
    Evaluate the objective function;
    Generate new $\mathbf{q}^i$, $i = 1, \ldots, N-1$ and $\mathbf{x}^i$, $i = 1, \ldots, N$.
end
```

Algorithm 1: Direct MS algorithm

24.3 First-Order Descent Filter Method

The herein proposed first-order descent filter method relies on descent directions for two constraint violation functions (handled separately) and for the objective function in order to converge towards the optimal solution of the NLP problem. One of the constraint violation functions emerges from the 'continuity constraints' violation (including initial state and final state constraints) and the other comes up from the state and control algebraic equality and inequality constraints. We assume that the NLP problem is a non-convex constrained optimization problem (COP). For practical purposes, we assume that the OCP is in the *Mayer form*, the ODE system has initial and boundary state values, state and control variables are constrained by algebraic equality and inequality constraints, and the explicit 4th. order Runge-Kutta integration formula is used to solve the dynamic system in each subinterval $[t_i, t_{i+1}]$ using 5 points.

As stated in the last section, the decision variables of the COP are the initial state values at the nodes $\mathbf{x}^i \in \mathbb{R}^s$, $i = 1, \ldots, N$ and the control variables $\mathbf{q}^i \in \mathbb{R}^c$,

$i = 1, \dots, N-1$. Besides possible algebraic constraints on the state and control variables, the 'continuity constraints' (24.3), the initial state and the final state constraints must be added to the optimization problem formulation. Thus, our COP has the following form:

$$\begin{array}{ll} \displaystyle\min_{\mathbf{x}^i,\, i\in I_N;\mathbf{q}^i,\, i\in I} & y_s(T) \\ \text{s.t.} & g_j(\mathbf{y}^i(t;\mathbf{x}^i,\mathbf{q}^i),\mathbf{q}^i) \leq 0,\ t\in[t_i,t_{i+1}], i\in I, j\in F \\ & h_e(\mathbf{y}^i(t;\mathbf{x}^i,\mathbf{q}^i),\mathbf{q}^i) = 0,\ t\in[t_i,t_{i+1}], i\in I, e\in E \\ & \mathbf{y}^i(t_{i+1};\mathbf{x}^i,\mathbf{q}^i) - \mathbf{x}^{i+1} = 0, i\in I \\ & \mathbf{x}^1 - \mathbf{y}_0 = 0, \mathbf{x}^N - \mathbf{y}_T = 0\ , \end{array} \tag{24.4}$$

where $I = \{1, \dots, N-1\}$ and $I_N = I \cup \{N\}$. To solve the optimization problem (24.4), the set of ODE must be solved so that the 'continuity constraints' $\mathbf{y}^i(t_{i+1};\mathbf{x}^i,\mathbf{q}^i) - \mathbf{x}^{i+1} = 0$, the initial state and the final state constraints, the other equality and inequality constraints and the objective function are evaluated (see Algorithm 1). Since problem (24.4) has constraints, we seek optimal values for $\mathbf{x}$ and $\mathbf{q}$ such that all the constraints are satisfied—a feasible solution of the COP—and the objective function takes the least value.

24.3.1 Filter Methodology

To check solution feasibility, a measure for the violation of the constraints is adopted. To implement the herein proposed filter methodology, the constraints are fractionated into two sets and their violations are computed and handled separately. We denote the violation of the 'continuity constraints', initial state and final state constraints by the non-negative function:

$$\theta(\mathbf{x},\mathbf{q}) = \sum_{l\in L}\sum_{i\in I}(y_l^i(t_{i+1};\mathbf{x}^i,\mathbf{q}^i) - x_l^{i+1})^2 + \sum_{l\in L}(x_l^1 - y_{l_0})^2 + \sum_{l\in L}(x_l^N - y_{l_T})^2\ , \tag{24.5}$$

where $L = \{1, 2, \dots, s\}$, noting that $\theta(\mathbf{x},\mathbf{q})$ is zero if the solution $(\mathbf{x},\mathbf{q})$ satisfies these constraints, and is positive otherwise. These are the constraints that are more difficult to be satisfied and we need to priority drive the violation θ to zero as soon as possible so that the ODE integration runs as close as possible to the exact values of the state variables.

To evaluate the algebraic equality and inequality constraints violation, a non-negative function p, also based on the Euclidean norm of vectors, is used

$$p(\mathbf{x},\mathbf{q}) = \sum_{j\in F}\sum_{i\in I}\max\left\{0, g_j(\mathbf{y}^i(t;\mathbf{x}^i,\mathbf{q}^i),\mathbf{q}^i)\right\}^2 + \sum_{e\in E}\sum_{i\in I}h_e(\mathbf{y}^i(t;\mathbf{x}^i,\mathbf{q}^i),\mathbf{q}^i)^2, \tag{24.6}$$

and similarly, $p(\mathbf{x}, \mathbf{q}) = 0$ when the corresponding constraints are satisfied, and $p(\mathbf{x}, \mathbf{q}) > 0$ otherwise. The violation of these constraints is also forced to converge to zero.

The extension of the filter methodology [5] into the descent algorithm to solve the COP is equivalent to the reformulation of the problem (24.4) as a tri-objective optimization problem that aims to minimize both the feasibility measures, defined by the constraint violation functions $\theta(\mathbf{x}, \mathbf{q})$ and $p(\mathbf{x}, \mathbf{q})$, and the optimality measure defined by the objective function $y_s(T)$:

$$\min_{\mathbf{x}^i,\, i \in I_N;\mathbf{q}^i,\, i \in I} \left(\theta(\mathbf{x}, \mathbf{q}), p(\mathbf{x}, \mathbf{q}), y_s(T)\right) . \tag{24.7}$$

In our filter methodology, a *filter* $\mathscr{F}$ is a finite set of triples $(\theta(\mathbf{x}, \mathbf{q}), p(\mathbf{x}, \mathbf{q}), y_s(T))$ that correspond to points $(\mathbf{x}, \mathbf{q})$, none of which is dominated by any of the others in the *filter*. A point $(\hat{\mathbf{x}}, \hat{\mathbf{q}})$ is said to dominate a point $(\mathbf{x}, \mathbf{q})$ if and only if the following conditions are satisfied simultaneously:

$$\theta(\hat{\mathbf{x}}, \hat{\mathbf{q}}) \le \theta(\mathbf{x}, \mathbf{q}), \;\; p(\hat{\mathbf{x}}, \hat{\mathbf{q}}) \le p(\mathbf{x}, \mathbf{q}) \;\text{ and }\; \hat{y}_s(T) \le y_s(T) \;,$$

with at least one inequality being strict. The *filter* is initialized to $\mathscr{F} = \{(\theta, p, y_s) : \theta \ge \theta_{\max}, p \ge p_{\max}\}$, where $\theta_{\max}, p_{\max} > 0$ are upper bounds on the acceptable constraint violations. Let $\mathscr{F}_k$ be the *filter* at iteration k of the algorithm. To avoid the acceptance of a trial point $(\bar{\mathbf{x}}, \bar{\mathbf{q}})$ (approximation to the optimal solution), or the corresponding triple $(\theta(\bar{\mathbf{x}}, \bar{\mathbf{q}}), p(\bar{\mathbf{x}}, \bar{\mathbf{q}}), \bar{y}_s(T))$, that is arbitrary close to the boundary of the *filter*, the conditions of acceptability to the *filter* define an envelope around the filter and are as follows:

$$\begin{array}{l} \theta(\bar{\mathbf{x}}, \bar{\mathbf{q}}) \le (1-\gamma)\theta(\mathbf{x}^{(l)}, \mathbf{q}^{(l)}) \;\text{ or }\; p(\bar{\mathbf{x}}, \bar{\mathbf{q}}) < (1-\gamma)p(\mathbf{x}^{(l)}, \mathbf{q}^{(l)}) \\ \;\;\text{or }\; \bar{y}_s(T) \le y_s^{(l)}(T) - \gamma\left(\theta(\mathbf{x}^{(l)}, \mathbf{q}^{(l)}) + p(\mathbf{x}^{(l)}, \mathbf{q}^{(l)})\right) \end{array} \tag{24.8}$$

for all points $(\mathbf{x}^{(l)}, \mathbf{q}^{(l)})$ that correspond to triples $(\theta(\mathbf{x}^{(l)}, \mathbf{q}^{(l)}), p(\mathbf{x}^{(l)}, \mathbf{q}^{(l)}), y_s^{(l)}(T))$ in the *filter* $\mathscr{F}_k$. Points with constraint violations that exceed $\theta_{\max}$ or $p_{\max}$ are not acceptable. The constant $\gamma \in (0, 1)$ is fixed and the smaller the tighter is the envelope of acceptability. The above conditions impose a sufficient reduction on one of the feasibility measures or on the optimality measure for a point to be acceptable. When the point is acceptable to the *filter*, the *filter* is updated and whenever a point is added to the *filter*, all the dominated points are removed from it.

24.3.2 The First-Order Descent Filter Algorithm

The proposed first-order descent method is based on using gradient approximations of the functions, θ, p or y_s, of the tri-objective problem (24.7), to define search directions coupled with a simple line search to compute a step size that gives a simple decrease on one of the measures θ, p or y_s. Since θ is the most difficult to reduce, priority is given to searching along the (negative) gradient of θ or a (negative) combination of the gradient of θ with the gradient of p or y_s. See Algorithm 2. For easy of notation $\mathbf{v} = \left(x_1^1, \ldots, x_s^1, \ldots, x_1^N, \ldots, x_s^N, q_1^1, \ldots, q_c^1, \ldots, q_1^{N-1}, \ldots, q_c^{N-1}\right)^T$ is used to denote the vector of the decision variables ($\mathbf{v} \in \mathbb{R}^{n_D}$, $n_D = Ns + (N-1)c$).

Each component i of the gradient of θ with respect to the variable v_i, at an iteration k, is approximated by

$$\nabla_i \theta(\mathbf{v}^{(k)}) \approx \left(\theta(\mathbf{v}^{(k)} + \varepsilon \mathbf{e}_i) - \theta(\mathbf{v}^{(k)})\right) / \varepsilon \ , \quad i = 1, 2, \ldots, n_D \qquad (24.9)$$

for a positive and sufficiently small constant ε, being the vector $\mathbf{e}_i \in \mathbb{R}^{n_D}$ the i column of the identity matrix. Similarly for the gradients approximation of p and y_s.

To identify the best point computed so far, the below conditions (24.10) are imposed. Let $\mathbf{v}^{best}$ be the current best approximation to the optimal solution of problem (24.7). A trial point, $\bar{\mathbf{v}}$, will be the best point computed so far (replacing the current $\mathbf{v}^{best}$) if one of the conditions

$$\Theta(\bar{\mathbf{v}}) < \Theta(\mathbf{v}^{best}) \text{ or } \bar{y}_s(T) < y_s^{best}(T) \qquad (24.10)$$

holds, where $\Theta = \theta + p$. At each iteration, the algorithm computes a trial point $\bar{\mathbf{v}}$, approximation to the optimal solution, by searching along a direction that is the negative gradient of θ, or a negative convex combination of the gradients of θ and p, θ and y_s, or p and y_s, at the current approximation $\mathbf{v}$. The selected direction depends on information related to the magnitude of θ and p, at $\mathbf{v}$. For example, if $p(\mathbf{v})$ is considered sufficiently small, i.e., $0 \le p(\mathbf{v}) \le \eta_1$, while $\theta(\mathbf{v}) > \eta_1$ (for a small error tolerance $\eta_1 > 0$), then the direction is the negative gradient of θ at $\mathbf{v}$. The search for a step size $\alpha \in (0, 1]$ goals the reduction of θ ('$M \leftarrow \theta$' in Algorithm 2). On the other hand, if both p and θ are considered sufficiently small, then the direction is the negative convex combination of the gradients of θ and y_s, although the search for α forces the reduction on θ.

If both θ and p are not small yet (situation that occurs during the initial iterations) the direction is along the negative convex combination of the gradients of θ and p, although the line search forces the reduction on θ. However, if $0 \le \theta(\mathbf{v}) \le \eta_1$ but $p(\mathbf{v}) > \eta_1$, then the direction is along the negative convex combination of the gradients of p and y_s and the line search forces the reduction on p. Further details are shown in the Algorithm 2.

The new trial point is accepted for further improvement if it satisfies the conditions to be acceptable to the current filter (see conditions (24.8)), although each trial point is considered as a new approximation to the optimal solution only if it is better than the previously saved best point, according to (24.10). In this situation, a new *outer* iteration—indexed by k in Algorithm 2—is carried out unless the convergence conditions are satisfied (see (24.11) below). If the trial point is accepted but it does not satisfy (24.10), θ, p and y_s are evaluated at the trial point and a new *inner* iteration—indexed by It—is carried out. This *inner* iterative process runs for a maximum of $It_{\max}$ iterations.

```
Input: $N, T, k_{\max} > 0, It_{\max} > 0, \eta_1 > 0$
Output: $\mathbf{v}^{best}, \theta^{best}, p^{best}, y_s^{best}$
Set $k = 0$, exit = “false”; Initialize $\mathscr{F}$;
Set initial $\mathbf{v}$;
Compute $\theta = \theta(\mathbf{v})$, $p = p(\mathbf{v})$, $y_s = y_s(T)$; Update $\mathscr{F}$;
Set $\mathbf{v}^{best} = \mathbf{v}$, $\theta^{best} = \theta$, $p^{best} = p$, $y_s^{best} = y_s$;
while $k < k_{\max}$ and exit = “false” do
    Set $k = k + 1$, $It = 0$, $it_{no} = 0$, accept = “true”, stop = “false”;
    while $It < It_{\max}$ and stop = “false” do
        Set $It = It + 1$, $F_{It} = It/It_{\max}$;
        Compute $\mathbf{G}_\theta \approx \nabla\theta(\mathbf{v})$, $\mathbf{G}_p \approx \nabla p(\mathbf{v})$, $\mathbf{G}_{y_s} \approx \nabla y_s(T)$ using (24.9);
        if accept = “true” then
            if $\theta \le \eta_1$ and $p \le \eta_1$ then
                Set $\mathbf{G} = (1 - F_{It})\mathbf{G}_\theta + F_{It}\mathbf{G}_{y_s}$; $M \leftarrow \theta$;
            else
                if $p \le \eta_1$ and $\theta > \eta_1$ then
                    Set $\mathbf{G} = \mathbf{G}_\theta$; $M \leftarrow \theta$;
                else
                    if $\theta \le \eta_1$ and $p > \eta_1$ then
                        Set $\mathbf{G} = (1 - F_{It})\mathbf{G}_p + F_{It}\mathbf{G}_{y_s}$; $M \leftarrow p$;
                    else
                        Set $\mathbf{G} = (1 - F_{It})\mathbf{G}_\theta + F_{It}\mathbf{G}_p$; $M \leftarrow \theta$;
                    end
                end
            end
        else
            Set $it_{no} = it_{no} + 1$;
            if $it_{no} < (It_{\max} - 1)$ then
                Set $\mathbf{G} = (1 - F_{It})\mathbf{G}_\theta + F_{It}\mathbf{G}_p$; $M \leftarrow \theta$;
            else
                Set $\mathbf{G} = (1 - F_{It})\mathbf{G}_{y_s} + F_{It}\mathbf{G}_\theta$; $M \leftarrow y_s$;
            end
        end
        Compute $\alpha \in (0, 1]$ such that $M(\mathbf{v} - \alpha\mathbf{G}) < M(\mathbf{v})$; Set
          $\bar{\mathbf{v}} = \mathbf{v} - \alpha\mathbf{G}$, $\bar{\theta} = \theta(\bar{\mathbf{v}})$, $\bar{p} = p(\bar{\mathbf{v}})$, $\bar{y}_s = \bar{y}_s(T)$;
        if $\bar{\mathbf{v}}$ is acceptable to filter (according to (24.8)) then
            Set $\mathbf{v} = \bar{\mathbf{v}}$, $\theta = \bar{\theta}$, $p = \bar{p}$, $y_s = \bar{y}_s$;
            Set accept = “true”; Update $\mathscr{F}$;
            if $\bar{\mathbf{v}}$ is the best computed so far (see (24.10)) then
                $\mathbf{v}^{best} = \bar{\mathbf{v}}$, $\theta^{best} = \bar{\theta}$, $p^{best} = \bar{p}$, $y_s^{best} = \bar{y}_s$;
                if convergence conditions (24.11) are satisfied then
                    Set stop = “true”, exit = “true” (convergence);
                end
                Set stop = “true”;
            end
        else
            Set accept = “false”;
        end
    end
end
```

Algorithm 2: Descent-filter algorithm

The trial point might not be acceptable to the *filter*, in which case another *inner* iteration is tried. If the number of iterations with non acceptable trial points reaches $It_{\max}$, the new direction is along the negative convex combination of the gradients of θ and y_s (with a reduction on y_s in the line search); otherwise, the negative convex combination of the gradients of θ and p (with a reduction on θ in the line search) is tested.

The convergence conditions are said to be satisfied at a new trial point—the best point computed so far, $\mathbf{v}^{best}$,—if

$$\theta(\mathbf{v}^{best}) < \eta_1 \text{ and } p(\mathbf{v}^{best}) < \eta_1 \text{ and } perror = \left(\left|y_s^{best} - y_s^{pr.best}\right| / \left|y_s^{best}\right|\right) < \eta_2, \tag{24.11}$$

for small error tolerances $\eta_1 > 0$ and $\eta_2 > 0$, where the superscript *pr.best* refers to the previous best point. The *outer* iterative process also terminates if the number of iterations exceeds $k_{\max}$.

