JYVÄSKYLÄ STUDIES IN COMPUTING
162

Hong Wang

Evolutionary Design Optimization with Nash Games and Hybridized Mesh/Meshless Methods in Computational Fluid Dynamics

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 162

Hong Wang

Evolutionary Design Optimization with Nash Games and Hybridized Mesh/Meshless Methods in Computational Fluid Dynamics

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 2
joulukuun 18. päivänä 2012 kello 10.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 2, on December 18, 2012 at 10 a.m.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2012

Evolutionary Design Optimization with Nash Games and Hybridized Mesh/Meshless Methods in Computational Fluid Dynamics

JYVÄSKYLÄ STUDIES IN COMPUTING 162

Hong Wang

Evolutionary Design Optimization with Nash Games and Hybridized Mesh/Meshless Methods in Computational Fluid Dynamics

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2012

Editors
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-5007-1
ISBN 978-951-39-5007-1 (PDF)

ISBN 978-951-39-5006-4 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2012

ABSTRACT

WANG, Hong
Evolutionary Design Optimization with Nash Games and Hybridized Mesh /Meshless Methods in Computational Fluid Dynamics
Jyväskylä: University of Jyväskylä, 2012, 68 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 162)
ISBN 978-951-39-5006-4 (nid.)
ISBN 978-951-39-5007-1 (PDF)
Finnish summary
Diss.

This research focuses on computational intelligent systems for solving computational fluid dynamics optimization problems.

Several approaches are used to discretize two-dimensional Euler equations: finite volume methods, least-squares meshless methods and hybridized mesh /meshless methods. A new fast meshless method coupled with ad hoc artificial dissipation parameters is developed. Numerical experiments show the flexibility, accuracy and efficiency of the fast meshless method for solving aerodynamics problems. A fast dynamic cloud technique based on a Delaunay graph mapping strategy is proposed to move clouds of points in the computational domain for unsteady simulations, position reconstructions and shape optimizations.

Evolutionary algorithms have become more popular in solving design optimization problems in industrial environments. In that direction, hybridized game coalitions coupled with evolutionary algorithms for optimization are opening avenues to efficiently capture solutions of complex design problems. Genetic algorithms coupled with Nash games and hierarchical genetic algorithms are investigated to position reconstructions and shape optimizations. The performance and accuracy of the above computational intelligent systems are evaluated and compared on a series of model test cases.

Finally, two optimization test cases are implemented from the Finnish design test case database with the above methods. The test cases are BI-NACA0012 position reconstruction and two-dimensional shock control bump optimization installed on a natural laminar flow airfoil (RAE5243). Numerical results illustrate the potential of the above methodology for solving design optimization problems. As a result of numerous experiments with comparisons, the adaptive hybridized mesh/meshless methods coupled with evolutionary algorithms and game strategies provide the designer useful software tools for efficiently solving computational fluid dynamics optimization problems.

Keywords: hybridized mesh/meshless methods, dynamic cloud, adaptive meshless method, evolutionary algorithms, Nash games, hierarchical genetic algorithms, position reconstruction, shape optimization

Author

Hong WANG
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Jacques Périaux
Department of Mathematical Information Technology
University of Jyväskylä
Finland
and
International Center for Numerical Methods in Engineering (CIMNE)
Polytechnic University of Catalonia
Spain

Professor Pekka Neittaanmäki
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Ning Qin
Head of Thermo–Fluids Group
Department of Mechanical Engineering
University of Sheffield
UK

Professor Hiroshi Suito
Department of Environmental and Mathematical Sciences
Okayama University
Japan

Opponents

Docent Antti Hellsten
Lappeenranta University of Technology
Dispersion Modeling group
Air Quality Research
Finnish Meteorological Institute
Finland

Professor Viktor Zakharov
Department of Mathematical Modeling of Energetic Systems
Faculty of Applied Mathematics and Control Processes
Saint Petersburg University
Russia

ACKNOWLEDGEMENTS

Since it is impossible to acknowledge all who have contributed to my thesis, I would like to thank the most influential individual contributors.

First and foremost, I would like to express my greatest gratitude to my supervisors, Professor Jacques Périaux, Professor Pekka Neittaanmäki and Professor Raino Mäkinen. This thesis would not have been possible without their guidance and constant support. I consider myself to be immensely fortunate to have had them as my supervisors. I am amazed, particularly, at the breadth of knowledge and interests that Professor Périaux possesses. His curiosity, enthusiasm and passion for science have been always an inspiration for me. My sincere thanks to the reviewers for providing positive and timely feedback, and especially to Professor Ning Qin for giving me new insights into my work. I thank Doctor Antti Hellsten who kindly agreed to act as my opponent, for diligently reading the manuscript, and for providing very helpful critical comments.

It is with heartfelt gratitude that I thank Professor Hongquan Chen, Professor Zhili Tang and Professor Chunhua Zhou, Professor Ning Zhao from Nanjing University of Aeronautics and Astronautics (China) and Doctor Chris DongSeop Lee from the International Center for Numerical Methods in Engineering at the Universitat Politècnica de Catalunya (Spain) and Doctor Zhihua Ma from Manchester Metropolitan University (UK) for their inspiring and fruitful discussions on my topics. I feel deeply privileged to have worked with and learned from them. Their deep insights and scientific taste have contributed significantly to shaping my thought process and scientific pursuits.

I would also like to thank all my colleagues who worked daily with me in the development of the Design TEKES project to set up a Finnish database defining optimization test cases and storing results of contributors. Among them, I should mention Tero Tuovinen, Tuomo Varis and Jyri Leskinen who shared the office and ideas with me along my research, making my student life enjoyable and truly memorable. I am grateful to the wonderful and efficient staff at the Department of Mathematical Information Technology: Anu Penttilä, Kati Valpe, Marja-Leena Rantalainen and Tiina Lampinen in particular, for helping me navigate through administrative matters with much ease. Furthermore, I appreciate my friends Fengyu Cong, Muaed Kabardov, Olli Mali, Marjaana Nokka, Karthik Sindhya, Tytti Saksa, Litang Tsai, Xiaoyu Xu, Huihan Yang and Xiling Zeng for their invaluable friendship, warm company over the years and for making a home away from home possible.

I wish to acknowledge the generous financial support during my doctoral studies provided by Finnish Doctoral Programme in Computational Sciences (FICS), Jyväskylä Doctoral Program in Computing and Mathematical Sciences (COMAS) and the Department of Mathematical Information Technology.

Above all, I am forever indebted to my parents for their understanding, encouragement and endless support through thick and thin. All my efforts would not have come to fruition without their care and love.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Hong Wang, Hongquan Chen and Jacques Périaux. A Study of Gridless Method with Dynamic Clouds of Points for Solving Unsteady CFD Problems in Aerodynamics. International Journal for Numerical Methods in Fluids. Vol. 64, Issue 1, pages 98–118, 2010.

PII Hong Wang and Jacques Périaux. Genetic Algorithms Using a Gridless Euler Solver for 2D Inverse Problems in Aerodynamic Design. Evolutionary Methods for Design, Optimization and Control. T. Burczynski and J. Périaux (Eds.), pages 268–273, 2011, CIMNE.

PIII Hong Wang, Chris DongSeop Lee, Jacques Périaux and Hongquan Chen. CFD Inverse Problems Solved by Hybrid Mesh/Mesless Methods Using EAs and Nash Games in Aerodynamics. Adaptive Modeling and Simulation 2011. D. Aubry, P. Díez, B. Tie and N. Parés (Eds.), pages 501–510, 2012, CIMNE.

PIV Hong Wang and Jacques Périaux. A Fast Meshless Method Coupled with Artificial Dissipations for Solving 2D Euler Equations. Computers and Fluids. Vol. 71, pages 83–90, 2013.

PV Hong Wang, Hongquan Chen and Jacques Périaux. GAs and Nash GAs Using a Fast Meshless Method for CFD Design. Numerical Methods for Differential Equations, Optimization, and Technological Problems (Series *Computational Methods in Applied Sciences*). R. Repin, T. Tiihonen, and T. Tuovinen (Eds.), pages 93–105, 2013, Springer.

PVI Hong Wang, Jyri Leskinen, Chris DongSeop Lee and Jacques Périaux. Active Flow Control of Airfoil Using Mesh/Meshless Method Coupled to Hierarchical Genetic Algorithms for Drag Reduction Design. *To appear in Computational Methods in Engineering Design and Optimization (special issue of Engineering Computations).*

PVII Jyri Leskinen, Hong Wang and Jacques Périaux. Increasing Parallelism of Evolutionary Algorithms by Nash Games in Design Inverse Flow Problems. *To appear in Computational Methods in Engineering Design and Optimization (special issue of Engineering Computations).*

1 INTRODUCTION AND MOTIVATION

In the field of *Computational Fluid Dynamics* (CFD), iterative methods for solving Euler equations using traditional finite difference methods or finite volume methods were pioneered by S. K. Godunov [16] in the late 1960s and improved by many CFD investigators like B. Van Leer [31], P. L. Roe [62], S. J. Osher [51], A. Jameson [24, 23], T. H. Pulliam [58, 57] among others who introduced successively numerous theoretical and numerical inherent advances. Most of their approaches are computationally structured or unstructured mesh-based. The conventional finite difference method is based on the structured mesh, which is easy to generate but not very flexible in treating complex geometries. In order to be flexible enough to accommodate geometric complexity, unstructured mesh is usually preferred. However, more computational effort has to be paid for generating complicated unstructured meshes.

To completely eliminate the connectivity limitation of the mesh topology, a class of methods, namely meshless (gridless/mesh-free/element-free) methods are nowadays receiving increasing attention. These methods allow more flexibility for computing flows around complex configurations by replacing the mesh topology by clouds of points. Their interesting features motivated many researchers to study them and various meshless approaches have motivated computer scientists and analysts in different application fields. Among these, smooth particle hydrodynamics (SPH) [38, 15], the element free Galerkin (EFG) method [4, 37], the generalized finite difference (GFD) [33] method, the reproducing kernel particle method (RKPM) [35, 34], the finite point method (FPM) [50, 49], and the meshless local Petrov-Galerkin (MLPG) approach [2] are the best known methods. In aerodynamics, many researchers have been working on meshless methods since early 1990s (J. T. Batina [3], K. Morinishi [44, 45], H. Q. Chen [6, 7, 8], D. J. Kirshman [25], H. Luo [39], etc.).

Even though meshless methods do not deal with mesh quality (orthogonality, smoothness, skewness of elements/volumes, etc.), one major drawback of the meshless methods is the low computational efficiency when compared to the present mesh methods. To respond to this challenge, a hybridized mesh/meshless method was introduced by Z. H. Ma et al [40]. In their paper, the spatial deriva-

tives of governing equations are approximated by the weighted least-squares (WLS) method using clouds of points in the vicinity of the body and the finite volume method (FVM) using mesh elements in the rest of the computational domain. More recently, a fast meshless method by H. Wang and J. Périaux in Paper [PIV] has been developed for the solution of two-dimensional(2D) Euler equations with carefully designed *Artificial Dissipation* (AD) in terms of second and fourth order differences.

Most CFD research on meshless methods has been focused on numerical steady simulations. Unsteady flows associated with moving boundaries have been of continuously growing interest to aerodynamics researchers, and Paper [PI] succeeded in extending the WLS meshless methods to rigid moving boundary problems. In this paper, the spatial derivatives are directly approximated by using the WLS meshless method in each cloud of points. An upwind flux difference splitting scheme using Roe's approximate Riemann solver is used for the estimation of the inviscid flux. When using a mesh-based method to simulate these problems, a dynamic mesh technique based on deformation is usually considered in order to adapt the mesh to unsteady motions of rigid bodies. Due to restrictions imposed on the mesh quality and topology, it is very difficult to deform a mesh for large rigid movements of aerodynamic geometries. As mentioned above, meshless methods are not computationally penalized by such constraints and therefore have significant advantages over mesh-based methods in the particular case of unsteady flow computations.

Based on the success in both steady and unsteady 2D Euler simulations using FVM mesh-based methods, WLS meshless methods and hybridized mesh /meshless methods, these techniques have been implemented for solving design optimization problems in the present research. Over the past decades, *Evolutionary Algorithms* (EAs) have become one of the most widely used optimization methods. EAs are based on Darwinian evolution. They were pioneered by J. Holland [19] for adaptation in the 1970s and developed by D. E. Goldberg [17], Z. Michalewicz [41], C. A. Coello Coello [9], K. Deb [12], J. D. Schaffer [63], M. Mitchell [43], among others. The basic ideas of *Genetic Algorithms* (GAs) consists of populations of individuals which evolve in a search space and adapt to the environment by the use of different mechanisms such as crossover, selection and random mutation. Selection of better individuals after a deterministic classification of the population improves the search procedure. The selection is based only on computed values of the single objective function and does not require derivative information. This makes it possible to find optimal solutions in discontinuous design spaces with combined discrete and continuous design variables. Many strategies to extend EAs to multi-objective optimization problems with innovative approaches exist. These are discussed by many researchers like K. Deb [13, 12], K. Miettinen [42], J. Périaux [28, 54] and etc.

In this study, GAs in Paper [PII], Nash GAs in Papers [PIII, PV, PVII], and *Hierarchical Genetic Algorithms* (HGAs) in Paper [PVI] are employed. GAs and Nash GAs are implemented for efficiency comparison to reconstruct the targeted positions for several oscillating airfoils of prescribed surface pressure coefficients.

HGAs, which are studied in Paper [PVI], use three hierarchical topological layers. The top layer has a single population with two child populations in the intermediate layer, which in turn have two child populations on the bottom layer, resulting in a total of seven populations. In the HGAs optimization procedure, each population is handled by a GA and the elitist genetic information obtained in the lower layers migrates to the closest upper layer for setting up better building blocks, while at the same time, individuals below average go down the ladder and die. The HGAs allow the use of multi-fidelity flow analyzers as follows: high fidelity CFD modeling on the top layer, intermediate fidelity CFD modeling on the intermediate layer and low fidelity CFD modeling on the bottom layer. This multi-fidelity advantage makes good use of FVM mesh-based methods, WLS meshless methods and hybridized mesh/meshless methods, playing on different layers with various design optimization requirements.

1.1 Objectives of the research

This study consists of four main goals. First, a meshless method is applied for unsteady CFD problems. A fast dynamic cloud method based on the Delaunay graph mapping strategy is proposed to accurately redistribute clouds of points in the flow field without any iteration. A dual time-stepping approach is further implemented to advance 2D Euler equations via the Arbitrary Lagrangian-Eulerian (ALE) formulation in time.

The second goal of the study is to compare different evolutionary algorithms and games like GAs, Nash GAs and HGAs for optimization problems. Different 2D Euler analyzers are used in this research: a finite volume method, a meshless method, a fast meshless method coupled with new artificial dissipation, a hybridized mesh/meshless method. As for validation purpose, those methods are tested on position reconstruction problems. One of the model test cases implemented in this research is from the Finnish design test case database (TA10, Bi-NACA0012 geometry, available online in the Finnish design test case database [48, 68]).

The third goal of the study is to use different evolutionary algorithms like GAs, Nash GAs and HGAs for a real life shape design. One of the test cases selected in this study is from the Finnish design test case database (TA5, 2D shock control bump optimization, available online in the Finnish design test case database [48, 68]).

The fourth goal of the study is to develop an adaptive meshless method based on pressure gradients in the computational domain. To show the accuracy of the adaptive techniques, a meshless Euler simulation has been tested with adapted clouds of points with equal number of points for an inviscid supersonic flow in a channel. Moreover, one shape reconstruction test case with circular arc bump using adaptive clouds of points is achieved.

1.2 Organization of the thesis

The content of the thesis is organized in the following manner. In Chapter 2, mesh/meshless methods for solving 2D Euler equations are introduced. In Chapter 3, a review of evolutionary optimization methods used in the study is presented and the state-of-the-art of advanced EAs and game strategies is introduced. Chapter 4 describes two cloud moving techniques: the Delaunay graph mapping strategy based dynamic cloud technique with its application to rigid moving boundary problems and an adaptive meshless method with its applications to shape reconstruction for an inviscid supersonic flow in a channel. Chapter 5 gives two examples of how the mesh/meshless methods can be used for design problems and applied for two test cases defined in the Finnish design test case database. One demonstrates position reconstruction problems for oscillating Bi-NACA0012 in Section 5.1 using Nash games. The other one in Section 5.2 is a 2D real life application which is to reduce the drag of a shock control bump installed on an RAE5243 airfoil by optimizing its shape and location on the upper side of the airfoil using the HGAs methodology. Finally, Chapter 6 presents the summary of this thesis and concludes with future research directions.

1.3 Author's contributions in included papers

The core of this research deals with the hybridized mesh/meshless methods for solving evolutionary design optimization problems coupled to Nash games.

In Paper I, a dynamic cloud technique based on the Delaunay graph mapping strategy is introduced and developed by the author to simulate rigid moving boundary problems within the framework of meshless methods. The innovative method presented in this paper is a CFD application of meshless methods to unsteady flow simulations. Spatial derivatives are directly approximated by using the WLS meshless method in each cloud of points. An upwind flux difference splitting scheme using Roe's approximate Riemann solver is used for the estimation of the inviscid flux. A dual time-stepping approach using an explicit four-stage Runge–Kutta scheme is used to advance the flow equations in time. Computational results obtained for transonic and supersonic flows around a single NACA0012 airfoil and an RAE2822 airfoil in the steady case prove the accuracy of spatial discretization achieved by the present meshless method. Then the method is applied to unsteady flows around NACA0012 airfoil and NACA64A010 airfoil oscillation in pitch situations about their quarter chord, and the computational results obtained are in good agreement with the experimental data. Those tests demonstrate both the validity and practicality of the integration of the present method for solving optimization problems. The paper was mainly written by the author, who also conducted all the computations.

In Paper II, GAs have been used to reconstruct the targeted position for sev-

eral oscillating airfoils of prescribed surface pressure coefficients. A fast dynamic cloud technique based on the Delaunay graph mapping strategy is employed for simulations of different targeted positions. Spatial derivatives are directly approximated by using the WLS meshless method in each cloud of points. An upwind flux difference splitting scheme using Roe's approximate Riemann solver is used for the estimation of the inviscid flux, and time integration is further implemented by an explicit four-stage Runge–Kutta scheme. Position reconstructions of three oscillating airfoils operating in transonic regimes have been solved successfully with a GAs optimizer and the meshless Euler solver with dynamic cloud technique based on the Delaunay graph mapping strategy. Comparisons of targeted and computed parameters are presented to prove the ability of our method in position reconstruction in aerodynamic design. The paper was mainly written by the author, who also conducted all the computations.

Paper III investigates the numerical efficiency of mesh, meshless and hybridized mesh/meshless methods for solving inverse reconstruction problems dealing with multi-element geometries such as Bi-NACA0012 configuration operating at subsonic regimes. Spatial derivatives are directly approximated by using the WLS meshless method in each cloud of points. An upwind flux difference splitting scheme using Roe's approximate Riemann solver is used for the estimation of the inviscid flux, and time integration is further implemented by an explicit four-stage Runge–Kutta scheme accelerated by a local time-stepping approach and an implicit residual averaging method. The Delaunay graph mapping strategy is considered to move clouds of points in the meshless sub-domain close to the multi-element configuration avoiding the regeneration in the region of the mesh sub-domain during the optimization procedure. Results obtained by the hybridized mesh/meshless methods have the advantages of flexibility, accuracy and efficiency of both mesh method and meshless method. Numerical experiments clearly demonstrate the benefits of using a hybridized mesh/meshless Euler flow analyzer feeding an evolutionary optimizer. Numerical results have shown the potential and the flexibility of the hybridized mesh/meshless technique for inverse problems in aerodynamics. The paper was mainly written by the author, who also conducted all the computations.

In Paper IV, a new fast meshless method has been developed for the solution to 2D Euler equations, with new artificial dissipation parameters introduced in the discretization of second and fourth order derivatives. Additionally, an explicit five-stage Runge–Kutta scheme is utilized to reach the steady state solution. A local time-stepping approach and a residual averaging method are implemented to accelerate the density residual convergence. This paper compares the computational efficiency of the proposed meshless method, one finite volume method, and one meshless method in literature for compliated flows over one or two staggered airfoils. Numerical results show that the proposed meshless method has the same accuracy and a much lower computational cost when compared to the other two methods. Our numerical study demonstrates that the proposed method is efficient and accurate for aerodynamics design problems. The paper was mainly written by the author, who also conducted all the compu-

tations.

Paper V is focused on the efficiency of Nash GAs optimizers. A fast meshless method using second and fourth order artificial dissipation and a dynamic cloud technique based on the Delaunay graph mapping strategy is introduced to solve inverse CFD problems. The purpose of this paper is to use GAs and Nash GAs for position reconstructions of oscillating airfoils. The main feature of this paper is a detailed investigation on inverse problems in aerodynamics using both flexibility and efficiency of the fast meshless method. Comparisons of prescribed and computed aerodynamics parameters are presented for position reconstruction problems in aerodynamic design using both a fast meshless method coupled with artificial dissipation and a finite volume method. Numerical results are presented to illustrate the potential of the fast meshless method coupled with artificial dissipation and evolutionary algorithms, to solve more complex optimization problems. The author conducted the computations based on GAs and Nash GAs coded in Paper [III] and the fast meshless solver in Paper [IV]. The paper was mainly written by the author.

In Paper VI, the authors explore the use of HGAs in case of multi-fidelity models in three different layers for speeding up optimization process. The idea was introduced by M. Sefrioui and J. Périaux in the 1990s on supersonic flows in nozzles [64]. The top layer consists of a single sub-population operating on a precise model. On the middle layer, two sub-populations operate on a model of intermediate accuracy. The bottom layer, consisting of four sub-populations (two for each middle layer populations), operates on a coarse model. The method was validated on a real life optimization problem consisting of a two-dimensional shock control bump optimization installed on a natural laminar flow airfoil (RAE5243) from the Finnish design test case database. The precise model uses the fast meshless method coupled with artificial dissipation based on fine clouds of points distributed in the computational domain. The intermediate model uses a hybridized mesh/meshless method, and the coarse model uses a fast but less accurate finite volume method. The hierarchical approach brought considerable improvements in both CPU cost and in the quality of the final results compared to a single-population based GA. The author conducted all the computations, with the help of J. Leskinen and D. S. Lee, for the basic understanding of HGAs. The paper was mainly written by the author.

Paper VII is a study conducted for comparing different levels of parallelization of EAs. The selected algorithms are jDifferential evolution (jDE), jDE using the island model, Nash-jDE using geometry decomposition techniques, and *Global Nash Game Coalitions Algorithm* (GNGCA) described by J. Leskinen and J. Périaux in [32]. The methods were tested on the BI-NACA0012 geometry reconstruction, an optimization test case defined in the Finnish design test case database [48, 68]). The results show that GNGCA software is the most efficient or comparable to the Nash approach using two masters and two slaves. This occurred even though the threads responsible for the flow reconstruction were most of the time idle. For simulating an optimal load balancing, GNGCA was also run with two slave threads leading into further improvement in efficiency.

The island model did not bring any algorithmic improvement, probably because of the relative simplicity of the objective function landscape. The test case on the Bi-NACA0012 geometry reconstruction was brought by the author and has contributed to the comparisons of different results using Nash game strategy. The paper was jointly written by J. Leskinen and the author.

2 MESH/MESHLESS METHODS FOR SOLVING 2D EULER EQUATIONS

Meshless methods are different from traditional mesh methods, requiring only clouds of points distributed in the computational domain. In this chapter, a new fast meshless method is described and implemented for the solution of 2D Euler equations, with new artificial dissipation parameters introduced in the discretization of second and fourth order derivatives. Additionally, an explicit five-stage Runge–Kutta scheme is utilized to reach a steady state solution. A local time-stepping approach and a residual averaging method are implemented to accelerate the density residual convergence. A numerical example for two staggered NACA0012 airfoils using the above fast meshless method coupled with *Artificial Dissipation* (AD) is shown in Section 2.5.

2.1 Governing equations

The Euler equations represent the conservation principle for mass, momentum and energy of inviscid fluids. In a 2D Cartesian coordinate system, the Euler equations are expressed in the following form

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = 0 \tag{1}$$

where t is the time and (x, y) are the Cartesian coordinates. The vectors of conservative variables $\mathbf{W}$, inviscid fluxes $\mathbf{E}$ and $\mathbf{F}$ have the following components,

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{bmatrix} \tag{2}$$

where ρ is the fluid density, u is the x-velocity component, v is the y-velocity component, p is the pressure, and e is the total energy per unit volume. For an

ideal gas, e can be written as

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right) \tag{3}$$

where γ is the ratio of specific heat. Additionally, the equation of state is given by

$$p = \rho \bar{R} T \tag{4}$$

where T is the static temperature and $\bar{R}$ the ideal gas constant.

In the process of numerical calculation, we often use the non-dimensional parameters for simplicity. A different non-dimensional system may alter the equations appearance, but the underlying physical nature is not changed. The non-dimension system is described as follows

$$(x^\star, y^\star, t^\star) = \left(\frac{x}{l}, \frac{y}{l}, \frac{t}{l} C_0\right) \tag{5a}$$

$$(p^\star, \rho^\star, E^\star) = \left(\frac{p}{p_\infty}, \frac{\rho}{\rho_\infty}, \frac{E}{C_0^2}\right) \tag{5b}$$

$$(u^\star, v^\star) = \left(\frac{u}{C_0}, \frac{v}{C_0}\right) \tag{5c}$$

where $C_0 = \sqrt{p_\infty / \rho_\infty}$. Parameters with asterisk superscript denote the non-dimensional parameters, parameters with subscript ∞ mean the free stream and l denotes the reference length. Let α be the angle of attack, and the non-dimensional free stream physical quantities are

$$(p_\infty^\star, \rho_\infty^\star, E_\infty^\star) = \left(1, 1, \frac{1}{\gamma - 1} + \frac{1}{2} M_\infty^2 \gamma\right) \tag{6a}$$

$$(u_\infty^\star, v_\infty^\star) = (\sqrt{\gamma} M_\infty \cos\alpha, \sqrt{\gamma} M_\infty \sin\alpha) \tag{6b}$$

The Euler equations derived by (6) with non-dimensional parameters and dimensional parameters are in the same form. For convenience in writing, we do not use the asterisk superscript for non-dimensional parameters.

2.2 Spatial discretization

Meshless method is relatively new for the compressible flow applications compared to the traditional mesh methods including the finite difference, finite element and finite volume methods. In this section, basic theory of the WLS meshless methods is first described, then the spatial discretization procedure of Euler equations is presented.

2.2.1 Basic theory of the WLS meshless methods

The essential idea of meshless methods is to introduce a number of scattered points $P_i (i = 1, 2, 3, \ldots, N)$ to a domain Ω. It is not necessary to consider the

connectivity between the points. Figure 1 gives an example of the domain discretization using meshless points. For each point, several points around it are chosen to form a cloud of points [3, 49, 6, 7, 40]. Figure 2 shows a cloud of points C_i, in which the point i is named the center and the other points are called the satellites (P_{ij} is the midpoint between i and j).

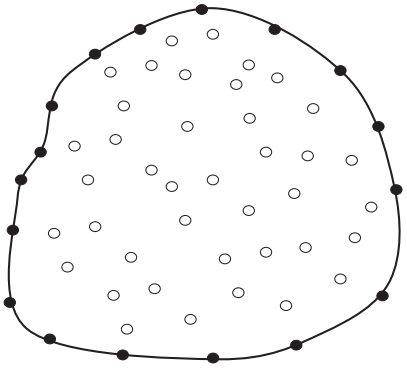

FIGURE 1 Meshless points distribution in domain Ω.

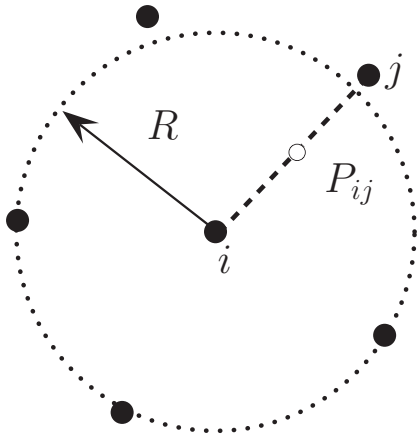

FIGURE 2 A typical meshless cloud of points.

The coordinate difference between the satellite j and the center i can be expressed as

$$h_j^i = x_j - x_i, \qquad l_j^i = y_j - y_i \tag{7}$$

For simplicity, we use h_j and l_j to denote h_j^i and l_j^i, respectively. The vector

$$\vec{r}_j^{\,i} = (h_j, l_j)$$

starts from i to j, its length is

$$r_j^i = \sqrt{h_j^2 + l_j^2} \tag{8}$$

and we define the reference radius of the cloud as

$$R_i = \min\left(r_1^i, r_2^i, \ldots, r_j^i, \ldots, r_{M_i}^i\right) \tag{9}$$

where M_i is the total number of the satellites in the cloud.

The least-squares method is an approach adopted within the framework of meshless methods in the present study to approximate the spatial derivatives of a function and its basic idea is described as follows [6, 7, 40]. Considering any sufficiently differentiable function $f(x, y)$ in a given small domain Ω_i, the Taylor series about a point $P_i(x_i, y_i)$ can be expressed in the following form

$$f = f_i + a_1 h + a_2 l + \frac{1}{2}a_3 h^2 + \frac{1}{2}a_4 l^2 + a_5 hl + O(h^3, l^3) \tag{10}$$

where $f = f(x, y)$ and $f_i = f(x_i, y_i)$ are the function values, $h = x - x_i$ and $l = y - y_i$ are the coordinate differences between the points. The coefficients $a_k (k = 1, 2, 3, 4, 5)$ represent the partial derivatives of the function at $P_i(x_i, y_i)$

$$a_1 = \left.\frac{\partial f}{\partial x}\right|_i,\ a_2 = \left.\frac{\partial f}{\partial y}\right|_i,\ a_3 = \left.\frac{\partial^2 f}{\partial x^2}\right|_i,\ a_4 = \left.\frac{\partial^2 f}{\partial y^2}\right|_i,\ a_5 = \left.\frac{\partial^2 f}{\partial x \partial y}\right|_i \tag{11}$$

Keeping the terms until second order, we obtain the approximate function value at point $P_j(x_j, y_j)$

$$\tilde{f}_j = f_i + a_1 h_j + a_2 l_j + a_3 \frac{h_j^2}{2} + a_4 \frac{l_j^2}{2} + a_5 h_j l_j \tag{12}$$

As the partial derivatives in Euler equations are of first order, the terms in the formula (10) being kept to first order is usually reasonable. Consequently, the approximate value is

$$\tilde{f}_j = f_i + a_1 h_j + a_2 l_j \tag{13}$$

and the error between the exact and approximate values is

$$e_j = f_j - \tilde{f}_j = f_j - \left(f_i + a_1 h_j + a_2 l_j\right) \tag{14}$$

A weight function is introduced here to construct e_j considering the effect of different distance between the satellite j and the center i based on Eq. (8) and Eq. (9) as

$$\omega_j^i = \frac{r_j^i}{R_i} \tag{15}$$

Then for cloud C_i, the total error can be estimated by the following expression

$$\Phi = \frac{1}{2}\sum_{j=1}^{M_i}\left(\omega_j^i e_j\right)^2 \tag{16}$$

In order to minimize the error Φ, its derivatives about a_1 and a_2 are set to zero

$$\frac{\partial \Phi}{\partial a_1} = \frac{\partial \Phi}{\partial a_2} = 0 \tag{17}$$

and we gain a set of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{18}$$

where

$$\mathbf{A} = \begin{bmatrix} \sum \omega_j^{i2} h_j^2 & \sum \omega_j^{i2} h_j l_j \\ \sum \omega_j^{i2} h_j l_j & \sum \omega_j^{i2} l_j^2 \end{bmatrix} \tag{19}$$

$$\mathbf{x} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \sum \omega_j^{i2} h_j (f_j - f_i) \\ \sum \omega_j^{i2} l_j (f_j - f_i) \end{bmatrix} \tag{20}$$