24.4 Numerical Experiments

The new direct MS method based on descent directions and the filter methodology has been tested with seven OCP. The MATLAB® (MATLAB is a registered trademark of the MathWorks, Inc.) programming language is used to code the algorithm and the tested problems. The numerical experiments were carried out on a PC Intel Core i7–7500U with 2.7 GHz, 256 Gb SSD and 16 Gb of memory RAM. The values set to the parameters are shown in Table 24.1.

First, three problems with free terminal time T are solved. A simple approach is to apply the change of variable $t = T\tau$, (with $dt = Td\tau$) which transforms the problem into a fixed boundary problem on the interval [0, 1] and treats T as an auxiliary variable. When the objective is to minimize T, an alternative is to add a new variable to the states vector $\mathbf{y} \in \mathbb{R}^{s-1}$ such that $y_s'(t) = 1$, with initial value $y_s(0) = 0$.

Problem 24.1 A simple car model (*Dubins car*) is formulated with three degrees of freedom where the car is imagined as a rigid body that moves in a plane [2]. The position of the car is given by (x, y, β) where x and y are the directions and β is the angle with the X axis. The problem is to drive in minimum time the car from a position to the origin:

Table 24.1 Parameter values

Parameter	Value	Parameter	Value
$\theta_{\max}$	$1E{+}03\,\theta(\mathbf{v}^{(0)})$	η_1	$1E{-}04$
$p_{\max}$	$1E{+}03\max\{p(\mathbf{v}^{(0)}), 1\}$	η_2	$1E{-}03$
γ	$1E{-}05$	$k_{\max}$	750
ε	$1E{-}06$	$It_{\max}$	s

$$\begin{aligned}
\min_{u(t)} \;& J(x(t), y(t), \beta(t), u(t)) \equiv T \\
\text{s.t. } & x'(t) = \cos(\beta(t)) \\
& y'(t) = \sin(\beta(t)) \\
& \beta'(t) = u(t), \quad t \in [0, T] \\
& x(0) = 4,\; y(0) = 0,\; \beta(0) = \tfrac{\pi}{2},\; x(T) = 0,\; y(T) = 0, \\
& |u(t)| \leq 2, \quad t \in [0, T]\;.
\end{aligned}$$

The results from both strategies to handle T free are shown in Table 24.2. The initial guesses were $x(t_i) = 2,\; y(t_i) = 0,\; \beta(t_i) = 1, i \in I_N$ and $u(t_i) = 0, i \in I$. The number of points considered in $[0, T]$ is 11. The table shows the values of J, θ and p achieved at iteration k, as well as the number of function evaluations, nfe, and the time in seconds, $time$. Optimal solution reported [2] is $J^* = 4.32174$. The results are considered quite satisfactory. We show in Fig. 24.1a, b the optimal states trajectory and control respectively, obtained from the run that considers the change of variable $t \to \tau$. Figure 24.1c displays the optimal control required to achieve identical states trajectory from the run that adds a new state variable. Slightly different optimal controls were obtained to reach identical states trajectory.

Problem 24.2 The resource allocation problem (*R allocation*) goals the assignment of resources in minimum time [2]:

$$\begin{aligned}
\min_{u(t)} \;& J(\mathbf{y}(t), \mathbf{u}(t)) \equiv T \\
\text{s.t. } & y_1'(t) = u_1(t) y_1(t) y_2(t) \\
& y_2'(t) = u_2(t) y_1(t) y_2(t), \quad t \in [0, T] \\
& y_1(0) = 1,\; y_2(0) = 2,\; y_1(T) y_2(T) = 10, \\
& y_1(t) \geq 0, y_2(t) \geq 0,\; u_1(t) + u_2(t) = 1,\; u_1(t) \geq 0,\; u_2(t) \geq 0,\; t \in [0, T]\;.
\end{aligned}$$

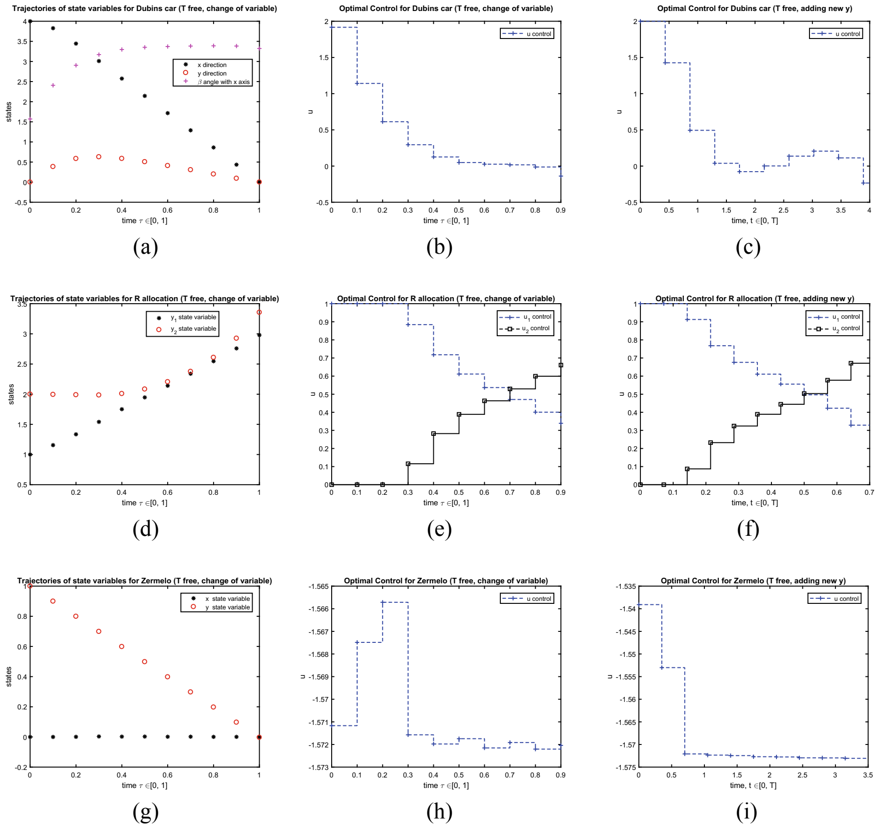
Since $u_2 = 1 - u_1$ the control vector can be reduced to a scalar $u_1 \equiv u \in [0, 1]$. Using the initial guesses $y_1(t_i) = 1,\; y_2(t_i) = 0, i \in I_N,\; u(t_i) = 0, i \in I$ and $N = 11$, the results are shown in Table 24.2. Optimal solution reported [2] is $J^* = 0.714118$. Figures 24.1d, e show the optimal states $y_1,\; y_2$ and control $u_1,\; u_2$ respectively, for the case where a change of variable is applied. Figure 24.1f shows the control for the case of handling T free through the adding of a new state variable. The states trajectory are similar to Fig. 24.1d.

Problem 24.3 Consider an unmanned aerial vehicle (*Zermelo*) flying in a horizontal plane with constant speed V, although the heading angle $u(t)$ (control input) (with respect to the X axis) can be varied. Winds are assumed to be in the Y direction with speed w. The objective is to fly from point A = (0, 1) to B = (0, 0) in minimum time:

$$\begin{aligned}
\min_{u(t)} \;& J(x(t), y(t), u(t)) \equiv T \\
\text{s.t. } & x'(t) = V\cos(u(t)) \\
& y'(t) = V\sin(u(t)) + w, \quad t \in [0, T] \\
& x(0) = 0,\; y(0) = 1,\; x(T) = 0,\; y(T) = 0 \\
& |u(t)| \leq \pi/2, \quad t \in [0, T]\;.
\end{aligned}$$

Table 24.2 Results obtained for the Problems 24.1, 24.2 and 24.3

Problem	Handling T	k	J	θ	p	nfe	$time$
Dubins car		1	4.7539	$2.7003E{+}01$	$0.0000E{+}00$		
	Adding new y_s	388	4.3329	$9.9908E{-}05$	$0.0000E{+}00$	44255	50.0
	Change of variable	192	4.3658	$9.6149E{-}05$	$0.0000E{+}00$	18044	19.8
R allocation		1	0.5714	$1.0484E{+}02$	$0.0000E{+}00$		
	Adding new y_s	472	0.7219	$9.7083E{-}05$	$0.0000E{+}00$	44486	48.9
	Change of variable	638	0.7232	$9.9168E{-}05$	$0.0000E{+}00$	47223	49.7
Zermelo		1	3.8500	$1.3863E{+}01$	$0.0000E{+}00$		
	Adding new y_s	323	3.5143	$9.6343E{-}05$	$2.8735E{-}05$	29595	32.3
	Change of variable	644	3.5249	$9.9618E{-}05$	$9.6530E{-}06$	46160	48.2



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 24.1 **a** States trajectory for *Dubins car*. **b** Optimal control for *Dubins car*. **c** Optimal control for *Dubins car* (when adding new y_s). **d** States trajectory for *R allocation*. **e** Optimal control for *R allocation*. **f** Optimal control for *R allocation* (when adding new y_s). **g** States trajectory for *Zermelo*. **h** Optimal control for *Zermelo*. **i** Optimal control for *Zermelo* (when adding new y_s)

For $V = 1$, $w = 1/\sqrt{2}$ and using the initial guesses $x(t_i) = 0$, $y(t_i) = 1, i \in I_N$ and $u(t_i) = 0, i \in I$, the results are shown in Table 24.2 for $N = 11$. A value near $T = 3.5$ is exhibited in [9]. The optimal states x, y and control u (from the run based on the change of variable $T \to \tau$) are shown in Fig. 24.1g, h respectively. Figure 24.1i presents the optimal control obtained from the run that adds a new variable to the states vector.

The next three problems are OCP of the *Lagrange form* and the last problem is already in the *Mayer form*.

Problem 24.4 In a continuous stirred-tank chemical reactor (*Tank reactor*), y_1 represents the deviation from the steady-state temperature, y_2 represents the deviation from the steady-state concentration and u is the effect of the coolant flow on the chemical reaction [10]:

$$\begin{aligned}
\min_{u(t)} \; & J \equiv \int_0^T (y_1(t)^2 + y_2(t)^2 + Ru(t)^2)\, dt \\
\text{s.t. } & y_1'(t) = -2(y_1(t) + 0.25) + (y_2(t) + 0.5) \exp\left(\tfrac{25 y_1(t)}{y_1(t)+2}\right) \\
& -(y_1(t) + 0.25)u(t) \\
& y_2'(t) = 0.5 - y_2(t) - (y_2(t) + 0.5) \exp\left(\tfrac{25 y_1(t)}{y_1(t)+2}\right), \quad t \in [0, T] \\
& y_1(0) = 0.05, \;\; y_2(0) = 0 \;.
\end{aligned}$$

The optimal solution reported in [10], for $T = 0.78$ and $R = 0.1$, is $J^* = 0.0268$. Using the initial guesses $y_1(t_i) = 0.05,\; y_2(t_i) = 0, i \in I_N$ and $u(t_i) = 0.75, i \in I$, with $N = 11$, the results are shown in Table 24.3. The proposed strategy has produced again a reasonably good solution. Figures 24.2a, b show the optimal states y_1, y_2 and control u respectively.

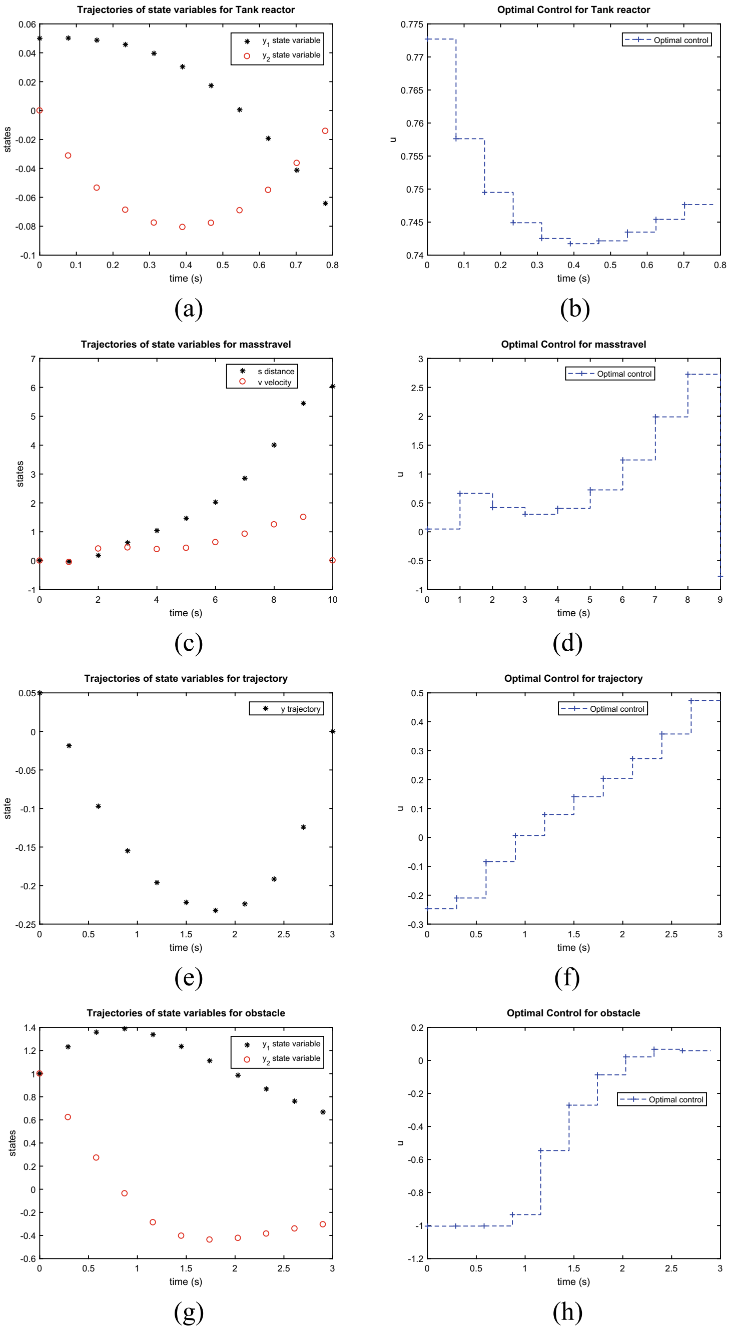
Problem 24.5 In the point mass maximum travel example (*masstravel*), the force $u(t)$ that moves a mass to the longest distance is to be found (with $T = 10$ fixed):

$$\begin{aligned}
\max_{u(t)} \; & J \equiv \int_0^T v(t)\, dt \\
\text{s.t. } & s'(t) = v(t) \\
& v'(t) = u(t) - k_0 - k_1 v(t) - k_2 v(t)^2, \quad t \in [0, T] \\
& s(0) = 0, \;\; v(0) = 0, \;\; v(T) = 0 \\
& |u(t)| \le g + k_3 v(t)^2, \quad t \in [0, T] \;.
\end{aligned}$$

The results, for $k_0 = 0.1,\; k_1 = 0.2,\; k_2 = 1,\; k_3 = 1$ and $N = 11$, are shown in Table 24.3. The initial guesses were $s(t_i) = 1,\; v(t_i) = 2, i \in I_N$ and $u(t_i) = 5, i \in I$. When transforming the above form into the *Mayer form*, the objective function value is just $s(T)$ (thus no new state variable was added to the states vector). To confirm convergence, the problem is also solved with $\eta_1 = 1E{-}10,\; \eta_2 = 1E{-}06$ in

Table 24.3 Results obtained for the Problems 24.4, 24.5, 24.6 and 24.7

	k	J	θ	p	nfe	Time
Tank reactor	1	0.0046	$1.12E{-}02$	$0.0000E{+}00$		
	176	0.0357	$9.9503E{-}05$	$0.0000E{+}00$	16320	18.0
Masstravel	1	3.2633	$6.9821E{+}01$	$1.6000E{+}02$		
	69	6.0311	$7.9855E{-}05$	$0.0000E{+}00$	4830	5.3
	128§	6.0256	$9.2528E{-}11$	$0.0000E{+}00$	8963	9.7
Trajectory	1	0.6457	$1.6043E{+}01$	$1.0424E{+}01$		
	56	0.2691	$9.3978E{-}05$	$0.0000E{+}00$	3922	4.4
	307§	0.2635	$8.8477E{-}11$	$0.0000E{+}00$	21494	22.5
Obstacle	1	0.0000	$2.4395E{+}00$	$0.0000E{+}00$		
	341	2.3257	$9.2300E{-}05$	$2.5452E{-}05$	26208	27.1
	750§	2.4616	$1.3062E{-}08$	$4.8821E{-}10$	52702	53.7



Trajectories of state variables for Tank reactor

Optimal Control for Tank reactor

Trajectories of state variables for masstravel

Optimal Control for masstravel

Trajectories of state variables for trajectory

Optimal Control for trajectory

Trajectories of state variables for obstacle

Optimal Control for obstacle

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 24.2 States trajectory and optimal control. **a** States for problem *Tank reactor*. **b** Control for problem *Tank reactor*. **c** States for problem *masstravel*. **d** Control for problem *masstravel*. **e** States for problem *trajectory*. **f** Control for problem *trajectory*. **g** States for problem *obstacle*. **h** Control for problem *obstacle*

(24.11)—identified with § in Table 24.3. Figures 24.2c, d contain the states and control respectively.