If the matrix $\mathbf{A}$ is not singular, this system of equations can be solved by the following simple strategy

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{21}$$

The solutions can be written into linear combinations of the function values at different points

$$a_1 = \left.\frac{\partial f}{\partial x}\right|_i = \sum_{j=1}^{M_i} \alpha_j (f_j - f_i), \;\; a_2 = \left.\frac{\partial f}{\partial y}\right|_i = \sum_{j=1}^{M_i} \beta_j (f_j - f_i) \tag{22}$$

where α_j and β_j are computed from (21). By removing the dependence of the point i itself in the derivative approximation to satisfy,

$$\left.\frac{\partial f}{\partial x}\right|_i = \sum_{j=1}^{M_i} \widehat{\alpha}_j f_j, \;\; \left.\frac{\partial f}{\partial y}\right|_i = \sum_{j=1}^{M_i} \widehat{\beta}_j f_j \tag{23}$$

where $\widehat{\alpha}_j = \alpha_j + \widehat{\omega}_j \alpha_i$ and $\widehat{\beta}_j = \beta_j + \widehat{\omega}_j \beta_i$.
They can also be estimated using the following formulae

$$a_1 = \sum_{j=1}^{M_i} \alpha_{ij} (f_{ij} - f_i) = \sum_{j=1}^{M_i} \widehat{\alpha}_{ij} f_{ij} \tag{24a}$$

$$a_2 = \sum_{j=1}^{M_i} \beta_{ij} (f_{ij} - f_i) = \sum_{j=1}^{M_i} \widehat{\beta}_{ij} f_{ij} \tag{24b}$$

where $\widehat{\alpha}_{ij} = \alpha_{ij} + \widehat{\omega}_{ij} \alpha_i$ and $\widehat{\beta}_{ij} = \beta_{ij} + \widehat{\omega}_{ij} \beta_i$, the subscript ij denotes the midpoint P_{ij} between i and j, and f_{ij} is estimated at the midpoint [6, 7, 8, 45].

2.2.2 Discretization of Euler equations

The formulae from (18) to (24) provide a way to compute the spatial derivatives within a cloud of points using the WLS meshless method. The Euler equations are required to be satisfied for every cloud of points in the domain. For any cloud C_i, (1) can be written as

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_i + \left(\frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right)_i = 0 \tag{25}$$

For the inviscid fluxes, let

$$\mathbf{Q}_i = \left(\frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right)_i \tag{26}$$

Substituting (24) into (26)

$$\mathbf{Q}_i = \sum_{j=1}^{M_i} \widehat{\alpha}_{ij}\mathbf{E}_{ij} + \sum_{j=1}^{M_i} \widehat{\beta}_{ij}\mathbf{F}_{ij} \tag{27}$$

In order to prevent odd-even decoupling of the solution and oscillations near shock waves, additional artificial dissipation terms $\mathbf{D}_i$ are required for the present work. The governing equation can be written as follows

$$\frac{\partial \mathbf{W}_i}{\partial t} = -\left(\mathbf{Q}_i - \mathbf{D}_i\right) \tag{28}$$

The artificial dissipation $\mathbf{D}_i$ is a blend of second and fourth-order differences

$$\mathbf{D}_i = \sum_{j=1}^{M_i} \left[\varepsilon_{ij}^{(2)}\left(\mathbf{W}_j - \mathbf{W}_i\right) - \varepsilon_{ij}^{(4)}\left(\nabla^2\mathbf{W}_j - \nabla^2\mathbf{W}_i\right)\right] \tag{29}$$

where

$$\nabla^2\mathbf{W}_i = \sum_{j=1}^{M_i} \mathbf{W}_j - M_i\mathbf{W}_i \tag{30}$$

A pressure-based sensor is used to switch off the fourth-order differences at shocks and the second-order differences in smooth portions of the flow field. In (29), the coefficients $\varepsilon_{ij}^{(2)}$ and $\varepsilon_{ij}^{(4)}$ are defined as

$$\varepsilon_{ij}^{(2)} = K^{(2)}\lambda_{ij}\max\left(\mu_i, \mu_j\right) \tag{31a}$$

$$\varepsilon_{ij}^{(4)} = \lambda_{ij}\max\left[0, K^{(4)} - \varepsilon_{ij}^{(2)}\right] \tag{31b}$$

with the pressure-based sensor μ and λ_{ij} given by

$$\mu_i = \frac{\left|\nabla^2 p_i\right|}{\sum_{j=1}^{M_i}\left(p_i + p_j\right)} \tag{32a}$$

$$\lambda_{ij} = \left|\widehat{\alpha}_{ij}u + \widehat{\beta}_{ij}v\right| + c\sqrt{\widehat{\alpha}_{ij}^2 + \widehat{\beta}_{ij}^2} \tag{32b}$$

where u, v and c are the local velocities and speed of sound, and $K^{(2)}$ and $K^{(4)}$ are parameters taken as $1/2$ and $1/64$, respectively.

2.3 Temporal discretization

In the present work, Euler equations are treated by the method-of-line, which separates the spatial and temporal discretization. The semi-discrete form of the governing equations in cloud C_i is written as follows

$$\left.\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t}\right|_i = \mathbf{R}_i \tag{33}$$

where $\mathbf{R}_i$ is the residual value. An explicit scheme is used for the time discretization on (33) yielding

$$\frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} = \mathbf{R}_i \tag{34}$$

The superscripts n and $(n+1)$ denote the time levels. Here $\mathbf{W}^n$ refers to the flow solution at the present time t, and $\mathbf{W}^{n+1}$ represents the solution at the time $(t + \Delta t)$. An explicit five-stage Runge–Kutta time integration scheme is used,

$$\begin{cases} \mathbf{W}_i^{(0)} = \mathbf{W}_i^n \\ \mathbf{W}_i^{(1)} = \mathbf{W}_i^{(0)} + \alpha_1 \Delta t_i \mathbf{R}_i^{(0)} \\ \mathbf{W}_i^{(2)} = \mathbf{W}_i^{(1)} + \alpha_2 \Delta t_i \mathbf{R}_i^{(1)} \\ \mathbf{W}_i^{(3)} = \mathbf{W}_i^{(2)} + \alpha_3 \Delta t_i \mathbf{R}_i^{(2)} \\ \mathbf{W}_i^{(4)} = \mathbf{W}_i^{(3)} + \alpha_4 \Delta t_i \mathbf{R}_i^{(3)} \\ \mathbf{W}_i^{(5)} = \mathbf{W}_i^{(4)} + \alpha_5 \Delta t_i \mathbf{R}_i^{(4)} \\ \mathbf{W}_i^{n+1} = \mathbf{W}_i^{(5)} \end{cases} \tag{35}$$

where α_k, $k = (1, 2, 3, 4, 5)$ represents the stage coefficients,

$$\alpha_1 = \frac{1}{4}, \quad \alpha_2 = \frac{1}{6}, \quad \alpha_3 = \frac{3}{8}, \quad \alpha_4 = \frac{1}{2}, \quad \alpha_5 = 1 \tag{36}$$

2.4 Acceleration techniques

The main disadvantage of the explicit Runge–Kutta time integration scheme in the previous section is that the maximum permissible time step is limited by the Courant–Friedrichs–Lewy (CFL) stability condition [5]. To relax this restriction and accelerate the convergence, a local time-stepping approach and an implicit residual averaging method are employed in this study.

The local time step Δt_i of discrete point is given by the following equation [5]

$$\Delta t_i = \frac{\sigma}{\sum_{j=1}^{M_i} \left| \widehat{\alpha}_{ij} u + \widehat{\beta}_{ij} v \right| + c\sqrt{\widehat{\alpha}_{ij}^2 + \widehat{\beta}_{ij}^2}} \tag{37}$$

where σ denotes the CFL number.

In the time marching equation, let $\mathbf{R}_i$ represent the residual at point i. In the meshless method, a new residual can be given by [5]

$$\mathbf{R}'_i = \frac{\mathbf{R}_i + \varepsilon \sum_{j=1}^{M_i} \mathbf{R}'_j}{1 + \varepsilon M_i} \tag{38}$$

where $\varepsilon = [0.2, 0.5]$. It can be accomplished by performing two Jacobi iterations. The above techniques increase the CFL number up to 5 in the present work.

2.5 Numerical Results

For validating the fast meshless method coupled with AD, one test case of two staggered NACA0012 airfoils operating at flow conditions as Mach number 0.8 and angle of attack 1.25°, is simulated using the fast meshless method coupled with AD, the finite volume method (FVM) described in [24] and the meshless method in [40]. The following results can be found in Paper [IV]. It is noted that all numerical simulations in Paper [IV] were performed on a computer named *TELECASTER* and equipped with Intel(R) Core(TM) 2 Quad CPU Q9650 @ 2.00GHz 2.99 GHz with 3.00 GB of RAM.

Figure 3 provides both a global view and a close-up view of clouds of points distributed on two staggered NACA0012 airfoils. Figure 4 shows both the global view and the close-up of the mesh distributed on the same airfoils. For the meshless method, a total of 9064 clouds of points were used in the global domain, whereas 17408 mesh elements were used for the mesh method. Figure 5 shows the comparison of surface pressure coefficients for this test case using the fast meshless method coupled with AD, the FVM and the meshless method in [40].

TABLE 1 Comparison of convergence history on two staggered NACA0012 airfoils in terms of the CPU time cost.

Methods	CPU (mins)
Meshless method - AD	1.24
Mesh method - Jameson 1981	3.10
Meshless method - Ma 2006	14.93

Figure 6 shows the comparison of convergence history for the number of iterations using the fast meshless method coupled with AD, the FVM and the meshless method in [40]. The fast meshless method coupled with AD in this test case saves up to 92% of the iteration cost while the FVM saves 78% compared to the meshless method in [40]. The comparison of convergence history on two staggered NACA0012 airfoils in terms of the CPU time cost is listed in Table 1: the fast meshless method coupled with AD takes 1.24 minutes; the FVM takes 3.10 minutes, which is more than 2 times slower compared to the fast meshless

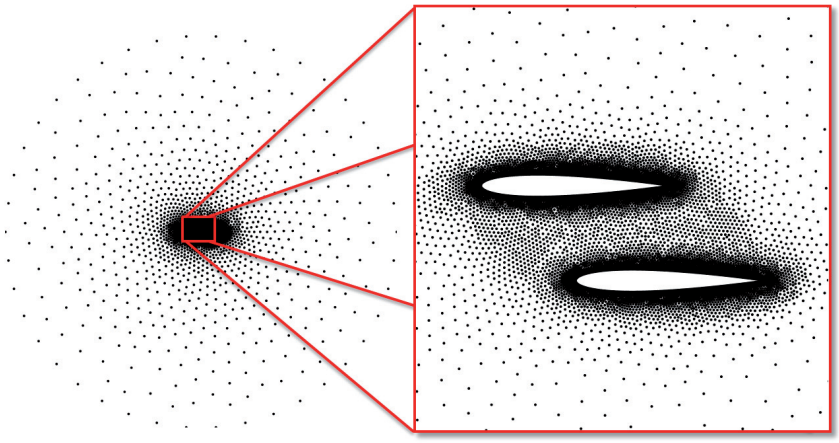

FIGURE 3 Global and close-up views of clouds of points for two staggered NACA0012 airfoils.

method coupled with AD; and the meshless method in [40] takes 14.93 minutes, which is more than 10 times slower compared to the fast meshless method coupled with AD.

Other numerical test cases presented in Paper [PIV] show that, by using the proposed fast meshless method coupled with AD, a rapid convergence to a steady state can be obtained with the fully explicit five-stage Runge–Kutta time integration scheme. This new fast meshless Euler solver presented in this chapter is used intensively to solve optimization CFD problems in Papers [PV] and [PVI].

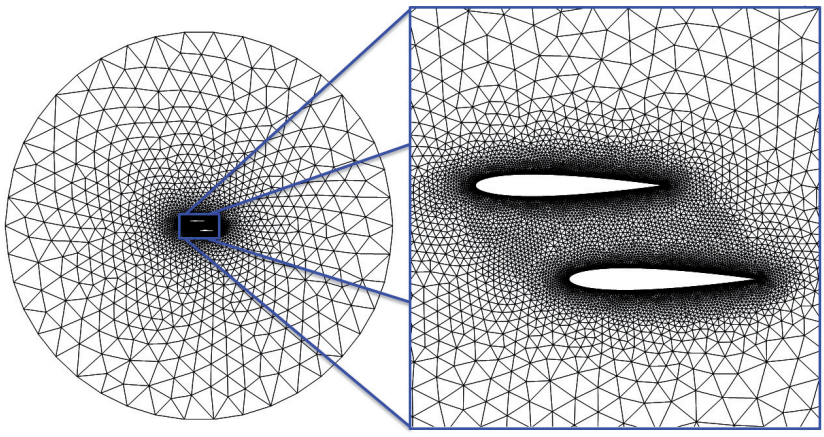

FIGURE 4 Global and close-up views of the unstructured mesh for two staggered NACA0012 airfoils.

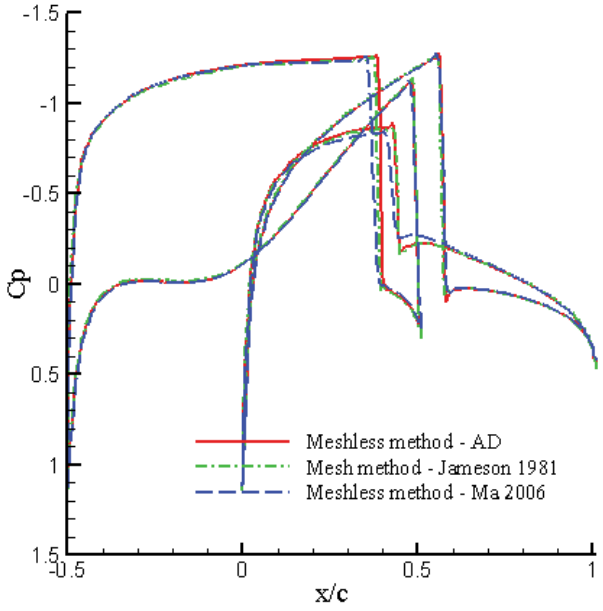

FIGURE 5 Comparison of surface pressure coefficients on two staggered NACA0012 airfoils.

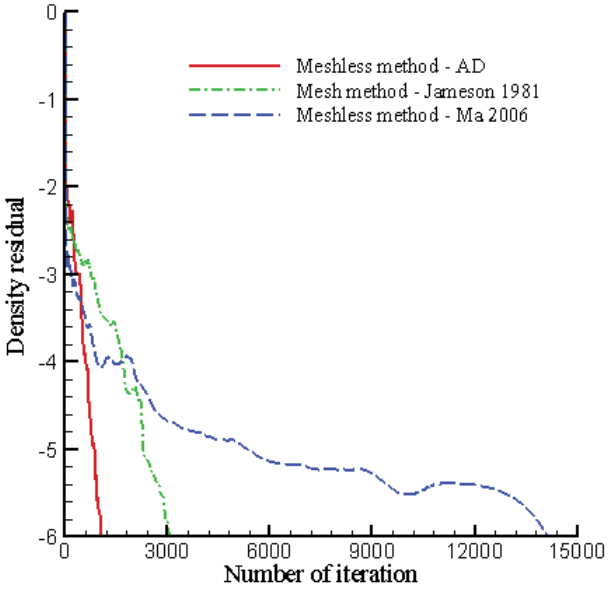

FIGURE 6 Comparison of convergence history on two staggered NACA0012 airfoils in terms of the number of iterations.