Problem 24.6 (*trajectory*) Find $u(t)$ that minimizes J (with $T = 3$ fixed) [4],

$$\begin{aligned} \min_{u(t)} \; & J \equiv \int_0^T (y^2(t) + u^2(t))\, dt \\ \text{s.t.} \; & y'(t) = (1 + y(t))y(t) + u(t), \quad t \in [0, T] \\ & y(0) = 0.05, \;\; y(T) = 0, \\ & |y(t)| \le 1, \;\; |u(t)| \le 1, \;\; t \in [0, T]\ . \end{aligned}$$

The obtained results for $N = 11$, with the initial guesses $y(t_i) = 1, i \in I_N$ and $u(t_i) = 0, i \in I$, are displayed in Table 24.3. Results with $\eta_1 = 1E{-}10, \eta_2 = 1E{-}06$ in (24.11) are also included. The Fig. 24.2e, f present the states and control respectively.

Problem 24.7 The obstacle problem (*obstacle*) can be reformulated as [3] ($T = 2.9$):

$$\begin{aligned} \min_{u(t)} \; & J \equiv 5y_1(T)^2 + y_2(T)^2 \\ \text{s.t.} \; & y_1'(t) = y_2(t) \\ & y_2'(t) = u(t) - 0.1(1 + 2y_1(t)^2)y_2(t) \\ & y_1(0) = 1, \;\; y_2(0) = 1, \\ & 1 - 9(y_1(t) - 1)^2 - (\tfrac{y_2(t) - 0.4}{0.3})^2 \le 0, \\ & -0.8 - y_2(t) \le 0, \;\; |u(t)| \le 1, \quad t \in [0, T] \end{aligned}$$

Using the initial guesses $y_1(t_i) = 0, \; y_2(t_i) = 0, i \in I_N, \; u(t_i) = 0, i \in I$ and $N = 11$, the results are shown in Table 24.3. This problem is also solved with $\eta_1 = 1E{-}10$, $\eta_2 = 1E{-}06$ in (24.11) to analyze the convergence issue. Figures 24.2g, h show the states y_1, y_2 and control u respectively.

24.5 Conclusions

A first-order descent method based on a filter methodology is proposed to solve a finite-dimensional nonlinear optimization problem that arises from the use of a direct multiple shooting method for OCP. The implemented filter method relies on three measures. The two feasibility measures are handled separately in order to give priority to the minimization of the ‘continuity constraints’ violation over the algebraic equality and inequality constraints violation and the objective function. This priority is patent by the use of search directions that are along either the negative of the gradient of the ‘continuity constraints’ violation function or a negative convex combination of that gradient and the gradient of the other constraints violation, or the objective function. Numerical derivatives are implemented in order to avoid computing the first derivatives of the involved functions. The numerical experiments carried out until now have shown that the presented strategy is worth pursuing.

Issues related to the extension of the proposed method to solving retarded OCP with constant delays in the state variables and in the control are now under investigation and will be the subject of a future paper.

Acknowledgments We acknowledge the financial support of CIDEM, R&D unit funded by the FCT—Portuguese Foundation for the Development of Science and Technology, Ministry of Science, Technology and Higher Education, under the Project UID/EMS/0615/2019, and FCT within the Projects Scope: UIDB/00319/2020 and UIDB/00013/2020 and UIDP/00013/2020 of CMAT-UM.

References

1. Biegler L (2010) Nonlinear programming: concepts, algorithms, and applications to chemical processes. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA
2. Frego M (2014) Numerical methods for optimal control problems with applications to autonomous vehicles. PhD thesis, University of Trento
3. Schlegel M, Stockmann K, Binder T et al (2005) Dynamic optimization using adaptive control vector parameterization. Comput Chem Eng 29(8):1731–1751
4. Diehl M, Bock HG, Diedam H et al (2006) Fast direct multiple shooting algorithms for optimal robot control. In: Diehl M, Mombaur K (eds) Fast motions in biomechanics and robotics. Lecture notes in control and information sciences, vol 340, pp 65–93
5. Fletcher R, Leyffer S (2002) Nonlinear programming without a penalty function. Math Program Ser A 91(2):239–269
6. Audet C, Dennis JE Jr (2004) A pattern search filter method for nonlinear programming without derivatives. SIAM J Optim 14(4):980–1010
7. Assassa F, Marquardt W (2014) Dynamic optimization using adaptive direct multiple shooting. Comput Chem Eng 60:242–259
8. Schäder A, Kühl P, Diehl M et al (2007) Fast reduced multiple shooting methods for nonlinear model predictive control. Chem Eng Process 46(11):1200–1214
9. How JP 16.323 Principles of Optimal Control. Lecture 7 Numerical Solution in Matlab. Spring 2008. Massachusetts Institute of Technology: MIT OpenCourseWare. License: Creative Commons BY-NC-SA. http://ocw.mit.edu
10. Kirk DE (1970) Optimal control theory: an introduction. Prentice Hall Inc, NJ

Chapter 25
Irrigation Planning with Fine Meshes

Sofia O. Lopes, M. Fernanda P. Costa, Rui M. S. Pereira, M. T. Malheiro, and Fernando A. C. C. Fontes

Abstract In this work, we study a mathematical model for a smart irrigation system, formulated as an optimal control problem and discretized and transcribed into a nonlinear programming problem using a fine mesh. In order to solve the resulting optimization problem, one needs to use Optimization solvers. Hence, we implemented the proposed mathematical model in AMPL and solved it using the IPOPT solver on the NEOS server (https://neos-server.org/neos/index.html). We also tested the model creating several scenarios. The numerical results shows that the mathematical model produces qualitatively good responses. Moreover the execution times are made in few seconds.

Keywords Smart Irrigation · Optimal Control · IPOPT

S. O. Lopes (✉) · M. F. P. Costa · R. M. S. Pereira
Centre of Physics/Department of Mathematics, University of Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
e-mail: sofialopes@math.uminho.pt

M. F. P. Costa
e-mail: mfc@math.uminho.pt

R. M. S. Pereira
e-mail: rmp@math.uminho.pt

M. T. Malheiro
Centre of Mathematics/Department of Mathematics, University of Minho,
Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: mtm@math.uminho.pt

F. A. C. C. Fontes
SYSTEC, University of Porto, R. Dr. Roberto Frias, 4200-465 Porto, Portugal
e-mail: faf@fe.up.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_25

25.1 Introduction

Climate change is happening [1]. Global warming, extreme weather events, floodings and drought periods are expected even more frequently in the near future. There is strong evidence that Mankind is, to say the least "one of the main actors" promoting this profound change of weather on Earth.

The continuous growth of the world population is exerting a lot of pressure on the planet, which is no longer able to provide the necessary demands.

Water is probably one of the most important resources that will be evermore disputed by many countries. In the past, wars have arisen because of gold, oil, diamonds or other similar luxury goods. Water may become a luxury good very soon. Agriculture is responsible for using most of the planet's existing freshwater. Many regions of the world are already suffering from longer and profound droughts. To produce the increasing demanded food, agriculture must be efficient [2] without exhausting the soils or planet's reserves of freshwater.

The countries which face droughts like Portugal need to develop irrigation systems able to keep the crops safe saving as much water as possible. But most irrigation systems are of the type ON-OFF control. This means they will be activated with a level independent of crop's needs. Sometimes they irrigate too much (wasting water), and sometimes they use too little water (causing unnecessary stress on the crop).

To overcome this drawback, we developed a mathematical model based on Optimal Control Theory which will be able to track the water needs of the crop and provide only the necessary water to fulfil those needs.

As mentioned in [3], Optimal Control Theory emerged as a field of research in the 1950s in response to problems concerning the aerospace exploitation of the solar system [4]. Nowadays, optimal control is a recognized tool, known by its efficacy, applied in different areas, such as robotics [5], biological systems [6], agriculture problems [7], among many others. The goal of Optimal Control is to find a control law for a given system such that a certain optimality criterion is achieved.

Optimal control problems (OCP) are constrained problems that have a set of constraints defined by dynamic systems of ordinary differential equations.

In an OCP, it is possible to use different tools to solve the problem, to characterize it, to study the sensitivity of its variables, to study the stability of the problem and to apply predictive control to re-plan it [7–9].

The mathematical model presented here was firstly implemented and solved using a direct method available in MATLAB® (MATLAB is a registered trademark of the MathWorks, Inc.) and produces an optimal irrigation plan for a number of days (no more than 10), based on the weather forecast, moisture sensors in site and a set of parameters describing the type of soil, crop, irrigation and location. The optimal solution found showed that the proposed mathematical model is qualitatively correct [10, 11].

We will present a rebuilt model that copes with hourly data and also includes restrictions on when to irrigate the crop during the day. This new model was written in AMPL [12] and solved in the NEOS platform (https://neos-server.org/neos/index.html) using IPOPT solver [13]. Several scenarios are presented. Results are qualitatively good.

This paper is organized as follows. Section 25.2 is devoted to present the base mathematical model used and to explain its main features. Sect. 25.3 is dedicated to validation of the proposed model on a set of examples. Finally, Sect. 25.4 presents conclusions and future work.

25.2 Mathematical Model

Based on the mathematical model presented in Lopes et al. [14], we rebuilt it in such a way that rainfall forecast was obtained hourly and with access to the soil moisture at any time, via a moisture sensor applied on site. The fact that part of the data is now obtained on an hourly basis increases the number of variables of the optimization problem, by a factor of 24 ($N = 24$ is the number of hours of the irrigation plan as described in the proposed mathematical model—see Eq. (25.1)). Due to this, the problem becomes a large-scale optimization one, with 456 decision variables and 673 constraints. We note that, the size of the problem is defined by the number of variables plus the number of constraints. Problems that have sizes at least 1000 are considered large-scale problems [15].

Therefore, when an hourly basis is used problem (25.1) is large-scale one. Solving it using some of the optimization methods available in the fmincon solver from MATLAB®, is prohibitive due to memory requirements and CPU time.

The cost function to be minimized is the amount of water used in the irrigation system, defined by the sum of the control variables **u**, subject to the dynamic equation in which the variation of the moisture of the soil (the trajectory variables **x**) has to satisfy the water balance equation, and subject to the inequality constraints, namely, the amount of water that come from the tap can not be negative and the moisture of the soil must satisfy the hydric needs, x_{min}, of the crop. The last constraint allows us to prescribe the time of the day when the irrigation is released. In this case, we do not allow irrigation between a certain period of time, $[time_1, time_2]$. This is important in the summer, such that the irrigation plan takes place when temperatures are not too high. Thus, the proposed new model is defined as follows:

$$\begin{array}{lll} \min_{x_i,u_i} & h\sum_{i=1}^{N} u_i & \\ \text{s.t.:} & x_{i+1} = x_i + hf(t_i, x_i, u_i, x_{i+1}), & i = 1, \ldots, N \\ & x_i \geq x_{min} & \\ & 0 \leq u_i \leq U_{sup}, & i = 1, \ldots, N \\ & x_1 = x_0, & \end{array} \tag{25.1}$$

where U_{sup} represents the maximum irrigation possible. An extra constraint can be added to the model, in order to allow the farmer to irrigate the crops at a desirable time of the day:

$$U_{sup} = 0, \quad time_1 \leq mod(i, 24) \leq time_2 \tag{25.2}$$

where $mod(i, 24)$ gives the remainder of the integer division of i by 24. The function f present in the dynamic of the OCP is given by

$$f(t, x_l, u_l, x_r) = K_I \times u_l + K_R \times rfall(t) - K_C \times evtp_0(t) - loss(t, x_l, x_r),$$

with the term of losses defined by [16],

$$loss(t, x_l, x_r) = \begin{cases} k(t)x_l & \text{if } x_r \leq x_{FC} \\ x_r - x_{FC} + k(t)x_l & \text{if } x_r > x_{FC}. \end{cases}$$

The parameter h represents the time step, N is the number of time steps of the irrigation planning (24× number of days of the plan), K_I is the coefficient associated to the type of irrigation, K_R is a parameter associated to the rainfall ($rfall$), K_C is the coefficient associated to the type of crop, $evtp_0$ is the reference evapotranspiration and x_0 is the moisture of the soil at initial time. Notice that $k(t)$ depends on the type of soil and x_{FC} is the humidity of soil at available water capacity. More details about the model can be found in [14]. The fact that we now may have hourly data will allow us to have a more accurate solution. Weather data and soil moisture were taken from a database [17] in ISEP (Instituto Superior de Engenharia do Porto), evapotranspiration was calculated using Penman–Monteith model [18], crop and soil coefficients were obtained in the Raposo's book [19].

25.3 Results and Validation

In this section we focus on validating the proposed model. It was implemented firstly in MATLAB®, considering a daily basis. When a daily basis is considered, problem (25.1) is a small-scale one. To solve it, the fmincon solver of MATLAB® with the optimization 'active-set' algorithm was used. Furthermore, we used the fmincon solver with the other options by default. The optimization algorithm was able to obtain the optimal solution and the execution time took few minutes.

However, if the problem is a large-scale one, CPU times and memory requirements will become an issue if 'active-set' algorithm is chosen. We note that, the 'active-set' algorithm is no longer recommended since it is not a large-scale algorithm. The 'active-set' algorithm requires the storage of full matrices and use dense linear algebra. The storage of full matrices needs a significant amount of memory, and the dense linear algebra may require a long time to execute.

Hence, we opted to use the ‘interior-point’ algorithm which is a large-scale one, instead of the ‘active-set’ algorithm, in the fmincon solver. However, when we tried to solve one of this large-scale problems the obtained optimal solution was reported as may be inaccurate and the execution times took about 50 min.

Because of it (future evolutions of the model will be more and more complex), the model was written in AMPL language [12] and solved in NEOS Platform using the IPOPT solver [13] with its options by default. CPU time was much improved. Next, we present some scenarios and analyze the respective results relative to the validation of the proposed model.

25.3.1 Case 1—Irrigation Plan for a Set of Rainy Days in April

Here, we consider a simulation using real weather data from a set of days in April. Rainfall was heavy. No restrictions on the time for irrigation were considered in this scenario 1. Furthermore, we also assumed uniform rainfall during the whole days of the irrigation plan, and we considered a grass field in Oporto (Portugal).

The optimal solution of the problem (25.1) for scenario 1, namely, irrigation plan, moisture in the soil and the hydric needs of the crop, is shown in Fig. 25.1.

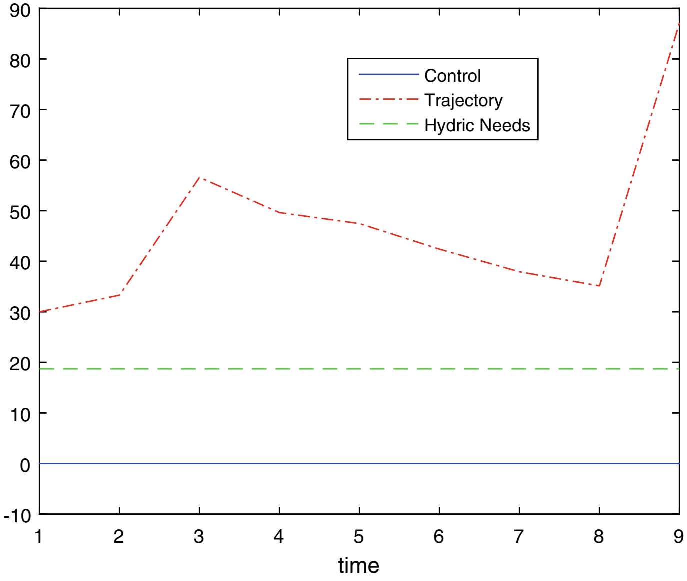


Fig. 25.1 Results for scenario 1—set of rainy days in April in the region of Oporto, Portugal. Coarse grid in time with time step = 1day

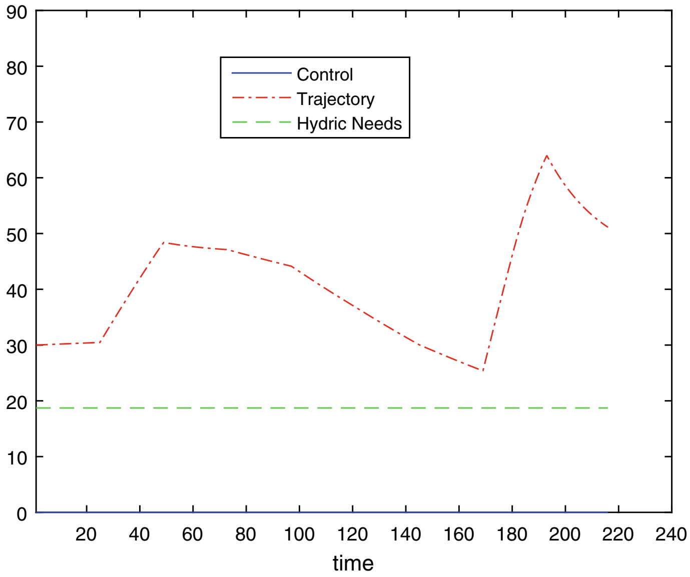


Fig. 25.2 Results for scenario 1—set of rainy days in April in the region of Oporto, Portugal. Finer grid in time with time step = 1 h

As expected, no irrigation is needed, since the moisture in the soil was well above the hydric needs of the crop. Water consumption for this scenario was 0 mm, since the control was not activated.

Next, we assumed hourly non-real data rainfall and it was considered uniform along the day. This meant that the number of variables increased by a factor of 24. When solving this problem using the fmincon solver of MATLAB®, the programme took about 50 min to obtain the optimal solution. In the NEOS platform with the IPOPT solver, just took a few seconds. Results are shown in Fig. 25.2.