3 GENETIC ALGORITHMS AND GAME STRATEGIES FOR DESIGN OPTIMIZATION

In this chapter, *Genetic Algorithms* (GAs) and game strategies are introduced and discussed as optimizers for CFD design optimization problems. *Evolutionary Algorithms* (EAs) for design optimization have been developed since the late 1960s and have been under intensive research due to their ability to capture global optima. Compared to most of the deterministic methods, EAs do not require gradient information, can work with non-differentiable objective functions and are noise tolerant. Disadvantages of EAs are mainly related to the large number of function evaluations needed. Many approaches have been proposed in order to improve the accuracy and high computational cost of the methods. In the following sections, the methods used in this study are reviewed and their computational implementation detailed.

3.1 Evolutionary algorithms

Many global optimization methods have been developed during past decades. The most popular evolution-based algorithms are undoubtedly EAs, merging into one name with several different schools, including GAs, Evolution Strategies (ESs) [61], Differential Evolution (DE) [66] and Genetic Programming (GP) [26]. EAs are population-based metaheuristic optimization algorithms. EAs work according to the so-called "survival of the fittest" Darwinian principle inspired by biological evolution. The main mechanisms of EAs include reproduction, mutation, recombination, and selection. Candidate solutions to an optimization problem are individuals of a population, with a minimized or maximized fitness function evolving in a search space – environment. Evolution of the population takes place over several generations. The most common form of EAs is GAs, which were first proposed by J. Holland [19]. GAs are traditionally represented on binary-coded chromosomes or strings. These chromosomes are modified by operators (crossover, mutation, selection) in order to search better solutions. Bet-

ter individuals with the selection – crossover – mutation procedure are created until a stopping criterion ends the optimization search.

In order to show how GAs operate in the computer, let us consider the following single-objective test function:

$$f(\mathbf{x}) = \sum_{i=1}^{N_{dim}} x_i^2, \quad x_i \in [-5.0, 5.0] \tag{39}$$

which has only one global minimum, $x_i = 0$, $i = 1, \ldots, N_{dim}$. N_{dim} is the total number of variables. In the following case, N_{dim} is chosen as 2.

The validity of GAs is tested using (39) with two variables. Parameter values in the GA are chosen as follows: 100 as the size of population, 0.9 as the probability of crossover and 0.01 as the probability of mutation.

The elite values are shown in Figure 7. The algorithm is able to converge into $f = 0.36 \times 10^{-08}$ successfully in only 32 generations.

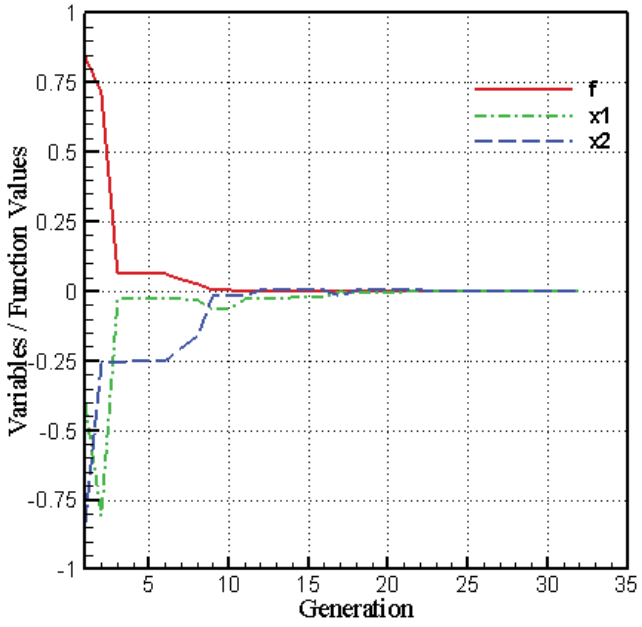

FIGURE 7 Convergence history of GAs $f = 0.0$, $(x_1, x_2) = (0.0, 0.0)$.

3.2 Multi-objective EAs and Game theory

Game theory was founded and established by mathematician J. von Neumann [69]. Most of his research on games was based on the zero-sum of the players, which is a mathematical representation of a situation in which a player's gain (or loss)

of utility is exactly balanced by the losses (or gains) of the utility of the other player(s). In the early 1950s, J. Nash [47] developed another game (*Nash equilibrium*) with the concept of "non-zero-sum", in which the sum of gains and losses by players are sometimes more or less than what they began with. Since then, game models have been broadened theoretically and applied to economics, politics, biology and other social and behavioral sciences. A frequent domain of applications of different games can be found in multi-objective design optimization.

In CFD design problems in particular, a multi-objective optimization problem $(N \geq 2)$ can be formulated mathematically as follows:

$$\begin{array}{rcl} \min\limits_{\mathbf{x}\in S} F(\mathbf{x}) & = & (f_1(\mathbf{x}), \ldots, f_N(\mathbf{x})) \\ \text{subject to } g_j(\mathbf{x}) & = & 0 \\ h_k(\mathbf{x}) & \leq & 0 \end{array} \tag{40}$$

where $j = 1, \ldots, M$, $k = 1, \ldots, P$, f_i are objective functions, g_j and h_k are equality and inequality constraints.

Solving a multi-objective problem by combination of weighted objectives has several drawbacks, such as a loss of information and a priori definition a priori of the weights associated to each objective. J. D. Schaffer [63] in 1985 was the first to propose the GAs approach for multiple objectives through a *Vector Evaluated Genetic Algorithm* (VEGA). Then D. E. Goldberg [17] proposed another solution with the Pareto ranking and sharing in order to distribute the solutions over the entire Pareto front (ensuring diversity) in 1989.

In the following, Section 3.2.1 presents a Pareto based multi-objective algorithm inspired by *Non Dominated Sorting genetic Algorithm* (NSGA) introduced by K. Deb [65, 13, 12]. Section 3.2.2 is focused on a non-cooperative multi-objective algorithm which is based on Nash equilibrium. In Section 3.2.3, the coupling of Nash game with GAs is described on two mathematical functions and hybridized Pareto-Nash Games are introduced in Section 3.2.4

3.2.1 Pareto Optimality

Pareto Optimality (named after V. Pareto [52]) is a concept in economics with applications to engineering. In a Pareto based efficient economic allocation, no one can be made better off without making at least one individual worse off. This cooperative approach was proposed by D. E. Goldberg [17] to distribute multiple objective solutions over the entire Pareto front. Then cooperative Pareto game coupled with EAs gave rise to standard algorithms for multi-objective optimization ([20, 65, 13]).

Figure 8 shows the Pareto optimality concept for a two-objective problem with cooperation of players. A candidate solution is Pareto optimal if there is no other outcome that makes every player at least as well off and at least one player strictly better off.

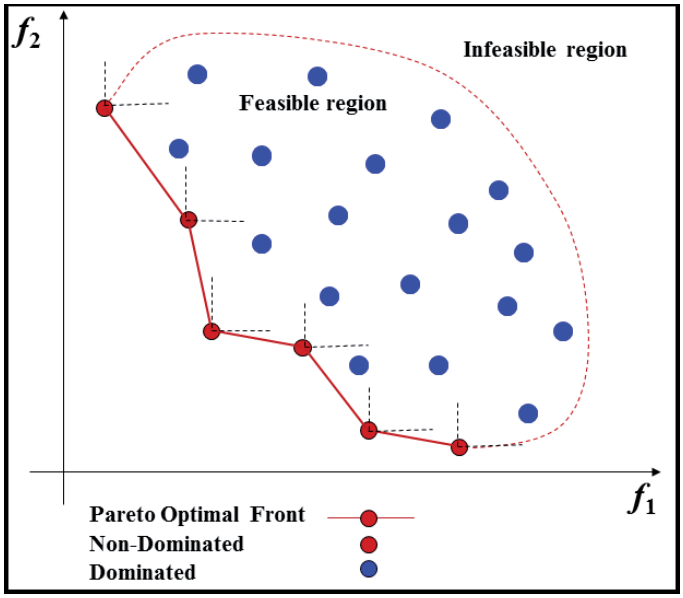

FIGURE 8 Pareto optimality.

3.2.2 Nash equilibrium

In the early 1950s, J. Nash [47] proved mathematically that finite games always have an equilibrium point at which all players choose actions which are best for them given their opponents' choices. This key concept of non-cooperative game theory has been a focal point of analysis since then.

The Nash equilibrium definition is a fundamental concept in the theory of games involving two or more players in which each player is assumed to know the equilibrium strategies of other players and no player has anything to gain by changing his(her) own strategy unilaterally. For instance, consider two players, *1* and *2*, with their own set of variables, x and y. Let S_1 and S_2 are the set of rational strategies for each player. A strategy pair $(x^\star, y^\star)$ is defined as Nash equilibrium in $S_1 \times S_2$ if and only if:

$$\begin{cases} f_1(x^\star, y^\star) &= \inf\limits_{x \in S_1} f_1(x, y^\star) \\ f_2(x^\star, y^\star) &= \inf\limits_{x \in S_2} f_2(x^\star, y) \end{cases} \tag{41}$$

Figure 9 is a flow chart of Nash GAs. Let S_1 and S_2 be the string representing the potential solution for a two-objective optimization, where x responds to the first criterion and y responds to the second one. Player 1 optimizes x while $y^\star$ is fixed by Player 2; Player 2 optimizes y while $x^\star$ is fixed by Player 1 in parallel. Each player has his(her) own GA with population. Nash equilibrium is reached when neither player can do better by unilaterally changing his(her) criterion.

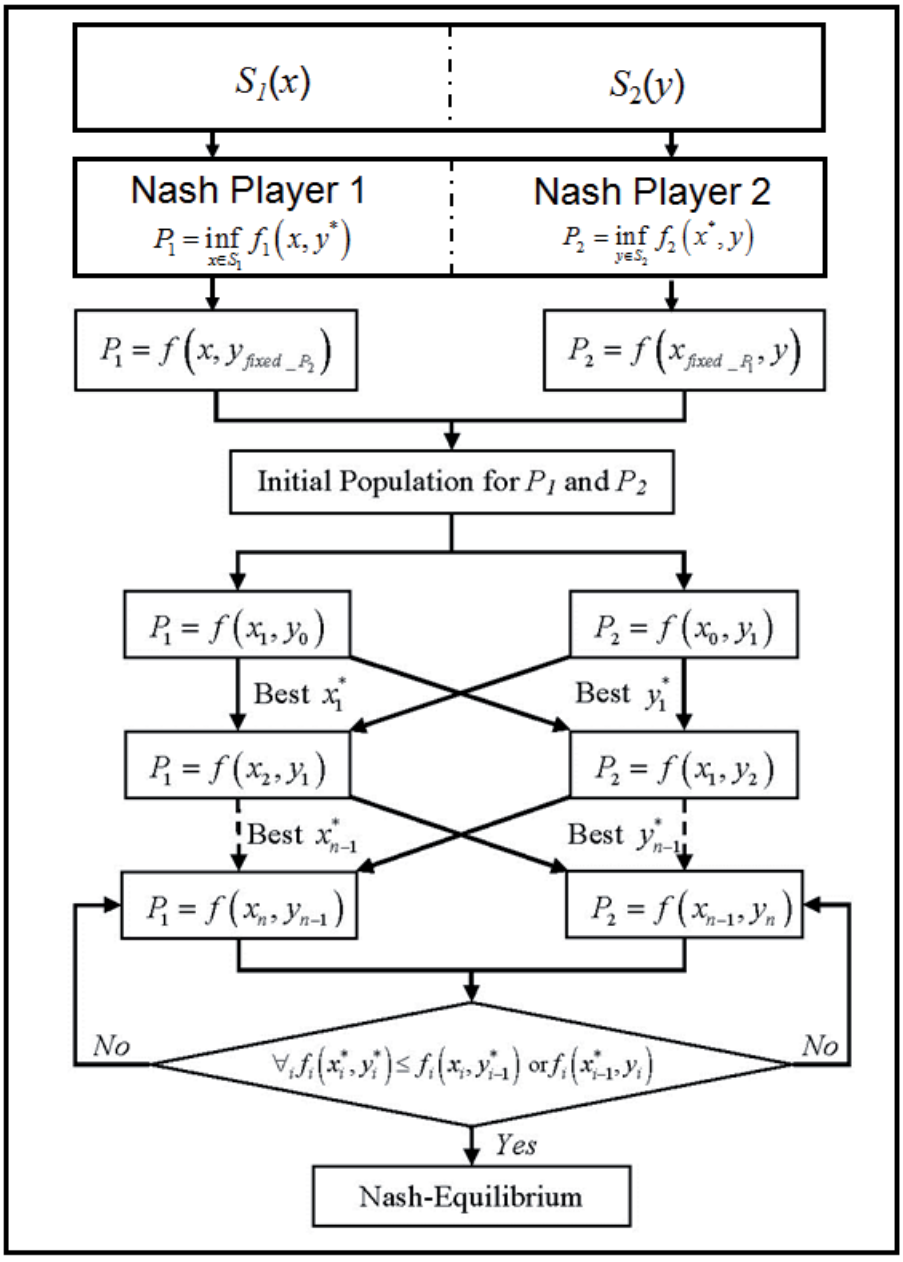

FIGURE 9 Nash GAs flow chart.

3.2.3 Merging Nash equilibrium and GAs

Here a non-cooperative multi-objective algorithm which is based on Nash equilibria is presented. In order to demonstrate the Nash approach, let us consider a Nash equilibrium for a minimization problem of two mathematical functions. There is a game played by two players, A and B, with the following objective functions:

$$\begin{cases} f_A(x_1, x_2) &= (x_1 - 1)^2 + (x_1 - x_2)^2 \\ f_B(x_1, x_2) &= (x_2 - 3)^2 + (x_1 - x_2)^2 \end{cases} \tag{42}$$

A way to find a Nash equilibrium of the minimization problem above analytically is to use the notion of *rational reaction sets*. Let D_A be the rational reaction set of Player A and D_B the rational reaction set of Player B:

$$\begin{array}{lllll} D_A(\bar{x}_1, x_2) \in \bar{S}_A \times \bar{S}_B & \text{such as} & f_A(\bar{x}_1, x_2) & \leq & f_A(x_1, x_2) \\ D_B(x_1, \bar{x}_2) \in \bar{S}_A \times \bar{S}_B & \text{such as} & f_B(x_1, \bar{x}_2) & \leq & f_B(x_1, x_2) \end{array} \tag{43}$$

It is easy to see that the rational reaction sets are the sets of the best solutions a player can achieve for different strategies of its opponent(s). The Nash equilibrium is the intersection of the two rational reaction sets D_A and D_B:

$$\begin{cases} D_A &= \left\{ x_1 \left| \frac{\partial f_A(x_1, x_2)}{\partial x_1} = 0 \right. \right\} \\ D_B &= \left\{ x_2 \left| \frac{\partial f_B(x_1, x_2)}{\partial x_2} = 0 \right. \right\} \end{cases} \tag{44}$$

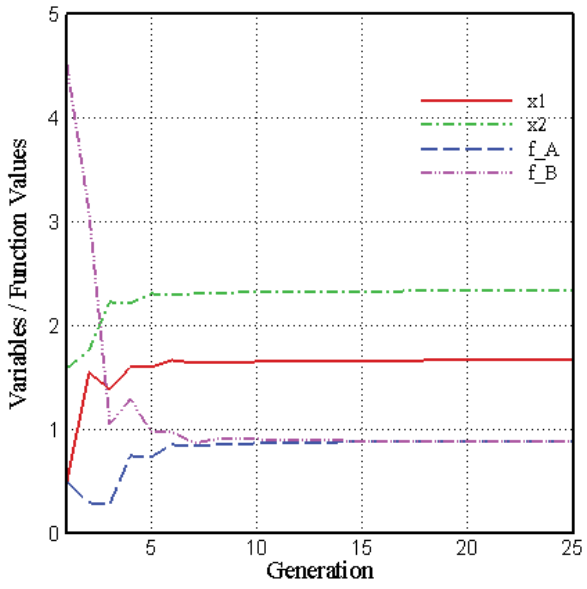

FIGURE 10 Convergence history of Nash algorithms into the Nash equilibrium $(f_A, f_B) = (0.88, 0.88)$, $(x_1, x_2) = (1.66, 2.33)$.