As expected, results are very similar. No irrigation is needed. The trajectory is different since we are using smaller time step. Herein the trajectory is potentially more accurate. In future, if we have real data this would reflect better the weather variations along the day. Nowadays, it is not rare to have a weather station in site and moisture sensors which are able to capture hourly data and transmit it to a server where they can be collected. Water consumption for this scenario was 0 mm since the control was not activated.

In scenario 2, all the rain of each day was considered to take place in a couple of hours, generating a greater rainfall in that short period of time. The temperature was modelled in each day by a parabola with the maximum value at 12 h and minimums at time 0 h and 23:59 and an average value equal to the average temperature of that day. The optimal solution of the problem (25.1) considering scenario 2 is shown in Fig. 25.3.

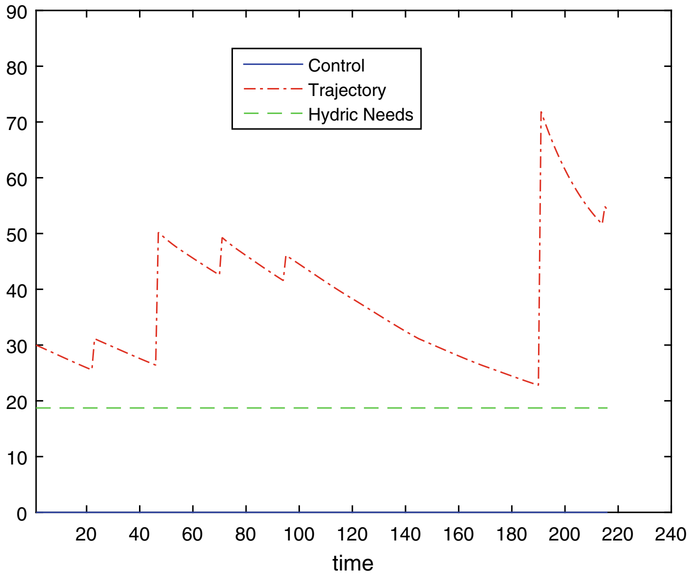


Fig. 25.3 Results for scenario 2—set of winter rainy days in the region of Oporto, Portugal. Finer grid in time with time step = 1 h. The rainfall occurs in just a couple of hours and the temperature is defined in each day by a parabola

As expected, results are no longer the same. The concentration of the rainfall in a couple of hours may produce a situation where the soil moisture sudden increases. Water consumption for scenario 2 was 0 mm, since the control was not activated.

25.3.2 Case 2—Irrigation Plan for a Set of Dry Days in August

25.3.2.1 For a Crop of Grass

Here, we consider a simulation using real weather data from a set of days in August. Rainfall was scarce (rain only took place on the 9th day and was very scarce). No restrictions on the time for irrigation were considered in the scenario 3a. Furthermore, we considered a grass field in Oporto (Portugal), and we assumed that we had only daily data. The optimal solution of the problem (25.1) for scenario 3a—the irrigation plan, moisture in the soil and the hydric needs of the crop—is depicted in Fig. 25.4.

As expected, irrigation is always needed, since the moisture in the soil was always near the hydric needs of the crop. If irrigation does not take place at any time step, the crop enters in stress. Water consumption for this example was 30.5768 mm.

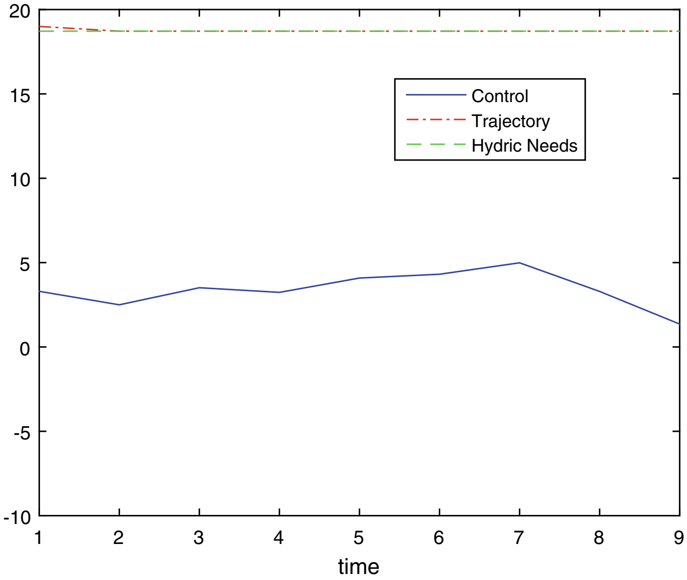


Fig. 25.4 Results for scenario 3a—set of dry August days in the region of Oporto, Portugal. Coarse grid in time with time step = 1day

In scenario 4a, hourly data was assumed to be available. It was generated assuming uniform rainfall during the whole days. Results can be seen in Fig. 25.5.

The control is composed by a series of step functions since for every 24 h, we consider uniform rainfall with average equal to the rainfall average of that day. Notice that this is not verified in Case 1, since no irrigation was needed. Results are potentially more accurate and if we had real data, this would better reflect the weather variations along a day. Water consumption for scenario 4a was 30.5492 mm. The total amount of irrigation is similar, but slightly smaller. This is due to the fact that the time steps are smaller.

In the next scenario, scenario 5a, we suppose that the rainfall in each day takes place in a couple of hours of that day and that the temperature was modelled (in each day) by a parabola with the maximum value at 12 h and minimums at 0 h and 23:59, and an average value equal to the average temperature of that day. The results for scenario 5a are shown in Fig. 25.6.

As expected, results are no longer the same. The concentration of the rainfall in a couple of hours of each day did not produce similar results as in Fig. 25.3 for scenario 2 (peaks in the trajectory), because in scenario 5a, rainfall was very scarce. The effect of considering parabolic arcs to model the daily temperature can be seen in Fig. 25.6. The consumption increases a bit (31.291 mm) due to the fact that rainfall was concentrated in a couple of hours for each day and the temperature along the day was given by a parabola.

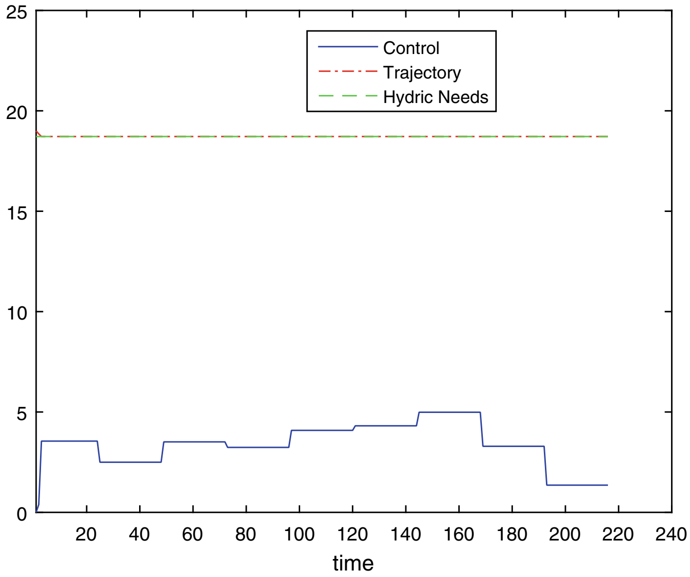


Fig. 25.5 Results for scenario 4a—set of dry August days in the region of Oporto, Portugal. Finer grid in time with time step = 1 h

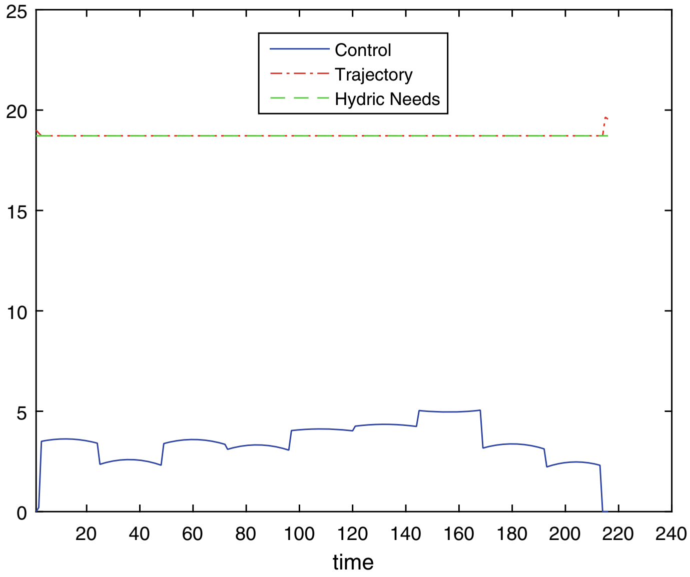


Fig. 25.6 Results for scenario 5a—set of August days in the region of Oporto, Portugal. Finer grid in time with time step = 1 h. The rainfall occurs in just a couple of hours and the temperature is defined in each day by a parabola

25.3.2.2 For a Crop of Mint

Next, using the same weather data for scenario 3a (note rain only took place on the 9th day and was very scarce), we consider that the crop is no longer grass, but now mint. This scenario is denoted by scenario 3b. The reference evapotranspiration coefficient is now 1.15 instead of 0.95. The optimal solution of problem (25.1) for scenario 3b is shown in Fig. 25.7. The water consumption is now 34.99 mm instead of 30.58 mm.

As expected water consumption increases relatively to scenario 3a, since evapotranspiration of the new crop is much higher.

Next, for the crop of mint, we consider the same weather data of the scenario 4a. This scenario is denoted by scenario 4b. In this scenario, hourly data was assumed to be available. It was generated assuming uniform rainfall during the whole days. The optimal solution of the problem (25.1) for scenario 4b is shown in Fig. 25.8.

The control is composed by a series of step functions since for every 24 h, we consider uniform rainfall with average equal to the rainfall average of that day. Water consumption was 34.84 mm. The total amount of irrigation is similar, but a slightly smaller. This is due to the fact that the time steps are smaller.

In next scenario, scenario 5b, the same weather data of scenario 5a was considered. We suppose that the rainfall in each day takes place in a couple of hours of that day and that the temperature was modelled (in each day) by a parabola with the maximum

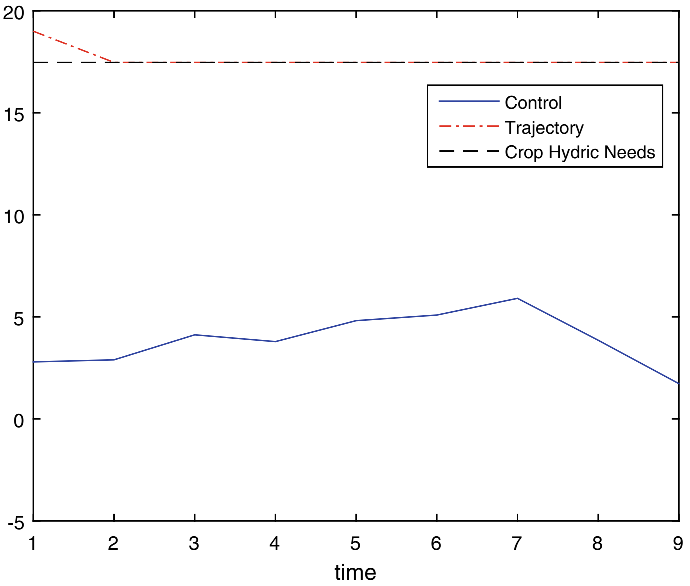


Fig. 25.7 Results for scenario 3b—set of dry August days in the region of Oporto, Portugal. Crop is mint instead of grass

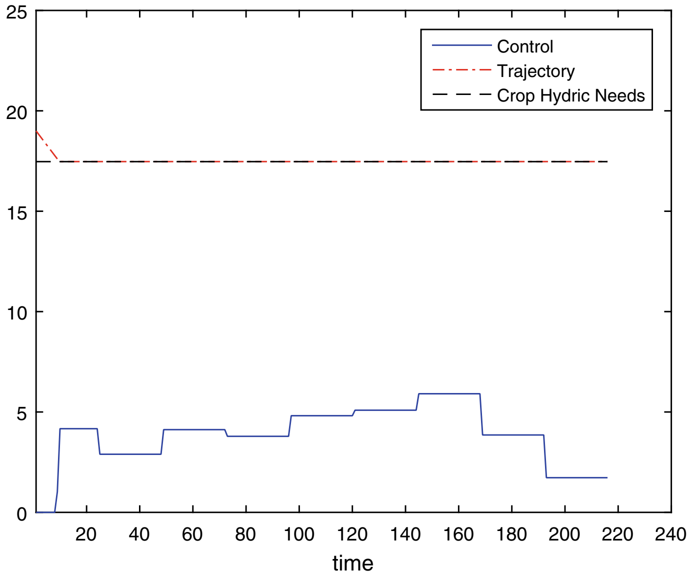


Fig. 25.8 Results for scenario 4b—set of dry August days in the region of Oporto, Portugal. Finer grid in time with time step = 1 h. Crop is mint instead of grass

value at 12 h and minimums at 0 h and 23:59, and an average value equal to the average temperature of that day. The optimal solution of the problem (25.1) for scenario 5b is shown in Fig. 25.9.

The consumption increases a bit (35.54 mm) due to the rainfall to be concentrated in a couple of hours for each day. As we can see from these last three scenarios, if the crop has a greater evapotranspiration, it consumes more water, as expected.

25.3.3 Case 3—Imposing a Constraint for the Daily Period of Irrigation

Here we will study the effect of imposing the constraint on when the irrigation takes place on the crop field (scenario 6a). We will consider the data from scenario 5a with a finer grid and uniform distribution of rainfall along every hour of each day. We will also consider irrigation cannot take place between 11h and 19h. We note that, due to the high temperatures in the summer, the mathematical model must be able to prevent that the crop does not dye. The optimal solution of the problem (25.1) for the scenario 6a is shown in Fig. 25.10.

Since irrigation has to stop during the hottest hours of the day, when it restarts it will produce a peak to compensate. You can also observe that after every peak there is a slight increase in the trajectory. The total amount of irrigation was 30.6097 mm.

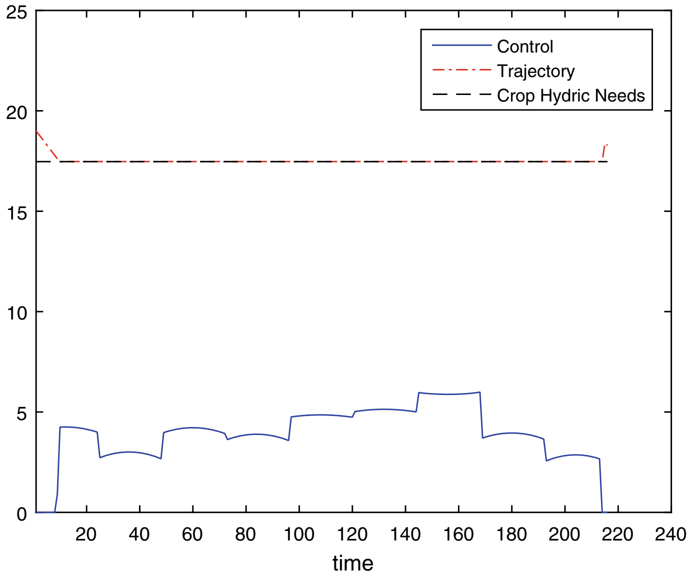


Fig. 25.9 Results for scenario 5b—set of August days in the region of Oporto, Portugal. Finer grid in time with time step = 1 h. The rainfall occurs in just a couple of hours and the temperature is defined in each day by a parabola. The crop is mint

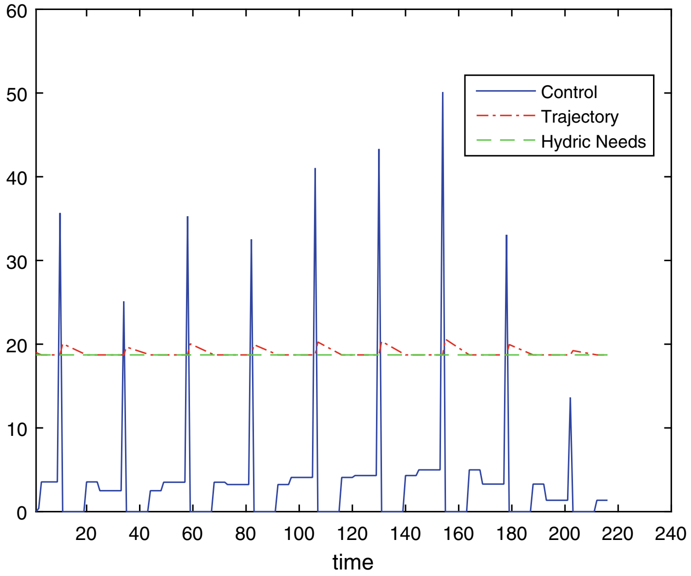


Fig. 25.10 Results for scenario 6a—including the constraint of not irrigating between 11h and 19h of each day. Finer grid in time with time step = 1 h

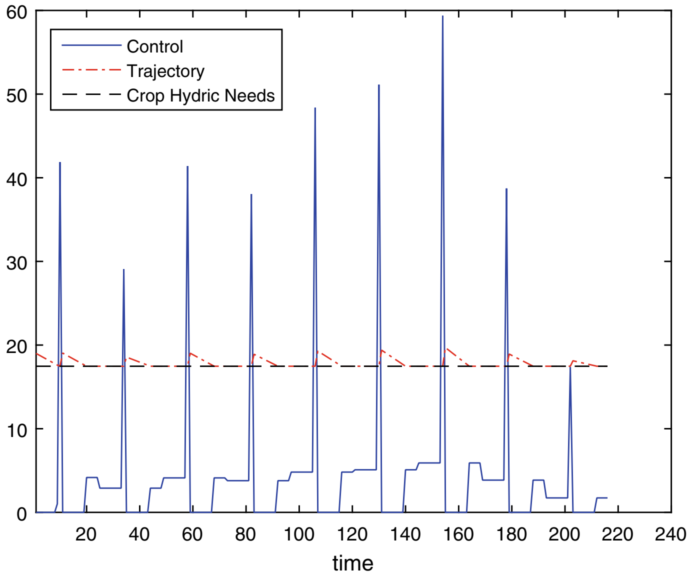


Fig. 25.11 Results for scenario 6b—including the constraint of not irrigating between 11 h and 19 h of each day. Finer grid in time with time step = 1 h. The crop is mint

The same procedure applies if the crop is mint (scenario 6b). We consider the data from scenario 5b with a finer grid and uniform distribution of rainfall along every hour of each day. We also consider irrigation cannot take place between 11h and 19h. The optimal solution of the problem (25.1) for the scenario 6b is shown in Fig. 25.11.