If we apply (44) to the mathematical example given above as (42), the intersection of the two rational reaction sets can be computed:

$$\begin{cases} \frac{\partial f_A(x_1,x_2)}{\partial x_1} = 0 \quad \Leftrightarrow \quad 2(x_1-1)+2(x_1-x_2)=0 \quad \Leftrightarrow \quad x_2 = 2x_1-1 \\ \frac{\partial f_B(x_1,x_2)}{\partial x_2} = 0 \quad \Leftrightarrow \quad 2(x_2-3)-2(x_1-x_2)=0 \quad \Leftrightarrow \quad x_2 = \frac{x_1+3}{2} \end{cases} \tag{45}$$

Solving (45) yields the results $x_1 = \frac{5}{3}$ and $x_2 = \frac{7}{3}$. Thus, the Nash equilibrium is the point $(x_1, x_2) = (\frac{5}{3}, \frac{7}{3}) \approx (1.66, 2.33)$. The corresponding values of the two objective functions are $(f_A, f_B) = (\frac{8}{9}, \frac{8}{9}) \approx (0.88, 0.88)$.

The validity of the Nash approach is tested using GAs with a population size of $NP = 30$ for each Nash player. Both the elite values and the function values during the evolutionary process are listed in Figure 10. The algorithm converges into the Nash equilibrium $(f_A, f_B) = (0.88, 0.88)$, $(x_1, x_2) = (1.66, 2.33)$ successfully within 20 generations.

The idea of Nash games can be applied to optimization. A multi-objective optimization problem can be solved as a Nash game where the different criteria are optimized by different players. For example, position reconstructions of BINACA0012 played by Nash GAs in Papers [PIII] and [PV] showed the advantage of using two Nash players to reach the targeted positions much faster than in the case of single GAs. Also, shape optimization involving minimization of drag and maximization of lift in aerodynamics can result in an adequate compromise at the Nash equilibrium if it is located close to the Pareto front, thus avoiding the time consuming Pareto search. Likewise, hybridizing Pareto games with competitive games can provide substantial speed-ups because the algorithm can be seeded with good solutions from the Nash game [28].

3.2.4 Hybridized Pareto-Nash Games

The hybridized Pareto-Nash games introduced in [27] can be interpreted as a game coalition between a cooperative Pareto game and a competitive Nash game solving simultaneously the same multi-objective optimization problem, capturing simultaneously and efficiently the Nash equilibrium and the set of Pareto non-dominated solutions. Figure 11 shows the hybridized Pareto-Nash games structure with three Nash players and Nash sub-players, with arrows indicating exchanges of information between Nash players and from Nash players to the Pareto player. Each Nash player has two hierarchical sub-players with their own optimization strategy. The elite design information (best chromosome) from Nash players are transferred to a Pareto player (in the computer to the Pareto tournament buffer) at every generation. From this coalition, an important speed up of the Pareto optimizer to capture a non-dominated solution is obtained by maintaining diversity through elite information from Nash games. Figure 12 shows a flow chart of hybridized Pareto-Nash games.

In order to illustrate the efficiency of the hybridized Pareto-Nash games, let us consider the following two-objective test case ZDT4 proposed by Zitzler-Deb-Thiele [11]. ZDT4 is a multi-modal problem with two objective mathematical

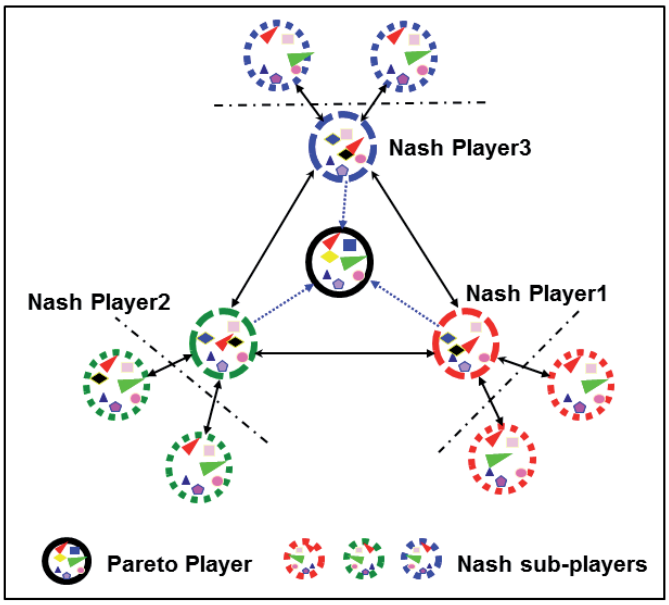

FIGURE 11 Hybridized Pareto-Nash games topology.

functions, and the multiple local Pareto front causes difficulties for many algorithms to converge to the true Pareto optimal front. The test case is analytically defined as follows:

$$\begin{cases} f_1(x_1) &= x_1 \\ f_2(g,h) &= gh \\ g(x_i) &= 1 + 10(n-1) + \sum_{i=1}^{N} \left(x_i^2 - 10\cos\left(4\pi x_i\right)\right) \\ h(f_1,g) &= 1 - \sqrt{\frac{f_1}{g}} \end{cases} \tag{46}$$

where $N = 10$, $x_1 \in [0, 1.0]$ and $x_{2-n} \in [-5.0, 5.0]$.

Using the hybridized Pareto-Nash game approach in 46, Nash player 1 and Nash player 2 only optimize x_1 and x_{2-n} by using their own strategy and objectives while the Pareto player considers cooperatively both x_1 and x_{2-n} to compute fitness functions f_1 and f_2.

Nash-equilibrium and Pareto optimal front obtained by the hybridized Pareto-Nash game coalition are compared to the *Non-Dominated Sorting genetic Algorithm II* (NSGAII Pareto software) by K. Deb [13] in Figures 13, 14, 15 with different number of generations stopping criteria 100, 150 and 200 respectively. It can be observed that the Pareto non-dominated solutions indicate a better global solution curve for the hybridized Pareto-Nash game approach than the NSGAII with the same population size of 100. In particular, the hybridized Pareto-Nash game approach shows a significantly better performance than the NSGAII at lower number of generations, but as the number of generations grows, the difference reduces to the Pareto optimal front.

Similar speed-ups obtained by a game coalition with numerical experiments on mathematical test cases ZDT1 (convex Pareto optimal front), ZDT2 (non-convex

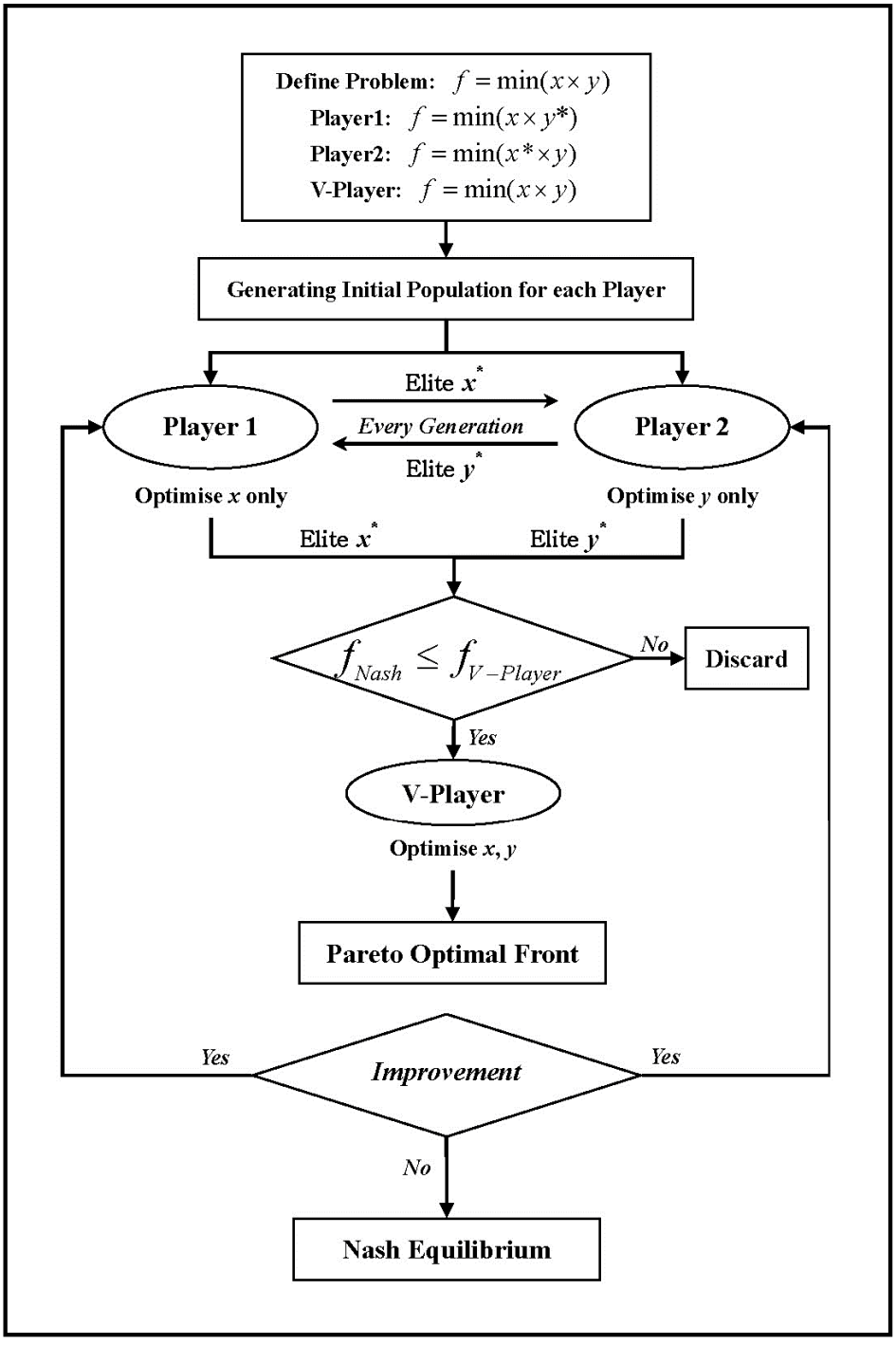

FIGURE 12 Hybridized Pareto-Nash games flow chart.

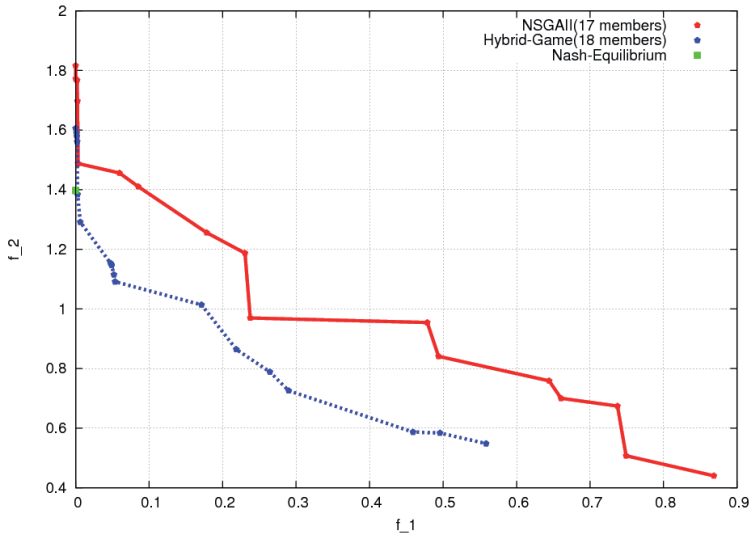

FIGURE 13 Convergence history of the ZDT4 with 100 generations.

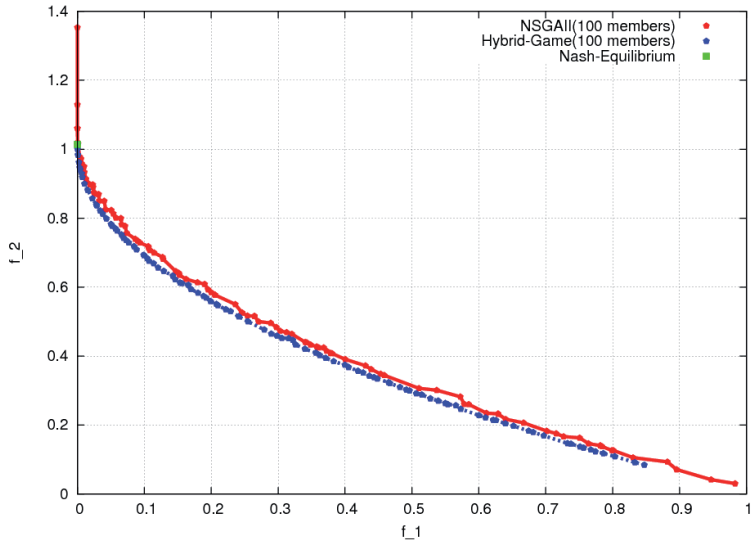

FIGURE 14 Convergence history of the ZDT4 with 150 generations.

Pareto optimal front), ZDT3 (non-continuous Pareto optimal front), ZDT6 (con-uniformly distributed Pareto front) defined in K. Deb [11] confirm the efficiency of this hybridized Pareto-Nash game. More recently, morphing wing design using the above methodology has demonstrated the potential of games coalitions in real life application [29]. For more details about performances and real applications of hybridized game with GAs in multi-objective and multidisciplinary design optimization problems in aeronautics, the reader can consult [55, 56].

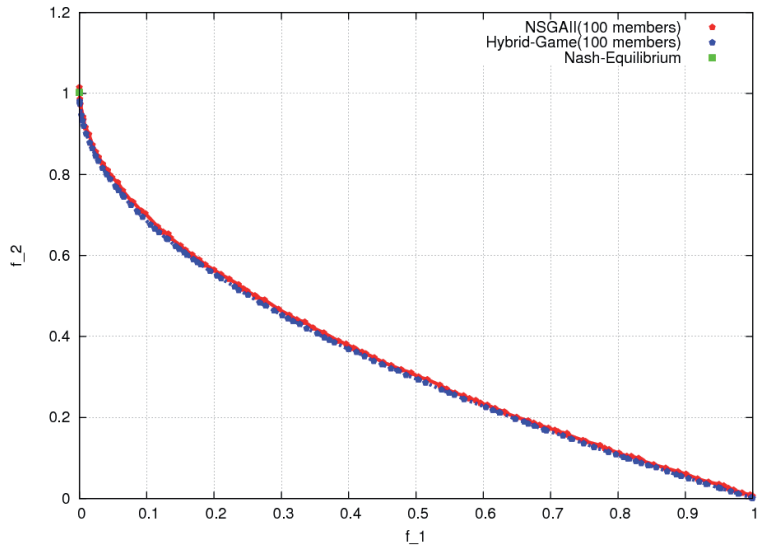

FIGURE 15 Convergence history of the ZDT4 with 200 generations.

3.3 Hierarchical Genetic Algorithms

HGAs [64], or in general, Hierarchical EAs (HEAs) are a special class of the island-model GAs [46]. They use a hierarchical topology for the layout of sub-populations. The concept of hierarchical topology is based on multi-population dealing with multi-residual of accuracy or multi-resolution of the computational points. The topology used in this research is illustrated in Figure 16 with seven sub-populations (each node stands for one sub-population). The benefits of using HGAs are a faster convergence towards a global solution and a fast search of a set of good quality solutions when compared to GAs considering only a fine mesh or a lower residual. In addition, the genetic operators parameters can be changed between different layers. In the coarse bottom layer, the objective function evaluation can be cheaper, allowing an explorative algorithm; on the accurate top level, the algorithm can exploit high-quality solutions.

The HGAs studied in this research use a three-layer hierarchical topology with seven sub-populations. The top layer has a single population with two child populations in the intermediate layer, which in turn have two child populations on the bottom layer resulting in a total of seven populations, as shown in Figure 16. The basic idea is that the top layer receives best solutions or genetic material from the intermediate layer and re-evaluates them by using high fidelity models; the intermediate layer receives random solutions from the top layer and best from the bottom layer and re-evaluates them using intermediate fidelity models; the bottom layer receives random solutions from the intermediate layer and re-evaluates them using low fidelity models. Therefore, HGAs allow the flexibility of using different mutation parameters for exploration or exploitation of the search space. In the following, the population at the bottom

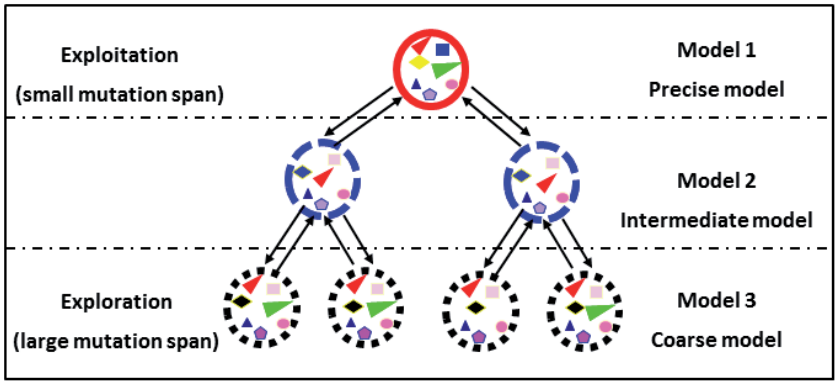

FIGURE 16 Hierarchical topology introduced by M. Sefrioui [64].

layer runs the exploration with a large mutation span while the population at the top layer runs the exploitation with a small mutation span which accelerate the optimization process.

As an example, HGAs with different levels of discretization are applied to one position reconstruction problem for an oscillating single NACA0012 airfoil in pitch at its quarter chord. The objective function is similar to the one defined in Section 5.1.2 but for one single airfoil. The targeted design parameter is the angle of attack with the value of 0.0°. The top layer uses 40 points, the intermediate layer 20 points and the bottom layer 10 points on NACA0012 airfoil. Each layer is run by the FVM Euler solver based on the Delaunay graph mapping strategy to move mesh elements in the computational domain. Figure 17 shows the comparisons of the convergence history for a single GA and HGAs in each subpopulation. After 4050 function evaluations, HGAs find the angle of attack at -0.00242034°, reaching the stopping criterion at 9.21802×10^{-08}; while GAs take 13300 function evaluations to find the angle of attack at 0.0109391°, reaching the stopping criterion at 7.31399×10^{-07}. In order to show the performance of HGAs, Figure 18 zooms on comparisons of convergence history for HGAs in each subpopulation. In terms of CPU time, the cost to solve this test case with HGAs is 98.5 minutes while GAs takes 189.3 minutes on a computer named *TELECASTER* which is equipped with Intel(R) Core(TM) 2 Quad CPU Q9650 @ 2.00GHz 2.99 GHz with 3.00 GB of RAM.

In order to take full benefit of the hierarchical topology in optimization problems, each layer uses simultaneously a different flow model and a different level of discretization. In Section 5.2, HGAs are implemented using different flow models for a real life application devoted to the drag reduction achieved by a shape and position optimized shock control bump installed on an RAE5243 airfoil. The top layer uses an accurate fast meshless or mesh Euler solver with fine clouds of points or mesh elements respectively. The intermediate layer uses a hybridized mesh/meshless Euler solver with intermediate mesh elements/clouds of points. The bottom layer uses the FVM Euler solver with a coarse mesh in order to explore efficiently the search space with a large mutation span. More detailed studies can be found in Paper [PVI].

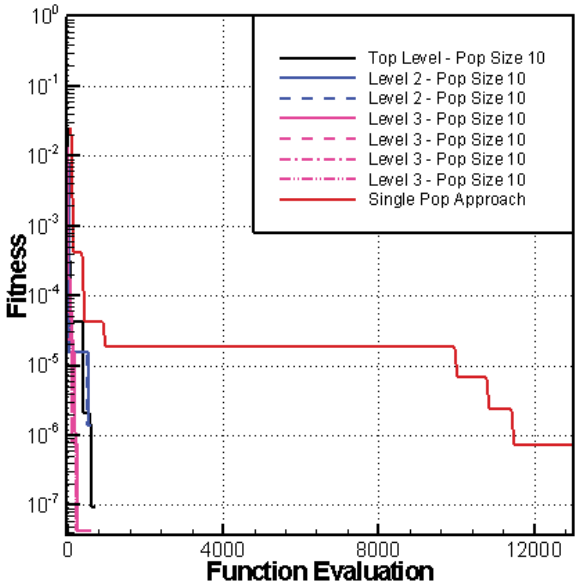

FIGURE 17 Convergence history for one oscillating NACA0012 airfoil position reconstruction using HGAs and GA.

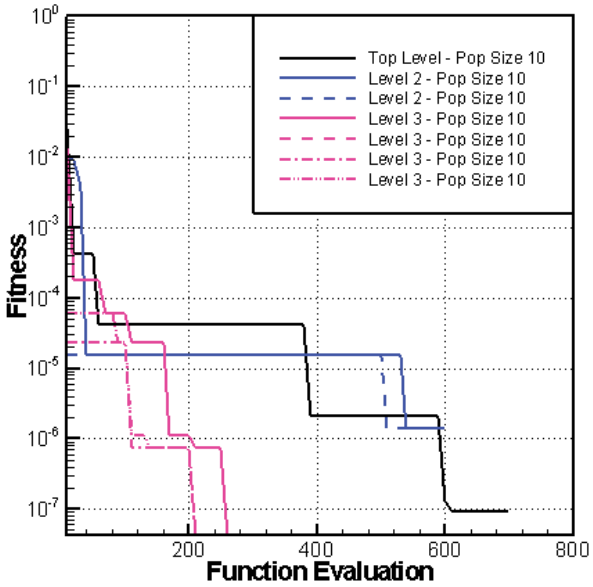

FIGURE 18 Convergence history for one oscillating NACA0012 airfoil position reconstruction using HGAs.

4 MOVING CLOUD TECHNIQUES

Optimization procedures require a technique for updating the distribution of clouds of points in order to maintain cloud quality within the framework of meshless methods. In aeronautical applications, examples of such situations abound in the simulation and optimization of high lift configurations, flow control devices, deflection of control surfaces, fuel tank separation, weapon release and many others. Various methodologies have been proposed for the movement of mesh elements/clouds of points, being either regenerated or dynamically updated. Moving the mesh elements/clouds of points is the most commonly used technique because the generation of a new mesh/meshless discretized domain is computationally expensive. The concept of the mesh/meshless updating has been adopted by many investigators. This is due to its flexibility in easily adopting different ways of modeling a fictitious elastic body, the so-called spring analogy scheme [14, 70]. It is also due to its computational efficiency compared to the mesh/meshless regeneration procedure.

As mentioned above, a popular technique, known as the spring analogy scheme, is based on a network of linear springs which solve static equilibrium equations for this network to determine new locations of points. However, existing spring analogy formulations have difficulties maintaining the clouds of points' quality for large deformation cases [71]. In this study, a novel dynamic cloud technique is proposed to move clouds of points according to new shapes or positions of aerodynamic geometries. It is capable of deforming clouds of points using the Delaunay graph mapping with ease and efficiency and guarantees the cloud quality. In Section 4.1, the dynamic cloud technique and its applications to rigid moving boundary problems are introduced and discussed. Moreover, Section 4.2 presents an algebraic adaptive meshless scheme, which locally adapts clouds of points in the computational domain, according to gradients of flow variables, iteratively during simulation or optimization procedures.

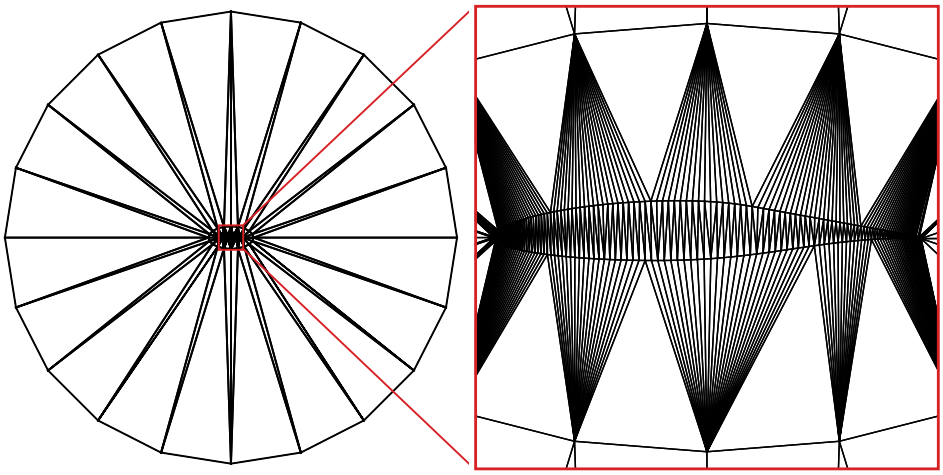

FIGURE 19 Global and close-up views of a Delaunay graph in the case of a single RAE5243 airfoil.

4.1 The dynamic cloud technique and its applications

A fast dynamic cloud technique is introduced to ensure that clouds of points can follow in an automated way the shape modifications during simulation or/and optimization procedures. This dynamic cloud technique based on the Delaunay graph mapping strategy [36] makes use of algebraic mapping principles which allow clouds of points being accurately redistributed in the flow field without any iteration. This has been adopted for the meshless methods in unsteady simulations in Paper [PI] and for design optimization problems in Papers [PII, PIII, PV, PVI].

4.1.1 Dynamic cloud technique: the Delaunay graph mapping strategy

As shown in Figure 19, a Delaunay triangulation graph of the computational field is set up with given points located on the boundaries of an RAE5243 airfoil. Since the graph covers the whole computational domain, any points within it must belong to one of the triangles in the graph. The triangulation contains every point P with coordinates (x, y) in the computational domain. If P lies inside an element T with vertices notated as E_1, E_2, E_3, the coordinates of point P can be expressed with the aid of barycentric coordinates (a_1, a_2, a_3)

$$\begin{cases} x = a_1 x_1 + a_2 x_2 + a_3 x_3 \\ y = a_1 y_1 + a_2 y_2 + a_3 y_3 \end{cases} \tag{47}$$

Here (x_i, y_i) are the Cartesian coordinates of vertex E_i. If S is the area of T and $S_i, i = 1, 2, 3$ are the areas of sub-triangles shown in Figure 20, then $a_i = \frac{S_i}{S}, i = 1, 2, 3$.

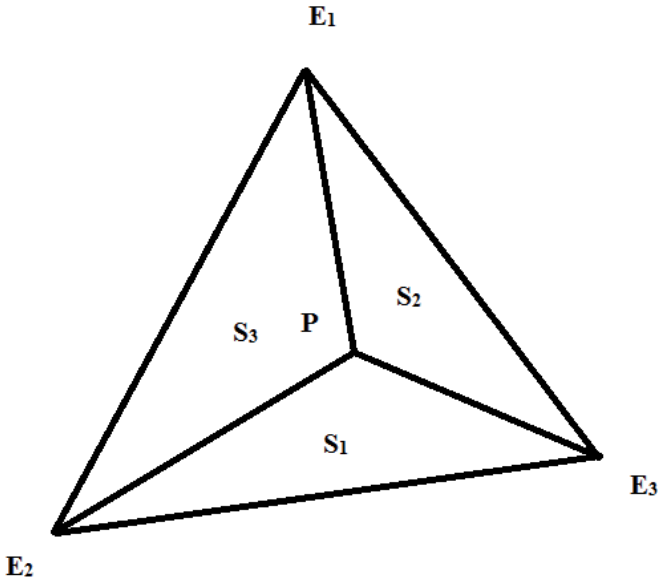

FIGURE 20 Relation of coefficients with the nodal points.

When the airfoil moves, all the background points are adjusted according to the movement of the boundary points. The vertex coordinates of the previously considered triangle become $(x_i', y_i'), i = 1, 2, 3$, and the new coordinates of the point P can be noted as

$$\begin{cases} x' = a_1 x_1' + a_2 x_2' + a_3 x_3' \\ y' = a_1 y_1' + a_2 y_2' + a_3 y_3' \end{cases} \tag{48}$$

Figure 21 illustrates the moved clouds of points for a 30° airfoil pitch, using the spring analogy scheme described in [14] while Figure 22 shows the points displacements obtained with the Delaunay graph mapping strategy. It is apparent that, compared to the spring analogy scheme, better results are achieved by the Delaunay graph mapping strategy in ensuring that clouds of points are following the movements of the body boundaries without iterations [36] in the computational domain.

4.1.2 Application to moving boundary problems

The following results are based on Paper [PI]. A standard test case of an oscillating NACA64A010 airfoil is considered. The dynamic cloud technique based on the Delaunay graph mapping strategy has been used to simulate rigid moving boundary problems within the framework of meshless methods. Spatial derivatives are directly approximated by using the WLS meshless method in each cloud of points. An upwind flux difference splitting scheme using Roe's approximate Riemann solver is used for the estimation of the inviscid flux. A dual time-stepping approach using an explicit four-stage Runge–Kutta scheme is used to

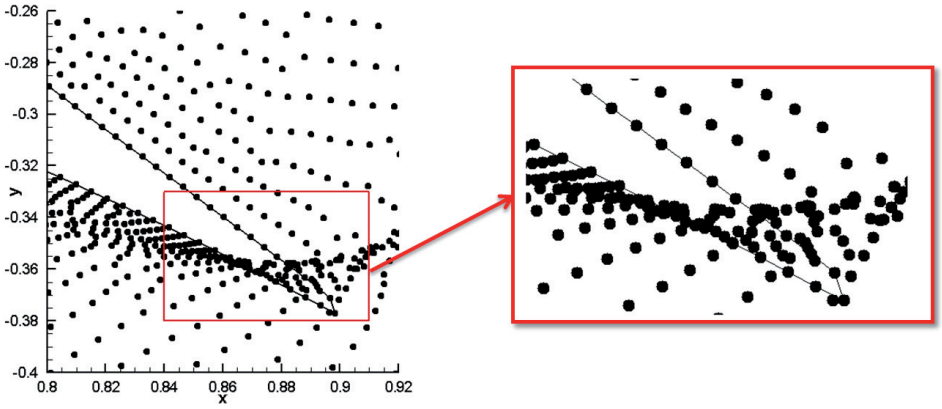

FIGURE 21 Moved clouds of points for a 30° airfoil pitch using the spring analogy scheme.