Since irrigation has to stop during the hottest hours of the day, when it restarts it will produce a peak to compensate. You can also observe that after every peak there is a slight increase in the trajectory. The total amount of irrigation was 34.91 mm. As expected the same pattern appears, the difference is due to the fact that evapotranspiration of mint to be greater than the one of grass.

25.4 Conclusions and Future Work

We designed a mathematical model based on Optimal Control Theory which, given a set of data (weather data, soil moisture, evapotranspiration, a set crop and soil coefficients), is able to produce an irrigation plan for the crop for a given number of days. We used data from a grass field in ISEP (Instituto de Engenharia do Porto) and collected in a local database [17]. Other parameters were consulted in the bibliography [18, 19].

We verified that the proposed mathematical model is able to produce solutions that correspond to the reality needs.

The inclusion of hourly data allows to consider scenarios where, for instance, rainfall is uniform or the opposite, and the rainfall can be concentrated in a couple of hours. We presented simulations in the previous section and results are qualitatively good. We note that, more accurate data with smaller time steps will potentially produce a smoother solution, with less consumption. It will also allow to take into account extreme events that occur in a matter of minutes/hours.

The inclusion of as constraint on when irrigation can take place during the day makes the model more realistic and allows us to avoid unnecessary damage to the crop. The numerical results allow us to conclude that the model produces qualitatively good responses.

In future, other tests will be done to adjust some parameters of the model and validate it in crop field. Other features still need to be tackled, such as: to see the effects of different types of soils; different types of crops; to consider a slope in the crop field that will produce losses due to runoff; to introduce new constraints that might enrich the model.

The final goal of the study is to produce a prototype of smart irrigation which step by step will help farmers to produce sustainable agriculture. The data needs to be collected in site automatically using a mini-weather station and uploaded to a server. A web page installed in that server collects some parameters which the farmer needs to provide accordingly to the type of soil, type of irrigation used, type of crop, number of days considered in the plan, etc. Once the data-file is completed, the user starts a simulation using our model and finally, obtains the Irrigation plan. The irrigation system needs to be properly designed but, if so, with a file providing the needs of water of the crop field at every time step, it can automatically start the irrigation when needed, guaranteeing the crop is safe and the waste of water is minimum.

Acknwoledgments FEDER/COMPETE/-NORTE2020/POCI/FCT funds through grants UID/EEA/-00147/20 13/UID/IEEA/00147/ 006933-SYSTEC, PTDC-EEI-AUT-2933-2014116858-TOCCATA, project, To CHAIR—POCI-01-0145-FEDER-028247 and UPWIND—POCI-FEDER-FCT-31447. This work was also supported by the Portuguese Foundation for Science and Technology (FCT) in the framework of the Strategic Funding UID/FIS/04650/2019 and UIDB/00013/2020 and UIDP/00013/2020 of CMAT-UM.

References

1. IPCC (2007) Climate change 2007: synthesis report. Contribution of working groups I, II and III to the fourth assessment report of the intergovernmental panel on climate change. Geneva, Switzerland (2008)
2. Haie N, Pereira RMS, Machado G, Keller AA (2012) Analysis of effective efficiency in decision making for irrigation interventions. Water Resour 39(6):700–707
3. Lopes SO, Costa MFC, Pereira RMS, Fontes FACC (2019) Replanning the irrigation systems. In: Proceedings of the 4th international conference on numerical and symbolic computation developments and applications, pp 533–540

4. Longuski JM, Guzmán JJ, Prussing JE (2014) Optimal control with aerospace applications. Springer, London
5. de Jager B, van Keulen T, Kessels J (2013) Optimal control of hybrid vehicles. Springer, London
6. Lenhart S, Workman JT (2007) Optimal control applied to biological models. Chapman and Hall/CRC Press, London
7. Lopes SO, Fontes FACC, Pereira RMS, de Pinho MDR, Ribeiro C (2014) Optimal control for an irrigation planning problem: characterisation of solution and validation of the numerical results. In: Controlo 2014, pp 157–167
8. Lopes SO, Fontes FACC, Costa MFP, Pereira RMS, Gonçalves AM, Machado GJ (2013) Irrigation planning: replanning and numerical solution. AIP Conf Proc 1558(1):626–629
9. Paiva LT, Fontes FACC (2017) Sampled-data model predictive control using adaptive time-mesh refinement algorithms. In: Controlo 2016, pp 143–153
10. Lopes SO, Fontes F, Pereira RMS, Machado GJ (2011) Irrigation planning in the context of climate change. In: Mathematical models for engineering science, pp 239–244
11. Lopes SO, Pereira RMS, Pereira P, Caldeira A, Fonte V (2019) Optimal control applied to an irrigation planning problem: a real case study in Portugal. Int J Hydrol Sci Technol 9(2):173–188
12. Fourer R, Gay DM, Kernighan BW (1993) AMPL: a modeling language for mathematical programming. Scientific, USA
13. Wachter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106(1):25–57
14. Lopes SO, Fontes FACC, Pereira RMS, de Pinho MDR, Gonçalves AM (2016) Optimal control applied to an irrigation planning problem. Math Probl Eng 10p
15. Benson HY, Shanno DF, Vanderbei RJ (2003) A comparative study of large-scale nonlinear optimization algorithms. High Perform Algorithms Softw Nonlinear Optim 82:95–127
16. Horton RE (1941) An approach toward a physical interpretation of infiltration capacity. Soil Sci Soc Am J 5(C):300–417
17. Meteorology in the Instituto Superior de Engenharia do Porto (ISEP) (2016). http://meteo.isep.ipp.pt/gauges. Accessed June 2016
18. Allen R, Pereira L, Raes B, Smith M (1998) Crop evapotranspiration-guidelines for computing crop water requirements. FAO Irrig Drain Pap 56:1–15
19. Raposo JR (1996) A REGA. Dos primitivos regadios às modernas técnicas de rega. Fundação Calouste Gulbenkian, Portugal

Chapter 26
Optimal Path and Path-Following Control in Airborne Wind Energy Systems

Manuel C. R. M. Fernandes, Luís Tiago Paiva, and Fernando A. C. C. Fontes

Abstract An Airborne Wind Energy System (AWES) is a concept to convert wind energy into electricity, which comprises a tethered aircraft connected to a ground station. These systems are capable of harvesting high altitude winds, which are more frequent and more consistent. Among AWES, there are Pumping Kite Generators (PKG) that involve a rigid or flexible kite connected to a motor/generator placed on the ground through a light–weight tether. Such PKG produces electrical power in a cyclical two–phased motion with a traction phase and a retraction phase. During the traction phase, the aim is to maximize power production. This goal is achieved by controlling the kite such that it performs an almost crosswind motion, keeping a low elevation angle in order to maximize the tether tension. During the retraction phase, the tether tension force is minimized by steering the kite while the tether is reeled–in. Such strategy assures that the cyclical two–phased motion has a positive electrical balance at the end of the overall cycle. In a first stage, we solve an optimal control problem to compute the optimal plan for the kite trajectory during the traction phase, maximizing power production. Such trajectory is then used to define a time–independent geometrical path, which in turn is used as the reference path for the path–following control procedure that is developed in a second stage, and for which results are also presented.

Keywords Airborne wind energy · Optimal control · Heading angle steering · Path-following

M. C. R. M. Fernandes (✉) · L. T. Paiva · F. A. C. C. Fontes
SYSTEC, Faculdade de Engenharia, Universidade do Porto, R. Dr. Roberto Frias, 4200-465 Porto, Portugal
e-mail: mcrmf@fe.up.pt

L. T. Paiva
e-mail: ltpaiva@fe.up.pt

F. A. C. C. Fontes
e-mail: faf@fe.up.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_26

26.1 Introduction

In the last decades there has been a fast growth of the investment and technical development of renewable energy systems, both within companies and academia. Among the renewable energy sources, wind is an important large scale alternative and it is still mostly unexplored. Wind is mainly harvested on–shore at low heights by wind turbines mounted on towers with a few dozen meters ($50-200$ m), nevertheless most of the existing wind energy is available at high altitudes and offshore.

Airborne Wind Energy Systems (AWES) aim at exploiting stronger and more consistent high–altitude winds. Recent solutions range from lighter than air concepts, airfoils with electrical generation on the aircraft, or on the ground, developed or currently being developed [1]. One of the most promising technologies are the Pumping Kite Generators (PKG) [2, 3]. These systems use a tethered kite—a flexible or rigid wing—that is connected to a winch drum coupled to a motor/generator placed on the ground.

PKG produce electrical power in a cyclical two–phased motion with a traction phase and a retraction phase. During the traction phase, the power production is maximized as the kite is controlled such that it performs an almost crosswind motion, keeping a low elevation angle in order to maximize the tether tension. When we reach the maximum tether length, the system enters in the retraction phase, where the kite is controlled such that the tether tension is minimized while the tether is reeled–in. Such strategy assures that, at the end of the cycle, the two–phased motion has positive electrical balance. Such systems exploit crosswind kite power as described by Loyd in 1980 [4]. The power harvesting potential of PKG is supported by two important factors:

1. wind speeds increase with height,
2. the aerodynamic lift ($\mathbf{F}^{\text{lift}}$) is proportional to the square of the apparent wind velocity,

$$\mathbf{F}^{\text{lift}} = \frac{1}{2} c_L(\alpha) A \mathbf{v}_a^2. \tag{26.1}$$

Therefore, the maximum mechanical power extracted from this renewable resource is obtained when the kite flies at high speeds in a crosswind direction. This operation principle can be applied in other fluids, such as water, as explored in [5]. Economical studies involving multiple PKG in a wind farm layout are already available [6, 7].

Since power harvesting capabilities are highly dependent on the flight trajectory and consequently on the control systems that steer the kite, the search for control systems optimality is of key importance for the development of such a technology. We address the Optimal Control Problem (OCP) of maximizing the power withdrawn from the wind during the traction phase using direct methods. For this purpose, we use a 3D model of the kite dynamics, considering all the forces acting on it, [8]. Then, we use the numerical results to describe an optimal path which is used as a reference for a path–following control strategy. According to such strategy, the

Table 26.1 Nomenclature

A	Wing reference area of kite (m^2)	$\mathrm{R_{GL}}$	Rotation matrix from G to L
a_t	Tether reel–out acceleration ($\mathrm{m\,s^{-2}}$)	$\mathrm{R_{LG}}$	Rotation matrix from L to G
c_D	Aerodynamic drag coefficient	r	Tether length (m)
c_L	Aerodynamic lift coefficient	ρ	Air density ($\mathrm{kg\,m^{-3}}$)
E	Energy produced (Ws)	T	Tether tension (N)
$\mathbf{F}^\mathrm{aer}$	Aerodynamic force (N)	$\mathbf{v}_a$	Apparent wind velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{F}^\mathrm{drag}$	Drag force (N)	$\mathbf{v}_w$	Wind velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{F}^\mathrm{cent}$	Centrifugal force (N)	v_t	Tether reel–out velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{F}^\mathrm{cor}$	Coriolis force (N)	$\mathbf{u}$	Control vector
$\mathbf{F}^\mathrm{lift}$	Aerodynamic lift force (N)	$\mathbf{x}$	State vector
$\mathbf{F}^\mathrm{inert}$	Inertial forces (N)	α	Angle of attack (rad)
$\mathbf{F}^\mathrm{th}$	Tether force (N)	ϕ	Azimuthal angle (rad)
g	Gravitational acceleration ($\mathrm{m\,s^{-2}}$)	β	Elevation angle (rad)
m	Mass (kg)	ψ	Roll angle (rad)
P	Power produced (W)	γ	Reference tracking angle (rad)
$\mathbf{p}$	Kite position (m)	τ	Local tangent plane

trajectory controller acts on the roll angle to change the kite heading direction in order to follow a reference point in the established optimal path (see [9, 10] for other path–following control approaches).

This paper is organized as follows. In Sect. 26.2, we describe a model for the kite power system. The nomenclature used is given in Table 26.1. In Sect. 26.3, we define the OCP for maximizing power production in the traction phase. In Sect. 26.4, we address the path definition and path–following control method. In Sect. 26.6, we outline the future work that is under development.

26.2 3D Kite Model

We consider three coordinate systems to model the kite:

Global G: An inertial Cartesian coordinate system (x, y, z) with the origin on the ground at the point of attachment of the tether and where x is aligned according to the wind direction $\mathbf{v}_w = (v_w, 0, 0)$, on the basis of $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$.
We consider that the kite is positioned in a point $\mathbf{p}$ with coordinates (x, y, z).

Local L: A non–inertial spherical coordinate system (r, ϕ, β) on the basis of $(\mathbf{e}_r, \mathbf{e}_\phi, \mathbf{e}_\beta)$ (Fig. 26.1).

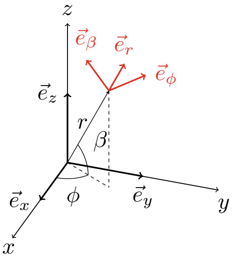


Fig. 26.1 The global and local coordinate systems

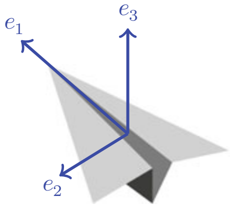


Fig. 26.2 The body coordinate system

Body B: A non–inertial Cartesian coordinate system attached to the kite body on the basis of $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ where $\mathbf{e}_1$ coincides with the kite longitudinal axis pointing forward, $\mathbf{e}_2$ in the kite transversal axis, points to the left wing tip, and $\mathbf{e}_3$ in the kite vertical axis is pointing upwards (Fig. 26.2).

Considering the kite mass-point position

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\beta)\cos(\phi) \\ r\cos(\beta)\sin(\phi) \\ r\sin(\beta) \end{bmatrix},$$

the rotation matrix from L coordinate system to G is

$$\begin{aligned} \mathrm{R}_{\mathrm{LG}} &= \begin{bmatrix} \mathbf{e}_r \; \mathbf{e}_\phi \; \mathbf{e}_\beta \end{bmatrix} \\ &= \begin{bmatrix} \cos(\beta)\cos(\phi) & -\sin(\phi) & -\sin(\beta)\cos(\phi) \\ \cos(\beta)\sin(\phi) & \cos(\phi) & -\sin(\beta)\sin(\phi) \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix}, \end{aligned}$$

and the rotation matrix from G coordinate system to L is $\mathrm{R}_{\mathrm{GL}} = \mathrm{R}_{\mathrm{LG}}^{-1} = \mathrm{R}_{\mathrm{LG}}^{\top}$.

We consider the apparent wind velocity $\mathbf{v}_a = \mathbf{v}_w - \dot{\mathbf{p}}$ and assume that its radial component $\mathbf{v}_{a,r}$ is always strictly positive and that the kite body longitudinal axis is at all times aligned with the apparent wind velocity, that is $\mathbf{e}_1 = -\mathbf{v}_a/\|\mathbf{v}_a\|$. Let ψ be the roll angle measuring rotation around the $\mathbf{e}_1$ axis. We consider that $\tilde{\mathbf{e}}_2 = \mathbf{e}_2$ is initially (for $\psi = 0$) in the plane τ, tangent to a sphere centred at the origin of G (containing the axis $\mathbf{e}_\phi$ and $\mathbf{e}_\beta$). We have that $\tilde{\mathbf{e}}_2 \perp \mathbf{e}_r$, and $\tilde{\mathbf{e}}_2 \perp \mathbf{e}_1$. We can then define $\tilde{\mathbf{e}}_2 = \dfrac{\mathbf{e}_r \times \mathbf{e}_1}{\|\mathbf{e}_r \times \mathbf{e}_1\|}$. Finally, we consider that the kite body rotates anti-clockwise around the $\mathbf{e}_1$ axis and we assume that the roll angle ψ can be controlled directly. Since the kite has some mass, the roll angle cannot be selected arbitrarily at each instant and we would have to control the angular acceleration and consequently alter the angular velocity and roll angle. However, since the translational movement of the kite is much slower than its rotation in the defined operational range, we can consider, as a simplification, that ψ is a directly actuated control variable. Using Rodrigues' formula to rotate $\tilde{\mathbf{e}}_2$ by ψ around $\mathbf{e}_1$, we obtain

$$\mathbf{e}_2 = \tilde{\mathbf{e}}_2 \cos\psi + (\mathbf{e}_1 \times \tilde{\mathbf{e}}_2)\sin\psi + \mathbf{e}_1(\mathbf{e}_1 \cdot \tilde{\mathbf{e}}_2)(1 - \cos\psi) \tag{26.2}$$

and finally, we define $\mathbf{e}_3$ forming a right-handed coordinate system $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$. The total force acting on the kite can be decomposed as

$$m\ddot{\mathbf{p}} = \mathbf{F}^{\text{th}} + \mathbf{F}^{\text{grav}} + \mathbf{F}^{\text{aer}}(\alpha), \tag{26.3}$$

where each force is computed as follows:

$$\begin{aligned}
\mathbf{F}^{\text{th}} &= -T\ \mathbf{e}_r = \begin{bmatrix} -T \\ 0 \\ 0 \end{bmatrix}_{\mathrm{L}}, \\
\mathbf{F}^{\text{grav}} &= -mg\ \mathbf{e}_z = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}_{\mathrm{G}} = \begin{bmatrix} -mg\ \sin\beta \\ 0 \\ -mg\ \cos\beta \end{bmatrix}_{\mathrm{L}}, \\
\mathbf{F}^{\text{aer}}(\alpha) &= \frac{1}{2}\rho A \|\mathbf{v}_a\|^2 (c_{\mathrm{L}}(\alpha)\mathbf{e}_3 - c_{\mathrm{D}}(\alpha)\mathbf{e}_1).