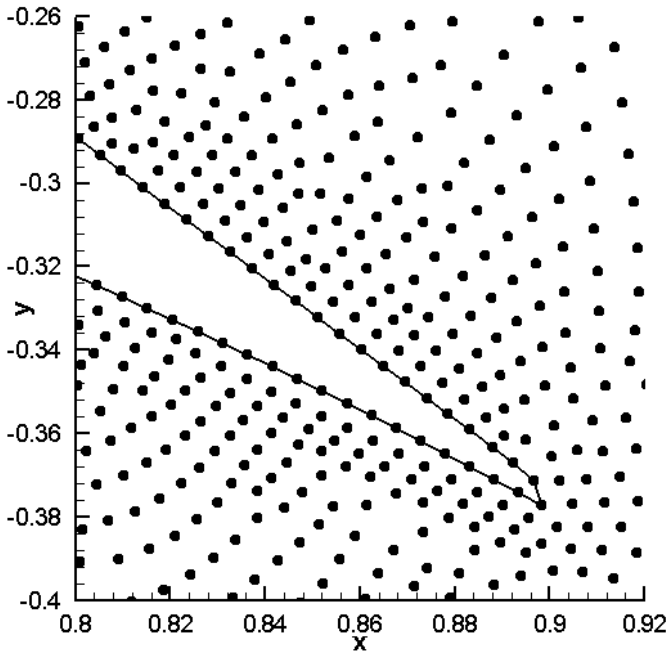

FIGURE 22 Moved clouds of points for a 30° airfoil pitch using the Delaunay graph mapping strategy.

advance the flow equations in time. The computational results are compared with the AGARD experimental data [10] and available referenced results in [70]. In this test case, the NACA64A010 airfoil oscillates in pitch about its quarter chord and the instantaneous angle of attack $\alpha(t)$ is given by

$$\alpha(t) = \alpha_m + \alpha_0 \cos(\omega t) \tag{49}$$

Here α_m is the mean angle of attack, α_0 is the magnitude of the pitching oscillation and k is the angular frequency related to the reduced frequency by the relationship $k = \frac{\omega c}{2U_\infty}$ while c is the airfoil chord length, and U_∞ is the free stream velocity. The test case is simulated with the following flow conditions:

$$Ma = 0.796, \quad \alpha_m = 0.0^\circ, \quad \alpha_0 = 1.01^\circ, \quad k = 0.202.$$

The computational domain is discretized for a single NACA64A010 airfoil with a total of 200 points on the airfoil and 4006 clouds of points in the flow field. Figure 23 shows both the global and the close-up views of the computational domain for the single NACA64A010 airfoil.

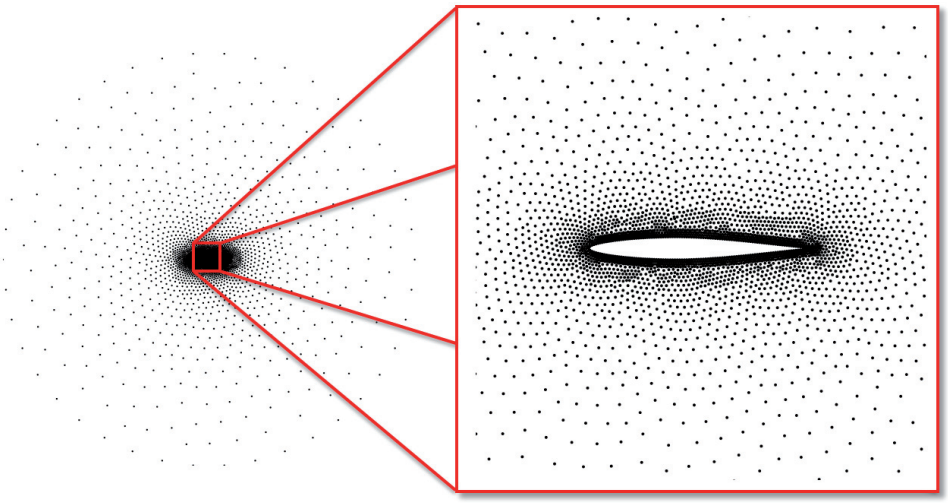

FIGURE 23 Global and close-up views of the computational domain around a single NACA64A010 airfoil.

Prior to performing the unsteady simulation, a steady flow solution is first made using the WLS meshless method with the Roe's Riemann solver for the single NACA64A010 airfoil based on the specified flow conditions as $Ma = 0.796$, $\alpha_m = 0.0^\circ$. The simulation of the unsteady flow field is then initiated from the converged steady solution. The initial steady simulation is shown in Figure 24.

The lift coefficient and the moment coefficient versus time during the oscillatory motion are presented in Figure 25 using dynamic could technique based on the Delaunay graph mapping strategy. It is clearly that the solution becomes periodic after roughly one oscillation.

Then Fourier analysis is implemented to decompose the time variation of the surface pressure coefficient into different modes. The real and imaginary components of the first mode are displayed in Figure 26. The good agreement with the AGARD experimental data [10] and available referenced results [70] demonstrate both the validity and practicality of the present method. More detail comparison and test case applications can be found in Paper [PI].

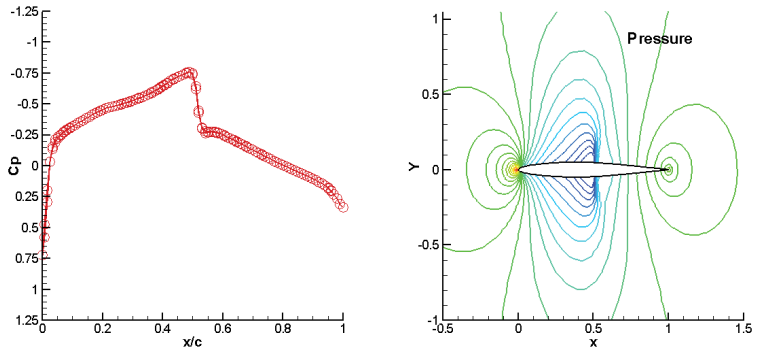

FIGURE 24 Initial flow field for simulation the prescribed oscillation of a single NACA64A010 airfoil.

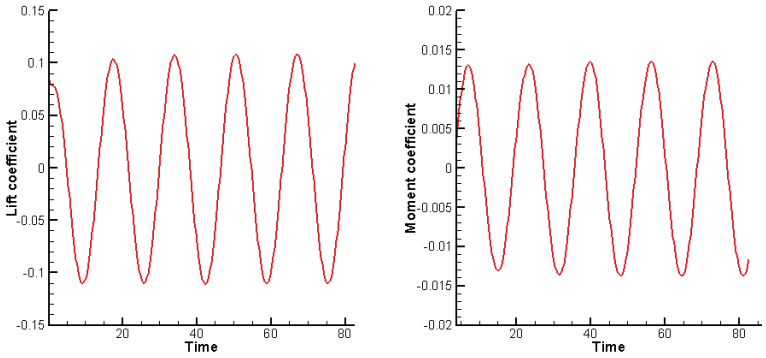

FIGURE 25 Comparison of variation of lift and moment coefficients with time for prescribed oscillation of a single NACA64A010 airfoil.

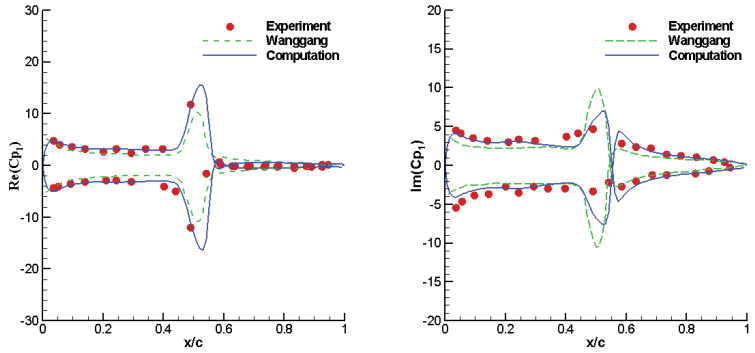

FIGURE 26 Comparisons of the first Fourier mode component of surface pressure coefficients with the AGARD experimental data and available referenced results for an oscillating NACA64A010 airfoil.

4.2 An adaptive meshless method and its applications

Based on the success of developing an efficient dynamic cloud technique which maintains the primary clouds of points qualities with rigid moving boundary problems, it is expected to obtain reliable results maintaining the same number of points in the computational domain at each time step or at each modified body shape or position in a design optimization procedure. It is therefore very important to bring an adaptive meshless method to adjust clouds of points automatically.

In past decades, many adaptive mesh schemes have been developed and become important tools for designers to increase the reliability and reduce the cost of numerical computations in many engineering problems. Through an effective adaptive scheme, the discretization error can be reduced via an automatic refinement of the computational region where the accuracy of the numerical solution is low, and therefore the prescribed accuracy [72, 1, 53] is achieved. Hsu et al. [21] proposed an algebraic mesh adaptation scheme based on the concept of arc equidistribution.

4.2.1 An adaptive meshless method

In this research, an algebraic adaptive meshless scheme based on a weighted reference radius equidistribution is presented. To illustrate this, the difference in 2D Cartesian coordinate system between center point i and its satellite point k can be written as:

$$\begin{cases} \Delta x_{ij} = x_j - x_i \\ \Delta y_{ij} = y_j - y_i \end{cases} \tag{50}$$

For each satellite point j, the distance between center point i and point j is

$$R_{ij} = \sqrt{\Delta x_{ij}^2 + \Delta y_{ij}^2} \tag{51}$$

and the reference radius of cloud C_i is defined as the longest distance between the point i and its satellite point j as

$$R_i = max(R_{i1}, R_{i2}, ..., R_{iM_i}) \tag{52}$$

where M_i is the total number of satellite points around point i. By using the concept of a weighted reference radius equidistribution for mesh from Hsu et al. [21], a weighted reference radius equidistribution for clouds of points based on the pressure gradient in the flow field can be expressed as:

$$\tilde{R}_i = \frac{1}{1 + \beta \mid \nabla P \mid_i} R_i \tag{53}$$

where β is a constant that controls the sensitivity to the pressure gradient. Thus, the movement of point j should be:

$$\begin{cases} \Delta x'_j = \left(\frac{1}{1+\beta \mid \nabla P \mid_i} - 1 \right) \Delta x_j + \Delta x_i \\ \Delta y'_j = \left(\frac{1}{1+\beta \mid \nabla P \mid_i} - 1 \right) \Delta y_j + \Delta y_i \end{cases} \tag{54}$$

considering the movement in point $i(\Delta x_i, \Delta y_i)$.

The adaptive meshless method with a moving technique is implemented and tested on a circular arc bump geometry. The geometry of the channel is depicted in Figure 27 and discretized with 2940 clouds of points distributed in the computational domain. Supersonic flow conditions are Mach number 1.6 and angle of attack 0.0°. These flow conditions are high enough to form a normal shock slightly in front of the bump. This shock bends into an oblique shock, which eventually becomes the foremost oblique shock in a lambda shock structure near the upper wall of the channel. The normal shock segment of the lambda shock has a region of subsonic flow behind it, and the rear-most oblique part of the lambda shock downward intersects the lower wall near the trailing edge of the bump. The pressure contours are shown in Figure 28.

After three rounds of moved points adaptation based on the gradient of pressure in the flow field, Figure 29 presents the moved clouds of points distribution in the computational domain. A higher resolution of the shock can be observed in Figure 30. This comparison demonstrates the advantages of the adaptive meshless method.

4.2.2 Application to shape reconstruction problems

Since the adaptive meshless method achieves higher solution without adding any points in the flow field, one shape reconstruction test case with the above circular arc bump using adaptive clouds of points is considered in this study. The flow conditions and computational points are the same as in Section 4.2.1.

Let the thickness of the circular arc bump h be selected as one design parameter. The objective function is defined according to surface pressure coefficients as:

$$\min f(h) = \sum_{i=1}^{M} \left| C_p(h) - C_p(h^*) \right|_i^2 \tag{55}$$

where M is the total number of points distributed on the surface of the upper and the lower channel. The search space for the reconstruction is the interval $h \in [2.0, 8.0]$, and h^* is the targeted design parameter. The parameters values in the GA software are the following: 30 for the size of population, 0.85 for the probability of crossover and 0.01 for the probability of mutation. The stopping

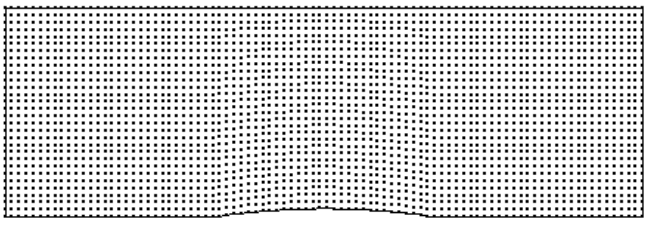

FIGURE 27 Original clouds of points for the channel with a circular arc bump.

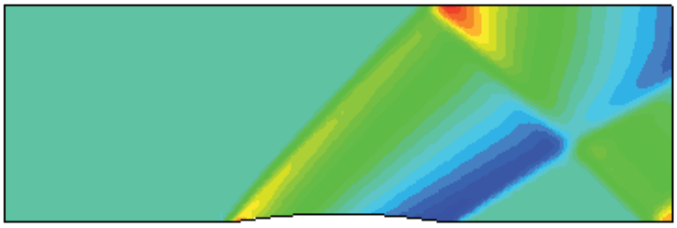

FIGURE 28 Original pressure contours for the channel with a circular arc bump.

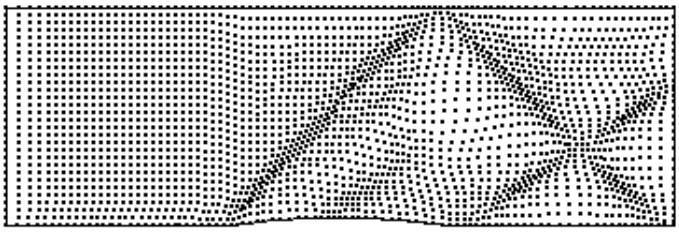

FIGURE 29 Adapted clouds of points for the channel with a circular arc bump.

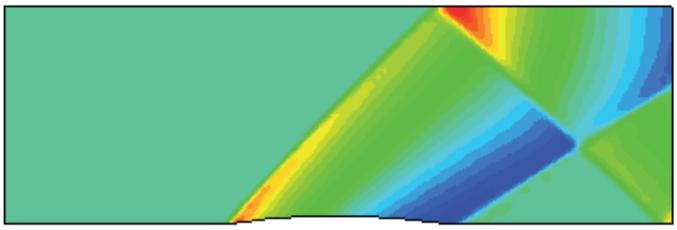

FIGURE 30 Adapted pressure contours for the channel with circular arc bump.

criteria are the fitness value $f(h) < 10^{-06}$ and the number of generation as 50.

Figure 31 shows the convergence history of the objective function reconstruction procedure. Convergence to zero of the fitness function means that GAs coupled with cloud movement have, within 40 generations, successfully rebuilt the circular arc bump with the targeted thickness. Figure 32 is the comparison of surface pressure coefficients of the targeted value and the obtained result. The red solid line stands for the targeted pressure distribution of 5% bump, and blue dots present the obtained pressure distribution of 5.01% bump. The obtained result is in good agreement with the targeted pressure distribution.

To conclude, the adaptive meshless method coupled with GAs has rebuilt the shape of the targeted circular arc bump based on the prescribed surface pressure. The results presented in this chapter are preliminary and will be consolidated by a measure of the uniformity of level of errors in the flow field to be minimized using sub-clouds and Nash algorithms as a multi-objective problems. The objective functions could be chosen as f_1, the distance between candidate

and prescribed pressure coefficients minimized in L_2 norm, and f_2, uniform level of errors minimized in L_2 norm. Furthermore, distributed optimization coupled with distributed levels of errors can both be run on High Performance Computing (HPC) in the future.

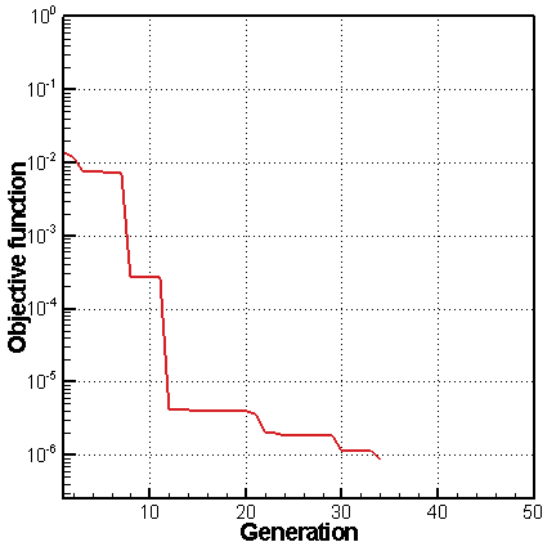

FIGURE 31 Convergence history of the objective function.