\end{aligned}$$

In the local coordinate system

$$\ddot{\mathbf{p}} = \begin{bmatrix} \ddot{r} \\ r\ddot{\phi}\cos(\beta) \\ r\ddot{\beta} \end{bmatrix}_{\mathrm{L}} + \underbrace{\begin{bmatrix} -r\dot{\beta}^2 - r\dot{\phi}^2\cos^2(\beta) \\ 2\dot{r}\dot{\phi}\cos(\beta) - 2r\dot{\phi}\dot{\beta}\sin(\beta) \\ 2\dot{r}\dot{\beta} + r\dot{\phi}^2\cos(\beta)\sin(\beta) \end{bmatrix}_{\mathrm{L}}}_{-\frac{1}{m}\mathbf{F}^{\text{inert}}} \tag{26.4}$$

where the second term corresponds to $-\frac{1}{m}\mathbf{F}^{\text{inert}}$ with $\mathbf{F}^{\text{inert}}$ representing the inertial forces (centrifugal and Coriolis). Now, we can write

$$m\begin{bmatrix} \ddot{r} \\ r\ddot{\phi}\cos(\beta) \\ r\ddot{\beta} \end{bmatrix} = \mathbf{F}^{\text{th}} + \mathbf{F}^{\text{grav}} + \mathbf{F}^{\text{aer}}(\alpha) + \mathbf{F}^{\text{inert}} \tag{26.5}$$

We assume that the tether acceleration $\ddot{r}$ is directly controlled by a_t. Denoting by T the tension on the tether at the ground station, we have $T = \mathrm{F}_r - ma_t$. Defining the state $\mathbf{x} = \left(r, \phi, \beta, \dot{r}, \dot{\phi}, \dot{\beta}\right)$ and the control $\mathbf{u} = (a_t, \alpha, \psi)$, the dynamics of the system can be stated as

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} r \\ \phi \\ \beta \\ \dot{r} \\ \dot{\phi} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \dot{\phi} \\ \dot{\beta} \\ a_t \\ \dfrac{1}{mr\ \cos(\beta)}\mathrm{F}_\phi \\ \dfrac{1}{mr}\mathrm{F}_\beta \end{bmatrix}. \tag{26.6}$$

26.3 Optimal Control Problem

We consider the problem of optimizing power production during the traction phase (see e.g., [11–13] for a reference on optimal control and on the corresponding numerical methods). The instant power production is given by $P(t) = \dot{r}T$ and the energy in the interval $[t_0, t_f]$ is

$$E(t_f) = \int_{t_0}^{t_f} P(t)\ \mathrm{d}t\ . \tag{26.7}$$

We address the production cycle problem (P) that has a free terminal state, which is achieved when the tether is at the maximum length. Considering $t \in [t_0, t_f]$, the problem (P) can be stated as:

$$\text{Maximize} \int_{t_0}^{t_f} \dot{r}T\ \mathrm{dt} \tag{26.8}$$

subject to dynamic constraints

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}) \qquad \text{æ}\ t \in [t_0, t_f]$$

input constraints

$$\begin{aligned} a_{\min} \leq a_t(t) \leq a_{\max} &\qquad \text{a.e. } t \in [t_0, t_f] \\ \alpha_{\min} \leq \alpha(t) \leq \alpha_{\max} &\qquad \text{a.e. } t \in [t_0, t_f] \\ \psi_{\min} \leq \psi(t) \leq \psi_{\max} &\qquad \text{a.e. } t \in [t_0, t_f] \end{aligned}$$

the left end–point constraint

$$\mathbf{x}(t_0) = \mathbf{x}_0 = (r_0,\ \phi_0,\ \beta_0,\ \dot{r}_0,\ \dot{\phi}_0,\ \dot{\beta}_0)$$

and bounded–state constraints

$$\begin{aligned} r_{\min} \leq r(t) \leq r_{\max} \qquad & \forall t \in [t_0, t_f] \\ \phi_{\min} \leq \phi(t) \leq \phi_{\max} \qquad & \forall t \in [t_0, t_f] \\ \beta_{\min} \leq \beta(t) \leq \beta_{\max} \qquad & \forall t \in [t_0, t_f] \end{aligned}$$

26.4 Path–Following Control

26.4.1 Flight Path Specification

From the Optimal trajectory obtained in Sect. 26.3, we parametrize a path in the (ϕ, β) space that will be followed cyclically by the kite during the reel-out phase. The resulting trajectories tend to vary with different parametrizations of the problem, such as different wind speeds and kite characteristics. However, they usually follow an almost sinusoidal evolution of ϕ and β. The main frequencies of these variations can be obtained by a Fourier Transform and the ratio between both principal frequencies allows us to define a Lissajous curve for a time independent path. An example of the expected path can be the lemniscate curve:

$$\begin{cases} \phi = \phi_0 + \varDelta\phi \cos(t) \\ \beta = \beta_0 + \varDelta\beta \sin(2t). \end{cases} \tag{26.9}$$

In this case we are considering a Gerono lemniscate which is also a Lissajous figure with frequency ratio 1:2. This path is independent of the tether length r, that will vary during the traction phase, since it is only defined in the (ϕ, β) space. Its plot on a plane is given in Fig. 26.3.

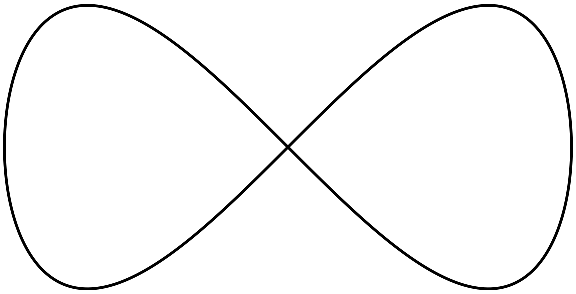


Fig. 26.3 A possible figure-of-eight Path in the (ϕ, β) space

26.4.2 Heading Angle Dynamics and Control

Heading angle dynamics and control are based on the proposed method in [14]. We aim to control the heading direction of the kite, through the roll angle, in order to follow a predefined path in the (ϕ, β) space. During the traction phase, the kite is expected to follow the desired path at high speed and, since the speed of the kite is typically much greater than the wind speed $\mathbf{v}_w$ or the reel-out speed $\dot{r}$, we may assume that the apparent wind velocity v_a and thus the longitudinal axis $\mathbf{e}_1$ are in the plane τ. Therefore, the angle between $\mathbf{e}_2$ and τ is similar to the roll angle. As shown in Fig. 26.4, the aerodynamic lift vector, aligned with $\mathbf{e}_3$, has a radial component and a component in τ, denoted by turning lift, which is responsible for the kite lateral acceleration defined as

$$a_\ell = \frac{1}{m}\mathbf{F}^{\text{lift}} \sin(\psi). \tag{26.10}$$

The controller proposed here is detailed in [14] and consists in a modification of the nonlinear guidance logic described in [15, 16]. The methods use a reference target approach to control the heading direction of the kite. Given the mass–point position of the kite $\mathbf{p}(\phi, \beta)$, we determine the closest point in the desired path Q, defining the cross–track distance between them as d. Then, a reference target point R is defined as the point distancing L from Q in a forward direction along the path and a vector $\mathbf{L_1}$ is defined as the vector from the kite position $\mathbf{p}$ to the reference point R (see Fig. 26.5). Finally, we compute the angle η between the kite velocity $\dot{p}$ and $\mathbf{L_1}$ that serves as a reference to the desired heading direction, so that the kite trajectory follows R. We act on the roll angle ψ in order to control the angle η towards zero. A simple proportional controller, $\psi(t) = K\eta(t)$ can be shown to be an adequate

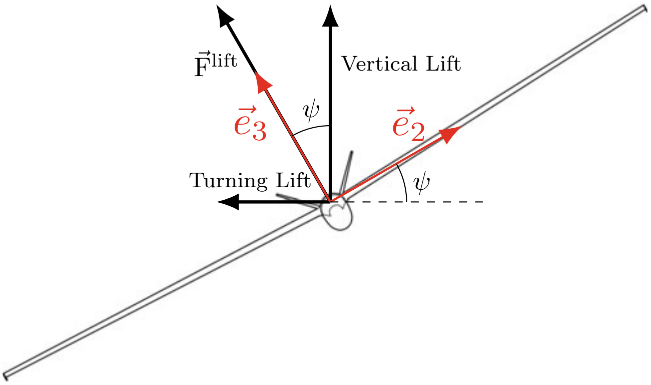


Fig. 26.4 Lift decomposition of a wing during a turn [14]

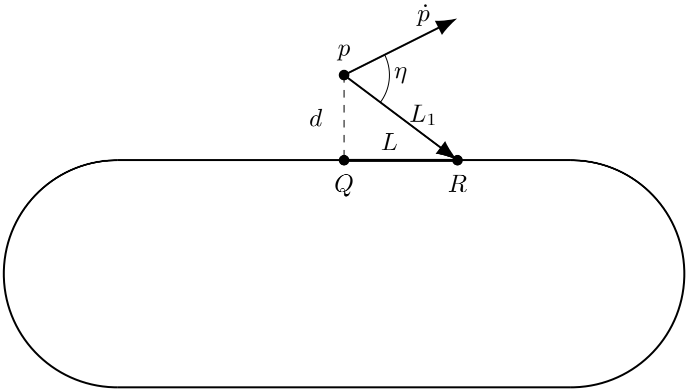


Fig. 26.5 Path to be followed and signals involved [14]

steering command, see [17, 18]. As in [15], our command is based on computing the needed lateral acceleration, which is the required centripetal acceleration for the kite to follow a circular trajectory from the current position towards the reference point. This sideways acceleration is given by

$$a_s = 2\frac{V^2}{L_1}\sin(\eta). \tag{26.11}$$

From Eqs. (26.10) and (26.11), we obtain

$$\psi = \arcsin\left(2\,m\frac{V^2\sin(\eta)}{\mathbf{F}^{\text{lift}}L_1}\right). \tag{26.12}$$

Since the range for the possible values of the roll angle is limited ($\psi \in [-\psi_{\max}, \psi_{\max}]$), the control with saturation is given by

$$\psi = \min\left\{\psi_{\max}, \max\left\{-\psi_{\max}, \arcsin\left(2\,m\frac{V^2\sin(\eta)}{\mathbf{F}^{\text{lift}}L_1}\right)\right\}\right\}. \tag{26.13}$$

26.5 Numerical Results

All simulations were carried out using the dynamical model (26.6) implemented in Simulink. We consider the parameters of simulation for the kite power system (KPS) with a small aircraft with 0.7 kg and 0.28 m^2 of wing area. The OCP problem, reported in Sect. 26.3, is solved using direct methods, namely using ICLOCS (Imperial College of London Optimal Control Software) interface coupled with IPOPT (Interior Point OPTimizer) solver. Then, the obtained solution is used to describe geometrically the

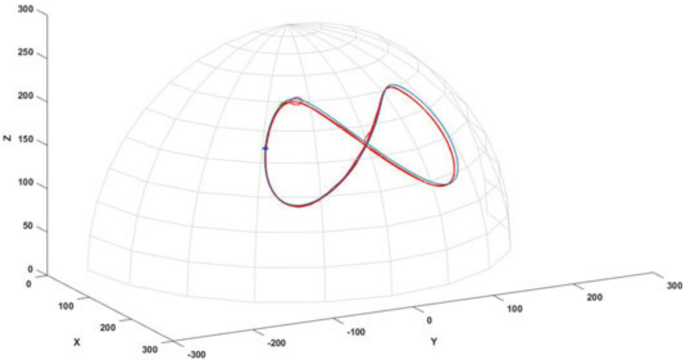


Fig. 26.6 Kite trajectory, figure–of–eight shape, for fixed tether length

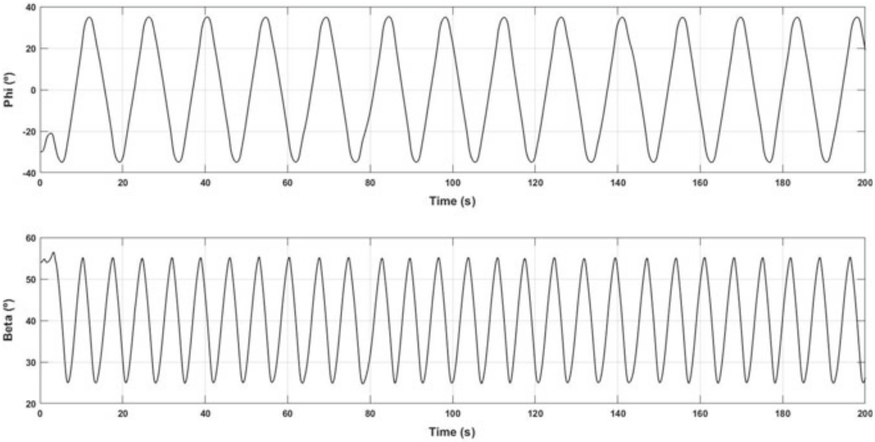


Fig. 26.7 Variation of ϕ and β over time during a figure–of–eight shape, for fixed tether length

desired optimal path. Finally, simulation and testing of the guidance control strategy for path-following, using the computed optimal path as a reference, is carried out as described in Sect. 26.4.

The controller drives the system to follow a pre-defined figure–of–eight shape path.

Firstly, we consider the case of a fixed tether length. This case is typical in a fly–gen AWES, i.e. a system with on–board generation. In Fig. 26.6, the blue line is the path to follow while the solid red line is the kite trajectory. As it can be seen, when applying our controller, the kite trajectory closely follows the desired pre–defined path, even when we simulate several cycles. Figure 26.7 shows the evolution of ϕ and β angles over time and Fig. 26.8 displays a closer look of part of the simulation in which the reference point evolution is also presented.

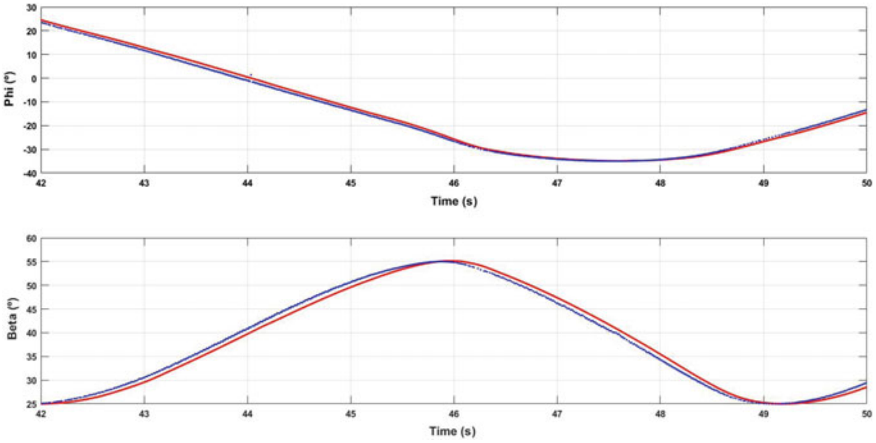


Fig. 26.8 Detail of path-following control: the red line represents the variation of ϕ and β of the kite position and the blue dots are relative to the reference points

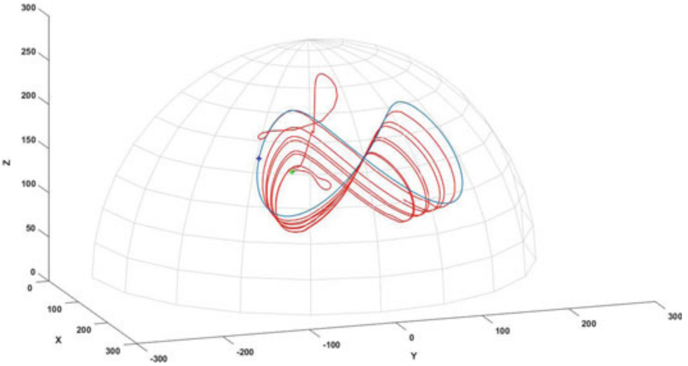


Fig. 26.9 Kite trajectory during a complete cycle

In a second case, we consider a varying tether length and a complete cycle comprising production and retraction phases. This case is typical of a ground–gen system, i.e. a system with ground–based generation. In Fig. 26.9, once again, the blue line is the path to follow while the red line is the kite trajectory. As it can be seen, when applying our controller during different cycle phases, the kite trajectory successfully follows the desired pre–defined path and when the tether reaches its maximum length the kite is reeled–in after being driven to the zenith position. Figure 26.10 depicts the evolution of ϕ and β angles over time during this simulation. In both simulations, the results confirm the performance of such controller for the kite to follow closely the pre–defined path.

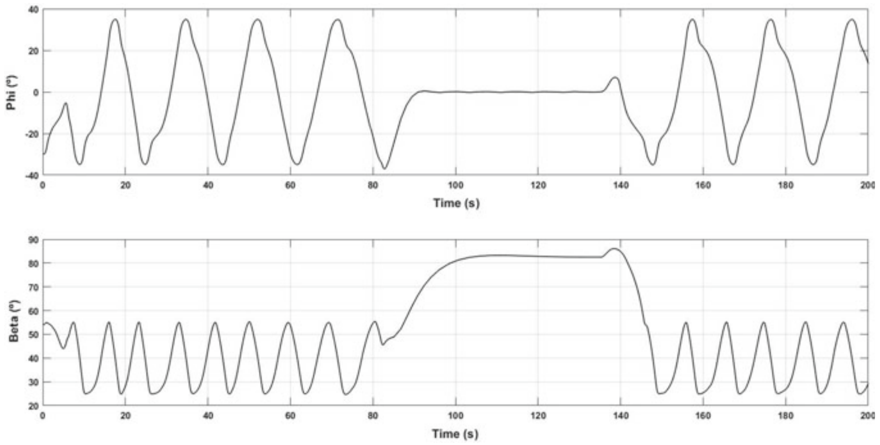


Fig. 26.10 Variation of ϕ and β over time for a complete cycle

26.6 Conclusion and Future Work

In this paper, we have performed an open-loop OCP in order to find the trajectory that would maximize power production and then applied a non-linear guidance control to follow a desired path. In the future, we will aim at closing the loop in order to apply real time optimization based control strategies, such as Model Predictive Control (MPC), both for a new path-following strategy but also to constantly re-define the best path to follow, applying Economical MPC methods.

Acknowledgments In this research, we acknowledge the support of FEDER/COMPETE2020/-NORTE2020/POCI/PIDDAC/ MCTES/FCT funds through grants SFRH/BPD/126683/ 2016, NORTE-01-0145-FEDER-000033–Stride, UID/IEEA/00147/006933–SYSTEC, and POCI-01-0145-FEDER-031447–UPWIND.

References

1. Cherubini A, Papini A, Vertechy R, Fontana M (2015) Airborne wind energy systems: a review of the technologies. Renew Sustain Energy Rev 51:1461–1476
2. Ahrens U, Diehl M, Schmehl R (eds) (2013) Airborne wind energy. Green energy and technology. Springer, Berlin
3. Schmehl R (ed) (2018) Airborne wind energy: advances in technology development and research. Green energy and technology. Springer, Singapore
4. Loyd ML (1980) Crosswind kite power. J Energy 4(3):106–111
5. Paiva LT, Fontes FACC (2018) Optimal electric power generation with underwater kite systems. Computing 100:1137–1153
6. Fernandes MC (2018) Airborne wind energy systems: modelling, simulation and economic analysis. Master's thesis, Universidade do Porto, Porto, June 2018
7. Faggiani P (2014) Pumping kites wind farm. Master's thesis, TU Delft, Netherlands

8. Paiva LT, Fontes FACC (2018) Optimal control algorithms with adaptive time-mesh refinement for kite power systems. Energies 11:475
9. Prodan I, Olaru S, Fontes FACC, Pereira FL, de Sousa JB, Maniu CS, Niculescu S-I (2015) Predictive control for path-following. from trajectory generation to the parametrization of the discrete tracking sequences. Developments in model-based optimization and control. Lecture notes in control and information sciences. Springer, Cham, pp 161–181
10. Caldeira AC, Fontes FA 2010) Model predictive control for path-following of nonholonomic systems. In: Proceedings of the 10th Portuguese conference on automatic control - CONTROLO 2010, Coimbra, Portugal, September 2010, pp 720–725
11. Vinter RB (2000) Optimal control. Springer
12. Betts JT (2001) Practical methods for optimal control using nonlinear programming. SIAM
13. Gerdts M (2012) Optimal control of odes and daes. De Gruyter
14. Silva GB, Paiva LT, Fontes FA (2019) A path—following guidance method for airborne wind energy systems with large domain of attraction. In: Proceedings of the 2019 American control conference - ACC'19, June 2019
15. Park S, Deyst J, How J (2004) A new nonlinear guidance logic for trajectory tracking. In: AIAA guidance, navigation, and control conference and exhibit. American Institute of Aeronautics and Astronautics
16. Park S, Deyst J, How JP (2007) Performance and Lyapunov stability of a nonlinear path following guidance method. J Guid Control Dyn 30:1718–1728
17. Fagiano L, Zgraggen AU, Morari M, Khammash M (2014) Automatic crosswind flight of tethered wings for airborne wind energy: modeling, control design, and experimental results. IEEE Trans Control Syst Technol 22:1433–1447
18. Fernandes MC, Silva GB, Paiva LT, Fontes FA (2018) A trajectory controller for kite power systems with wind gust handling capabilities. In: Proceedings of 15th international conference on informatics in control, automation and robotics (ICINCO), Porto

Chapter 27
Temperature Time Series Forecasting in the Optimal Challenges in Irrigation (TO CHAIR)

A. Manuela Gonçalves, Cláudia Costa, Marco Costa, Sofia O. Lopes, and Rui Pereira

Abstract Predicting and forecasting weather time series has always been a difficult field of research analysis with a very slow progress rate over the years. The main challenge in this project—The Optimal Challenges in Irrigation (TO CHAIR)—is to study how to manage irrigation problems as an optimal control problem: the daily irrigation problem of minimizing water consumption. For that it is necessary to estimate and forecast weather variables in real time in each monitoring area of irrigation. These time series present strong trends and high-frequency seasonality. How to best model and forecast these patterns has been a long-standing issue in time series analysis. This study presents a comparison of the forecasting performance of TBATS (Trigonometric Seasonal, Box-Cox Transformation, ARMA errors, Trend and Seasonal Components) and regression with correlated errors models. These methods are chosen due to their ability to model trend and seasonal fluctuations present in weather data, particularly in dealing with time series with complex seasonal

A. M. Gonçalves (✉)
Department of Mathematics and Centre of Mathematics, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: mneves@math.uminho.pt

C. Costa
Department of Mathematics, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: claudiacosta_mf@hotmail.com

M. Costa
Águeda School of Technology and Management and Centre for Research and Development in Mathematics and Applications, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
e-mail: marco@ua.pt

S. O. Lopes · R. Pereira
Department of Mathematics and Centre of Physics, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: sofialopes@math.uminho.pt

R. Pereira
e-mail: rmp@math.uminho.pt

A. Gaspar-Cunha et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 55, https://doi.org/10.1007/978-3-030-57422-2_27

patterns (multiple seasonal patterns). The forecasting performance is demonstrated through a case study of weather time series: minimum air temperature.

Keywords Irrigation · Temperature · Time series · Modeling · Forecasting · TBATS · Regression with correlated errors

27.1 Introduction

In a world where climate change and increasing social conflicts are a reality, a proper management of the existing scarce resources is vital. Thus, we will try to find the best technical solutions to improve the efficiency of their use in response to environmental concerns. Most irrigation systems on sale in the market are based on control with no prediction techniques. The excess of water in the soil, which is frequently a result of these techniques, is responsible for significant water waste. Understanding the behaviour of humidity in the soil by mathematical/statistical modeling allows, among others, an efficient planning of water use via irrigation systems [1].

According to IPMA, the Portuguese Institute for the Ocean and Atmosphere (September 30, 2017), about 81% of Portugal's mainland was in severe drought, 7.4% in extreme drought, 10.7% in moderate drought and 0.8% in weak drought. 2017 was an extremely dry year and, considering the data from January 1st, 2017 to December 27th, 2017, it will be among the 4 driest years since 1931 (all occurred after 2000), and the average annual total precipitation will be about 60% of what is deemed normal. The period from April to December, with persistently negative precipitation abnormalites, will be deemed the driest of the last 87 years. In the media, several news mention the various problems that Portugal has to face, such as producing animal feed, supplying water to the population and, very important, the lack of water for agriculture purposes. Water resources are mainly used in agriculture: about 70% of freshwater is used in agriculture. Consequently, there is much that can be done to save water, and this is of the utmost importance for our planet. Therefore, the main challenge in project The Optimal Challenges in Irrigation (TO CHAIR) is to study how to manage irrigation problems as an optimal control problem: the daily irrigation problem of minimizing water consumption. For that it is necessary to estimate and forecast weather variables (like minimum air temperature) in real time in each irrigation area, in order to determine, in particular, the evapotranspiration (related to the irrigation planning problem). Our data source are the records of the variable minimum air temperature observed in a farm located in Vila Real County, in northern Portugal, in the field of agriculture irrigation, registered in the period from January 23rd, 2015 to August 11th, 2018 on a daily basis. The main goal is to forecast these environmental variables at a location (in this case, at the farm), where there are historical observations but current measurements are not available (including various steps for forecasting (i.e., 7 days)).

27.2 Methodology

A time series is an ordered sequence of values of a variable at equally spaced time intervals, in this case daily minimum air temperature at a weather station. Time series forecasting is an important area in which past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. The model is then used to extrapolate the time series into the future. Forecasting methods are a key tool in decision-making processes in many areas, such as economics, agriculture, management or environment. There are several approaches to modeling time series, but we decided to study and to compare the accuracy of the TBATS model and the regression with correlated errors model for forecasting weather/meteorological time series, because both models can increase the chance of capturing the proprieties and the dynamics in the data and improving accurate forecasts. Both methods have the ability to deal with time series with high-frequency seasonality.

The time series analysis (of both processes) was carried out using the statistical software R programming language and specialized packages for modeling and forecasting [2].

27.2.1 TBATS

TBATS model is a time series model for series demonstrating multiple/complex seasonality. The TBATS model was introduced by De Livera, Hyndman and Snyder (2011, JASA). TBATS is an abbreviation denoting its salient types: T for trigonometric regressors to model multiple seasonality, B for Box-Cox transformations, A for ARMA errors (autoregressive moving average), T for trend and S for seasonality [3, 4]. The model including a Box-Cox transformation (the notation $y_t^{(w)}$ is used to represent Box-Cox transformed observations with the parameter w, where y_t is the observation at time t), ARMA errors, and T seasonal patterns is as follows:

$$y_t^{(w)} = \begin{cases} \frac{y_t^w - 1}{w}, w \neq 0 \\ \log y_t, w = 0 \end{cases}$$

$$y_t^{(w)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{T} S_{t-m_i}^{(i)} + d_t \tag{27.1}$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = (1 - \phi) b + \phi b_{t-1} + \beta d_t$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t \tag{27.2}$$

$$d_t = \sum_{i=1}^{p} \varphi_i d_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

where $m_1, \ldots, m_T$ denote the seasonal periods, ℓ_t is the local level in period t, b is the long-run trend, b_t is the short-run trend in period t, $s_t^{(i)}$ represents the i th seasonal component at time t, d_t denotes an ARMA (p, q) process and ε_t is a Gaussian white noise process with zero mean and constant variance σ^2. The smoothing parameters are given by α, β and γ_i for $i = 1, \ldots, T$. ϕ is the damping parameter representing the damped trend (the damping factor is included in the level and measurement equations as well as the trend equation).

To introduce the trigonometric representation of seasonal components based on Fourier series (trigonometric representation of seasonal components based on Fourier series), [5, 6], the $s_t^{(i)}$ can be rewritten as follows:

$$s_t^{(i)} = \sum_{j=1}^{k_i} S_{j,t}^{(i)}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t$$

$$S_{j,t}^{*(i)} = -s_{j,t-1} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$$

where $\gamma_1^{(i)}$ and $\gamma_2^{(i)}$ are smoothing parameters and $\lambda_j^{(i)} = 2\pi j / m_i$. The stochastic level of the ith seasonal component is represented by $s_{j,t}^{(i)}$, and the stochastic growth in the level of the ith seasonal component that is needed to describe the change in the seasonal component over time by $S_{j,t}^{*(i)}$. The number of harmonics required for the ith seasonal component is denoted by k_i. The approach is equivalent to index seasonal approaches when $k_i = m_i/2$ for even values of m_i, and when $k_i = (m_i - 1)/2$ for odd values of m_i. It is anticipated that most seasonal components will require fewer harmonics, thus reducing the number of parameters to be estimated. A deterministic representation of the seasonal components can be obtained by setting the smoothing parameters equal to zero.

A new class of innovations state space models is obtained by replacing the seasonal component $s_t^{(i)}$ in Eqs. (27.1) and (27.2) by the trigonometric seasonal formulation, and the measurement equation by

$$y_t^{(w)} = \ell_{t-1} + \phi b_{t-1} + \sum\nolimits_{i=1}^{t} S_{t-1}^{(i)} + d_t$$

This class is designated by TBATS, the initial T connoting "trigonometric". To provide more details about their structure, this identifier is supplemented with relevant arguments to give the designation

$$\text{TBATS}\ (\omega, \phi, p, q, \{m_1, k_1\}\{m_2, k_2\}, \ldots, \{m_T, k_T\})$$

A TBATS model requires the estimation of $2(k_1 + k_2 + \ldots + k_T)$ initial seasonal values.

27.2.2 Regression Model with Correlated Errors

An important approach in forecasting time series, particularly meteorological time series, involves fitting regression models (RM) to time series including trend and seasonality components. The RM models are originally based on linear modeling, but they also allow parameters such as trend and season to be added to the data. In this study, the trend parameter will be fitted with polynomial function, and the season parameter will be estimated with Fourier series. But for the RM to be validated, the error terms must be a sequence of uncorrelated and Gaussian (with mean 0 and variance constant).

Also, one of the most popular ways of time series modeling is autoregressive integrated moving average (ARIMA) modeling introduced by Box and Jenkins in 1960s to forecast time series [7]. An ARIMA(p, d, q) model can account for temporal dependence in several ways. Firstly, the time series is d-differenced to render it stationary. If $d = 0$, the observations are modeled directly, and if $d = 1$, the differences between consecutive observations are modeled. Secondly, the time dependence of the stationary process $\{Y_t\}$ is modeled by including p autoregressive models. Thirdly, q are moving average terms, in addition to any time-varying covariates. It takes the observation of previous errors. Finally, by combining these three models, we get the ARIMA model. Thus, the general form of the ARIMA models is given by:

$$\mathrm{y_t} = \mathrm{c} + \sum_{\mathrm{i=1}}^{\mathrm{p}} \varphi_{\mathrm{i}} \mathrm{y_{t-i}} + \sum_{\mathrm{j=1}}^{\mathrm{q}} \theta_{\mathrm{j}} \varepsilon_{\mathrm{t-j}}$$

where $\mathrm{y_t}$ is a stationary stochastic process, c is the constant that determines the level of the time series, ε_t is the error or white noise disturbance term, φ_i means autoregressive coefficient and θ_j is the moving average coefficient. For a seasonal time series, these steps can be repeated according to the period of the cycle, whatever time interval.

A regression model (RM) with correlated errors in which are incorporated external regressors in the form of Fourier terms (to account for the seasonal behavior) are added to an ARIMA (p, d, q) model [8]. These models are regression models (RM) which include a correction for autocorrelated errors, [9, 10]. Hence, we can add ARIMA terms to the regression model to eliminate the autocorrelation. To do this, we re-fit the regression model as an ARIMA (p, d, q) model with regressors, and specify the appropriate AR (p) or MA (q) terms to fit the pattern of autocorrelation we observed in the original residuals.

So, in this regression model we apply the Fourier series to model seasonal pattern by using Fourier terms with short-term time series dynamics allowed in the error, and we consider the following model:

$$y_t = c + \sum_{k=1}^{K} \left[\alpha_k sin \frac{2\pi kt}{m} + \beta_k cos \frac{2\pi kt}{m} \right] + e_t \tag{27.3}$$

where e_t is an ARIMA process, α_k and β_k are Fourier coefficients and m is a length of period. The value of K is chosen by minimizing forecast error measures [11].

27.2.3 Forecast Error Measures

Let's denote the actual observation for time period t by y_t and the estimated or forecasted value for the same period by $\hat{y}_t$, and n is the total number of observations. The most commonly used forecast error measures are the mean error (ME), the root mean squared error (RMSE), and the mean absolute error (MAE). They are defined by the following formulas, respectively:

$$ME = \frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2}$$

$$MAE = \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t|$$

When comparing the performance of forecast methods on a single dataset, the MAE is interesting for it is easy to understand, but the RMSE is more valuable because it is more sensitive than other measures to the occasional large error (the squaring process gives disproportionate weight to very large errors).

The MASE was proposed by Hyndman and Koehler (2006) for comparing forecast accuracies. The MASE is given by the formula:

$$\mathrm{MASE} = \frac{\mathrm{MAE}}{\mathrm{Q}}$$

where Q is a scaling statistic. For a seasonal time series, a scaling statistic can be defined using the seasonal naïve forecasts:

$$Q = \frac{1}{n-m} \sum_{j=m+1}^{n} \left| y_j - y_{j-m} \right|$$

where the seasonal naïve method accounts for seasonality by setting each prediction to be equal to the last observed value of the same season.

When comparing forecasting methods, the method with lowest ME, RMSE, MAE, or MASE is the preferred one. Frequently, different accuracy measures will lead to different results as to which forecast method is the best, [12, 13].

27.3 Minimum Air Temperature Forecasting

27.3.1 Data

In the present study, we focus on a minimum temperature dataset. Figure 27.1 shows the time series distribution in the total observed period: between January 23rd, 2015 to August 11th, 2018 (1327 days). The graphical representation clearly shows that time series exhibits seasonal behaviour, as is expected due to the environmental nature of the data. The daily data exhibits a strong annual seasonality (a period of 365 days, because we have excluded the days 29th of February during the period under observation) with extreme values in cold seasons. Moreover, the variation seems to be also larger in cold seasons than in warm ones.

Table 27.1 presents descriptive of statistics for the minimum temperature time series during the observed period by month. As expected, the minimum temperature is higher in the summer months and presents lower values in the winter months. The monthly standard deviations (SD) indicate a larger variability during the months of November, December, January and February. Minimum air temperature is characterised by a symmetric distribution (presenting values near zero) by month.

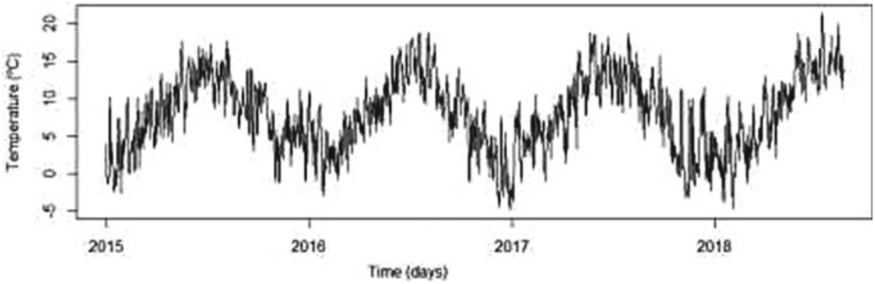


Fig. 27.1 Daily time series of minimum air temperature distribution in the farm during the observed period

Table 27.1 Descriptive statistics of the daily Minimum Air Temperature distribution by month

Months	Range	Mean	SD	Skewness	Nº Days
January	−4.90–11.60	3.03	4.02	0.14	102
February	−4.60–−10.40	2.89	3.39	0.06	112
March	−1.30–10.50	4.20	2.61	0.08	124
April	0.30–13.10	6.71	2.50	0.02	120
May	1.40–16.40	9.34	2.82	0.20	124
June	5.40–18.90	12.36	2.99	0.17	120
July	8.70–18.30	13.95	2.29	0.12	124
August	7.60–21.40	13.83	2.71	0.22	124
September	4.70–20.00	11.06	2.93	0.54	101
October	3.00–15.80	0.20	2.86	0.06	93
November	−2.10–12.10	4.80	3.62	0.27	90
December	−2.00–10.20	3.29	3.57	-0.07	93

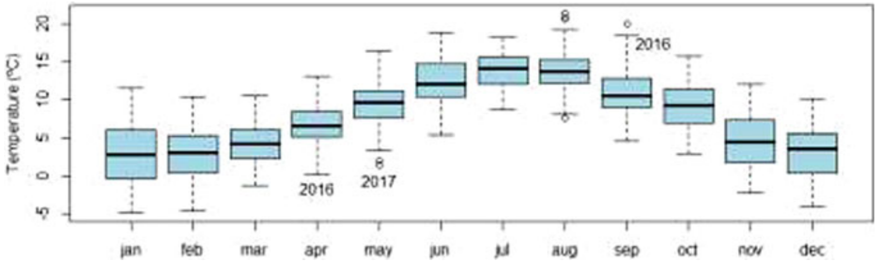


Fig. 27.2 Box-plots of the daily distribution of Minimum Air Temperature by month in the observed period

Figure 27.2 presents box-plots of daily minimum temperature by month. The box-plots are able to identify some moderate outliers in some months (April, May, August, and September).

27.4 Results

The results obtained from the application of TBATS and RM with correlated errors methods are reported in this section. The methods considered in this study are applied to two sets: training data (in-sample data) and testing data (out-of-sample data) in order to testify the accuracy of the proposed forecasting models. The selected training period was from January 23rd, 2015 to January 22nd, 2018 (first 1095 observations) and was used in order to fit the models to data, and the test period with the last 232 observations (period between January 23rd, 2018 to August 11th, 2018) was used to forecast. This approach gives the ability to compare the effectiveness of different methods of prediction.

27.4.1 TBATS

The minimum air temperature data are observed daily and show a strong annual seasonal pattern, so the length of seasonality of the time series is $m_1 = 365$. The time series exhibits an upward additive trend and an additive seasonal pattern, that is, a pattern for which the variation does change with the level of the time series.

As a second step, an ARMA model was fitted to the residuals with (p, q) combinations, and it was discovered that the TBATS $(1, 1, 0, 4, \{365, 3\})$ model minimizes the AIC. AIC is known as the Akaike's information criteria. The estimated parameters for the TBATS model are shown in Table 27.2. No Box-Cox transformation is necessary for this time series (so, $w = 1$). The estimated values of 0 for β and 1 for ϕ imply a purely deterministic growth rate with no damping effect. The model also implies that the irregular component of the series is correlated and can be described by an ARMA (0,4) process, and that a strong transformation is not necessary to handle nonlinearities in the series.

In the final model, $\sigma = 2.3109$ and the AIC is 9533.843.

In Fig. 27.3 are represented the original values of the air minimum temperature, the estimates in the modeling period (training period), the forecasts in the forecasting period (testing period) and the forecast intervals for a confidence level of 90% and 95% by applying the TBATS model.

Table 27.2 Estimated parameters for application of the TBATS method

Parameters estimates									
w	ϕ	α	β	γ_1	γ_2	θ_1	θ_2	θ_3	θ_4
1	1	0.0133	0	−1.4580e–05	−4.6799e–05	0.6868	0.3734	0.2035	0.0339

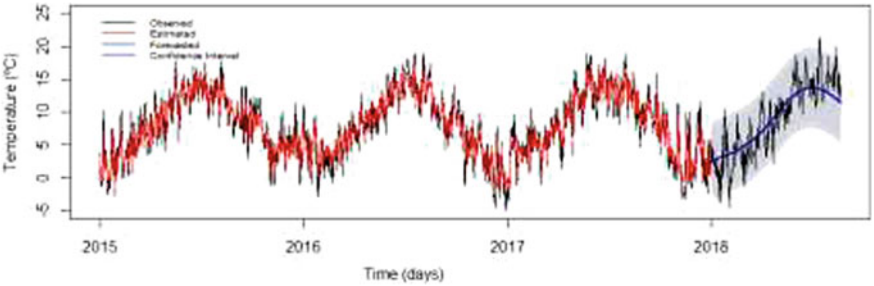


Fig. 27.3 Observed estimates and forecasts (with 90 and 95% confidence bounds) for minimum temperature time series using TBATS model

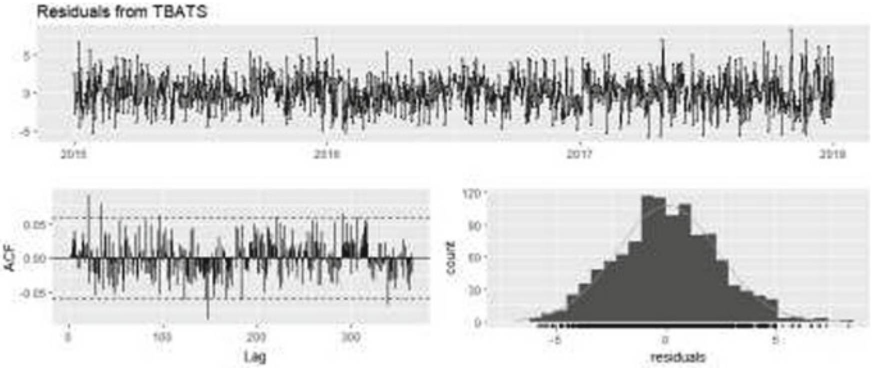


Fig. 27.4 Residuals time series, autocorrelation function and histogram of residuals (TBATS model)

The model validation was assessed by means of the residuals analysis (Fig. 27.4). The independency assumption was assessed by estimating the autocorrelation and the partial autocorrelation functions of residuals and the assumption that the residuals are identically normally distributed was also verified (by performing the Kolmogorov-Smirnov test).

27.4.2 Regression Model with Correlated Errors

In this study, we use the regression model in the basic form $y_t = b_t + s_t + e_t$, where b_t and s_t represent the trend and the seasonal components of the time series at time t, respectively. We apply the Fourier series to model the seasonal component (Eq. 27.3). The value of K can be chosen by minimizing predictions errors (minimizing AIC).

We consider the model with data having a long seasonal period (365 for daily data, i.e., $m = 365$). To choose the best RM, we ran the model by varying K, and the smallest forecast errors was when $K = 2$.

The analysis of residuals indicates the existence of a temporal correlation structure in the residuals. The minimum of AIC where $\mathrm{e_t}$ is an ARIMA (p, d, q) process, in this case an ARIMA (2, 0, 0), i.e., an AR (2) process.

Table 27.3 presents the estimated parameters and the respective standard errors, for the RM with AR (2) errors.

In the final model, $\sigma = 2.3130$ and the AIC is 4953.53.

In Fig. 27.5 are represented the original values of the air minimum temperature, the estimates in the modeling period (training period), the forecasts in the forecasting period (testing period) and the forecast intervals for a confidence level of 90 and 95% by applying the regression model with correlated errors (AR(2)).

Table 27.3 Estimated parameters for the application of the RM with correlated errors, and the correspondent standard errors

Parameters estimates						
c	ϕ_1	ϕ_2	α_1	α_2	β_1	β_2
7.9060	0.6903	−0.0911	−5.5754	0.4597	−0.0691	0.1797
Standard error of parameters estimates						
se (c)	$se(\phi_1)$	se(ϕ_2)	$se(\alpha_1)$	$se(\alpha_2)$	$se(\beta_1)$	$se(\beta_2)$
0.1737	0.0301	0.0301	0.2452	0.2450	0.2458	0.2455

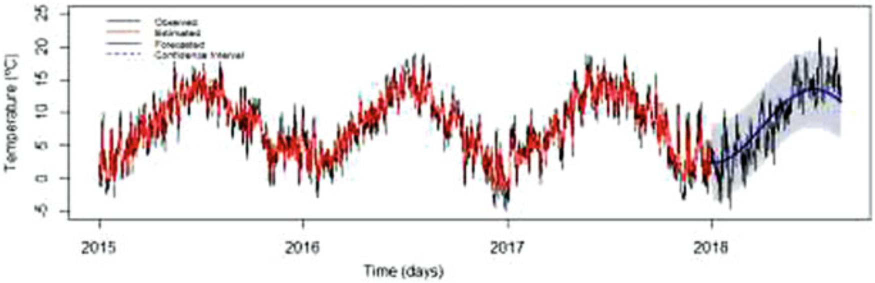


Fig. 27.5 Observed, estimates and forecasts (with 90 and 95% confidence bounds) by using RM with correlated errors

The regression model with correlated errors model validation was assessed by means of the residuals analysis (by applying the same assumptions made in the model validation of TBATS), Fig. 27.6.

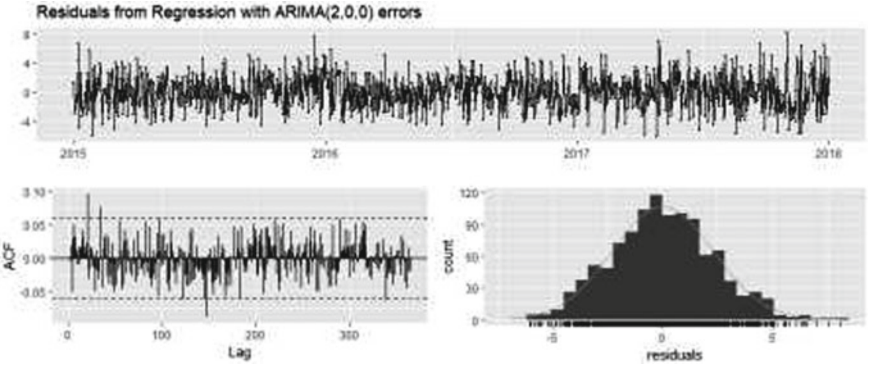


Fig. 27.6 Residuals time series, autocorrelation function and histogram of residuals (RM with correlated errors)

Table 27.4 Forecasting performance evaluation of TBATS and RM with correlated errors of minimum temperature time series

Model	ME	RMSE	MAE	MASE	
TBATS	0.0334	2.3110	1.8339	0.9040	Training period
RM with AR(2)	−0.0010	2.3056	1.8376	0.9059	Training period
TBATS	0.1766	2.9284	2.3395	1.1533	Testing period
RM with AR(2)	0.0229	2.8688	2.3006	1.1342	Testing period

27.4.3 Models Performance

The residuals performance in both processes modeling is consistent with the white noise process as seen in Figs. 27.4 and 27.6, so we can conclude the validity and adequacy of the two fitted models.

Table 27.4 shows the result of the accuracy measures calculated for training and testing periods for the two methods applied to the time series under study. The performance comparisons of the competing models (TBATS and RM with correlated errors) were evaluated using ME, RMSE, MAE, and MASE. The results obtained showed that the regression model with correlated errors, which requires fewer parameters to be estimated, is more accurate than TBATS, and performs better for all period times (training and test periods).

From the two models performed, we selected the most adequate model which has the lowest forecast error when comparing predicted data using a suitable test set: regression model with correlated errors. Therefore, RM with correlated errors can more efficiently capture the dynamic behaviour of the weather property, minimum air temperature, compared to TBATS.

27.5 Conclusions

In this study, we have shown that both TBATS and RM with correlated errors (for forecasting time series with complex seasonal patterns) can efficiently capture the behaviour of air temperature in the studied site. The obtained results show that the application of TBATS and RM with correlated errors methods to the minimum air temperature provides valuable insights into the studied data structures and their components, being a good basis for accurate estimations and forecasts. However we have to further explore the features of the two models, and we need to investigate more.

Our preliminary findings show that, in this case (minimum air temperature at this farm) RM with correlated errors is better than the TBATS model for forecasting, because within the scope of TO CHAIR project we intend to obtain accurate forecasts of this weather variable for the following 7 days in the farm in order to solve the irrigation problems.

Acknowledgments This research was partially funded by Portuguese funds through the Center for Research and Development in Mathematics and Applications (CIDMA) and the Portuguese Foundation for Science and Technology ("Fundação para a Ciência e a Tecnologia" – FCT), within the framework of project UID/MAT/04106/2019. This research was partially funded by Portuguese funds through the Portuguese Foundation for Science and Technology ("Fundação para a Ciência e a Tecnologia" – FCT), within the framework of project UID/MAT/00013/2013. FEDER/COMPETE/-NORTE2020/POCI//FCT funds through grants PTDC-EEI-AUT-2933-2014116858-TOCCATA, and TOCHAIR-POCI-01-0145-FEDER-028247 Financial support from the Portuguese Foundation for Science and Technology (FCT) in the framework of the Strategic Financing UIDIFIS/04650/2013 is also acknowledged. This work has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734205 – H2020-MSCA-RISE-2017.

References

1. Lopes SO, Fontes FACC, Pereira RMS, de Pinho MDR, Gonçalves AM (2016) Optimal control applied to an irrigation planning problem. Math Probl Eng 10. https://doi.org/10.1155/2016/5076879
2. R core Team (2018) R: a language and environment for statistical computing. Version 3.5.0 Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/
3. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast 22(4):679–688
4. Hyndman RJ, Koehler AB, Ord JK, Snyder RD (2008) Forecasting with exponential smoothing: the state space approach. Springer, Berlin
5. Harvey A (1989) Forecasting structural time series models and the Kalman filter. Cambridge University Press, New York
6. West M, Harrison J (1997) Bayesian forecasting and dynamic models, 2nd ed. Springer, New York
7. Box GEP, Jenkins G (1970) Time Series Analysis: forecasting and control. Holden-Day, San Francisco
8. Murat M, Malinowska I, Gos M, Krzyszczak J (2018) Forecasting daily meteorological time series using ARIMA and regression models. Int Agrophysics 32:253–264
9. Alpuim T, El-Shaarawi A (2009) Modeling monthly temperature data in Lisbon and Prague. Environmetrics 20:835–852
10. Alpuim T, El-Shaarawi A (2008) On the efficiency of regression analysis with AR(p) errors. J Appl Stat 35(7):717–737
11. Costa M, Monteiro M (2017) Statistical modeling of an air temperature time series of European cities. In: Daniels JA (ed) Advances in environmental research, vol 59. Nova Science Publishers, New York, 182–197
12. Costa M, Gonçalves AM (2011) Clustering and forecasting of dissolved oxygen concentration on a river basin. Stoch Env Res Risk Assess 25(2):151–163
13. Ye L, Yang G, Van Ranst E, Tang H (2013) Time-series modeling and prediction of global monthly absolute temperature for environmental decision making. Adv Atmos Sci 30:382–396