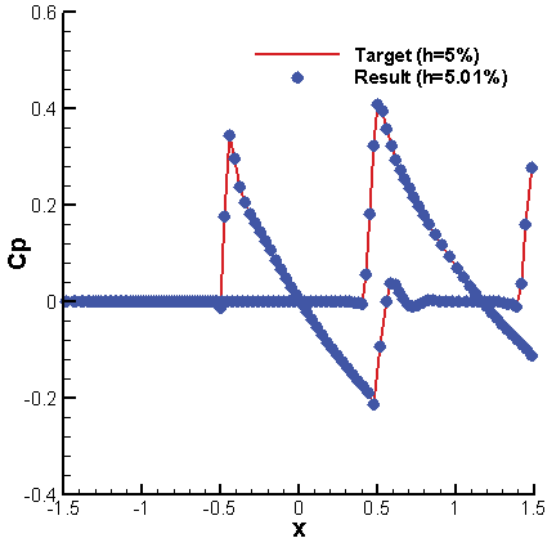

FIGURE 32 Comparison of pressure distribution of the targeted value and the obtained result.

5 EVOLUTIONARY DESIGN OPTIMIZATION WITH NASH GAS AND HGAS

In this Chapter, two optimization test cases have been selected from the Finnish design test case database developed at the University of Jyväskylä and set up by the scientific computing lab [48, 68]. The test cases BI-NACA0012 position reconstruction and 2D shock control bump optimization are computed using different optimizers (Nash GAs, HGAs and GAs) with different flow analyzers.

This chapter is divided into two sections. In Section 5.1, subsection 5.1.1 presents a numerical simulation using the hybridized mesh/meshless method on a single NACA0012 airfoil operating at transonic regime; in subsection 5.1.2, a GA and Nash GAs are chosen to rebuild the position reconstruction of BI-NACA0012 configuration using the hybridized mesh/meshless method. In Section 5.2, the HGAs methodology presented in Chapter 3 is implemented using different flow models to reduce the drag of an RAE5243 airfoil with a shock control bump installed the airfoil by optimizing its shape and location on the extrados.

5.1 Position reconstruction for BI-NACA0012

The position reconstruction test case solved in this study belongs to the Finnish design test case database (TA10, BI-NACA0012 geometry, available in [48, 68]). The following results can be found in Paper [PIII].

5.1.1 Simulation using mesh/meshless methods on a single airfoil

A single NACA0012 airfoil operating with transonic flow conditions, namely at 1.25° angle of attack and with Mach number 0.8, is selected for position reconstruction. Clouds of points are distributed in the vicinity of a single NACA0012 airfoil while mesh elements discretize the rest of the computational domain as shown in the left part of Figure 33. The boundary of the airfoil is defined with

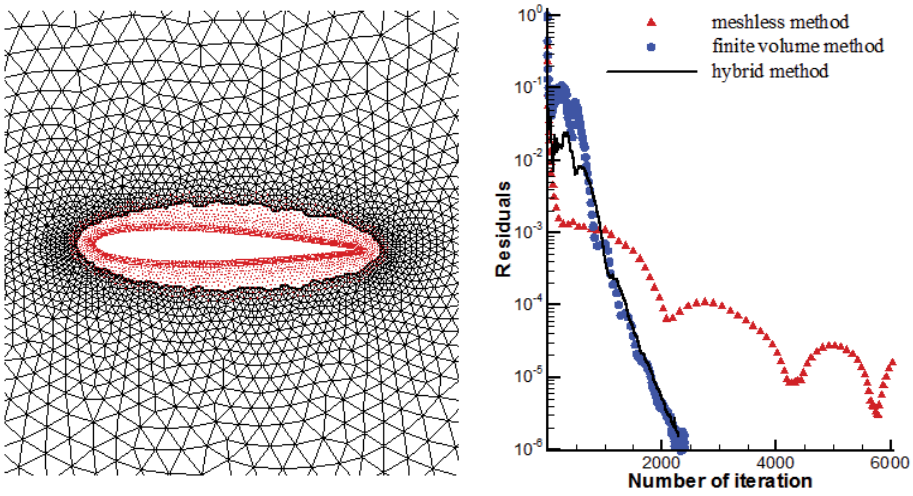

FIGURE 33 *Left:* A close-up view of the computational domain for a single NACA0012 airfoil. *Right:* Comparisons of convergence history for a single NACA0012 airfoil.

338 points. Close to the airfoil, 3208 clouds of points discretize the meshless computational sub-domain, and 5884 triangle elements discretize the computational sub-domain surrounding the meshless region. The spatial derivatives are directly approximated by using the WLS meshless method in each cloud of points. An upwind flux difference splitting scheme using Roe's approximate Riemann solver is used for the estimation of the inviscid flux, and time integration is further implemented by an explicit four-stage Runge–Kutta scheme accelerated by a local time-stepping approach and an implicit residual averaging method in this study.

The right part of Figure 33 compares the convergence history of the residual error of the Euler analyzer defined with the L_2 norm of the density. The red triangle, blue dot and black solid line represent, respectively, the level of residual errors with respect to iterations using the meshless, the finite volume method and the hybridized mesh/meshless method. The hybridized mesh/meshless method combines both the efficiency of the finite volume method and the flexibility of the meshless method.

5.1.2 Bi-NACA0012 position reconstruction using mesh/meshless methods

The number of points and the number of elements distributed in the whole computational domain is 1851 and 3472 respectively while 102 points are distributed on each NACA0012 airfoil surface. The flow conditions for computing this test case are Mach number 0.5 and angle of attack 1.0°. The left part of Figure 34 provides a partial view of clouds of points and mesh elements distributed around Bi-NACA0012.

Let Bi-NACA0012 oscillate in a pitching movement about their quarter chords, the rotating angles α_1, α_2 being selected as design parameters. The ob-

jective function of the GA optimizer is defined according to the surface pressure coefficients as

$$\min f(\alpha_1, \alpha_2) = \sum_{i=1}^{M_1} \left| C_p(\alpha_1) - C_p(\alpha_1^*) \right|_i^2 + \sum_{i=1}^{M_2} \left| C_p(\alpha_2) - C_p(\alpha_2^*) \right|_i^2 \tag{56}$$

where M_1 is the total number of points distributed on the surface of the upper airfoil and M_2 is the total number of points distributed on the surface of the lower airfoil. The search spaces are $\alpha_1 \in [-10.0^\circ, 10.0^\circ]$, $\alpha_2 \in [-10.0^\circ, 10.0^\circ]$, and α_1^*, α_2^* are targeted design parameters. Parameter values chosen for the GA optimizer are: 40 for the size of population, 0.85 for the probability of crossover and 0.01 for the probability of mutation.

To solve the same reconstruction problem using Nash GAs optimizer, the objective functions are defined as follows:

$$\begin{aligned} \min f_1(\alpha_1, \alpha_2^{**}) &= \sum_{i=1}^{M_1} \left| C_p(\alpha_1) - C_p(\alpha_1^*) \right|_i^2 + \sum_{i=1}^{M_2} \left| C_p(\alpha_2^{**}) - C_p(\alpha_2^*) \right|_i^2 \\ \min f_2(\alpha_1^{**}, \alpha_2) &= \sum_{i=1}^{M_1} \left| C_p(\alpha_1^{**}) - C_p(\alpha_1^*) \right|_i^2 + \sum_{i=1}^{M_2} \left| C_p(\alpha_2) - C_p(\alpha_2^*) \right|_i^2 \end{aligned} \tag{57}$$

Parameter values chosen for the Nash GAs optimizer are: 20 for the size of population, 0.85 for the probability of crossover and 0.02 for the probability of mutation.

On the right part of Figure 34, the convergence history of the objective function reconstruction procedure is shown using a single GA and Nash GAs fed by the hybridized mesh/meshless Euler flow analyzer. In this case, the convergence of the reconstruction using the Nash game strategy coupled with GAs is

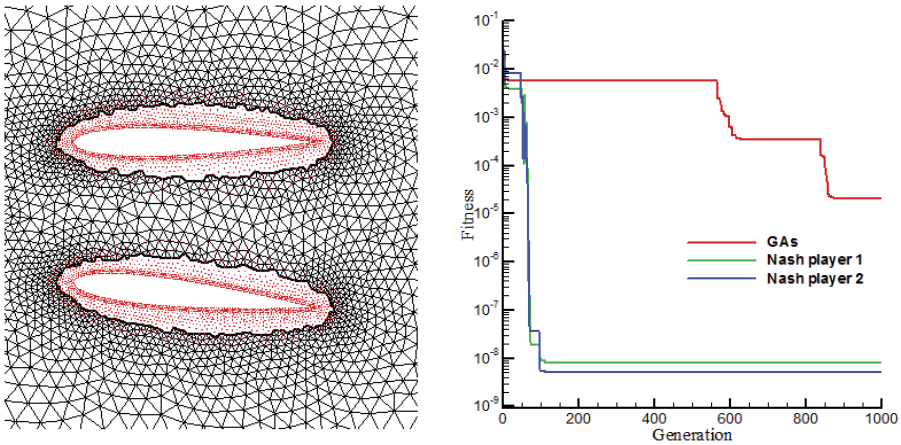

FIGURE 34 *Left:* A close-up view of the computational domain for BI-NACA0012. *Right:* Comparisons of convergence history for BI-NACA0012.

much faster than the one obtained with a single GA to rebuild the targeted BI-NACA0012 positions.

Based on the successful computerized results, hybridized mesh/meshless methods have shown significant advantages of flexibility, accuracy and efficiency of both mesh and meshless methods used for Euler flow analyzer coupled to evolutionary optimizers. These promising results open the door to solutions of more complex 2D and 3D inverse problems in aerodynamics. Other results about this optimization test case can be found in the Finnish design test case database [48] and also in Papers [V] and [VII].

5.2 A 2D Real life CFD application

Drag reduction in transonic civil aircraft has been and still is one of the long-standing interests in aeronautics design [60, 59, 30]. Despite continuous efforts made in the last two decades, reducing significantly the total drag of aerodynamic shapes remains a critical challenge to aircraft designers. This study investigates an active flow control technique called *Shock Control Bump* (SCB). The use of SCB is proposed to generate a pre-shock isentropic compression wave in order to reduce the total drag over an airfoil at a transonic speed. Numerical results illustrate how SCB modifies the flow features over the airfoil at transonic speeds and can reduce the total drag when compared to the baseline design. The following subsections demonstrate also how HGAs can improve optimization efficiency in terms of computational cost and design quality.

5.2.1 Definition of the active device CFD optimization problem

In this research, hierarchical and single GAs are applied to a real life optimization problem. For the definition of the test case, a lift-constrained optimization problem consisting of a 2D SCB installed on an RAE5243 airfoil is used. More details

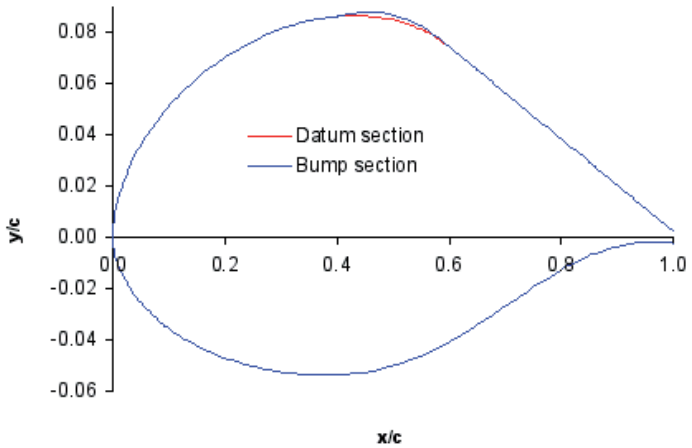

FIGURE 35 An RAE5243 airfoil with a 2D shock control bump.

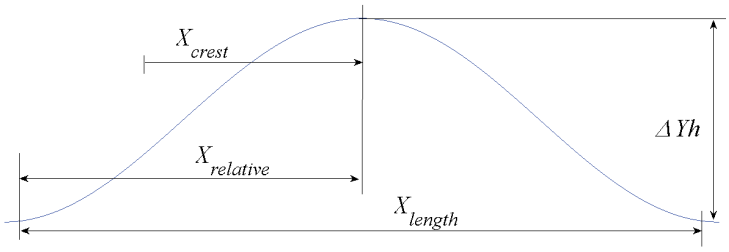

FIGURE 36 Bump design parameters.

about the definition of the test case can be found in the Finnish design test case database (TA5, 2D shock control bump optimization [48, 68]).

The objective is to minimize the drag based on flow conditions: Mach number 0.68 with a fixed lift coefficient 0.82. Figure 35 shows the SCB and an RAE5243 airfoil baseline. The four design variables, bump height, position, length and crest position, are represented in Figure 36. The design parameters (divided by the airfoil chord length which is taken by unity) are defined as bump crest position, bump starting point to crest, bump total length, and bump height. The allowed ranges of the design variables are listed in Table 2.

TABLE 2 Design parameters for 2D shock control bump.

Design parameters		Min	Max
Bump crest position	X_{crest}	0	1
Bump starting point to crest	X_{relative}	0	X_{length}
Bump total length	X_{length}	0	0.4
Bump height	ΔYh	0	0.05

In order to properly minimize the drag, a suitable Bézier spline parameterization of the bump shape is selected [18] to define the continuous shape of the SCB.

5.2.2 2D shock control bump for drag reduction

Both the single and hierarchical GAs optimizers are run with four design parameters and the relaxed iteration [22], which are based on the angle of attack update to satisfy the fixed lift coefficient constraint. Other approaches to treat equality constraints with Nash game strategies can be found in [67]. The chosen analyzer software is the Euler solver and the objective function to be minimized is the shock drag. Figure 37 shows the Mach number distribution in the Euler flow field around the baseline design.

Results of the optimization with the single GA provide the converged design parameter values of the SCB:

$$X_{\text{crest}} = 0.691;\ X_{\text{relative}} = 0.0774;\ X_{\text{length}} = 0.201;\ \Delta Yh = 0.0296.$$

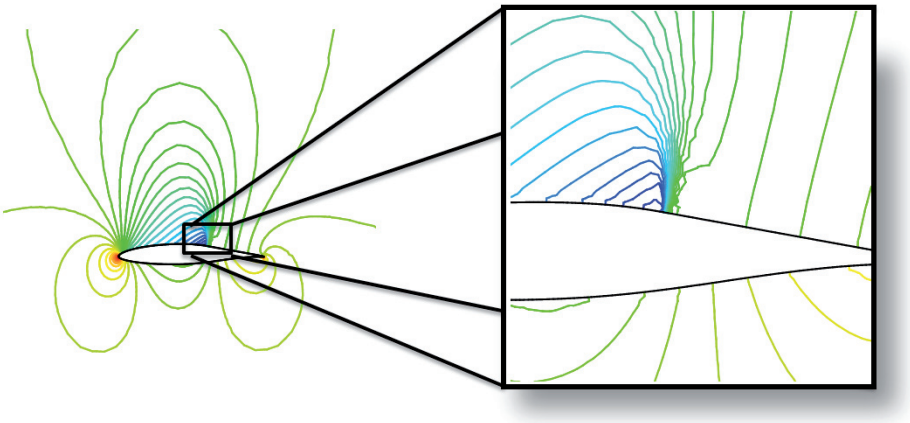

FIGURE 37 Mach number distribution in the flow field for the baseline.

The corresponding Mach number distribution in the flow field is shown in Figure 38. It can be seen that the shock is slightly weakened using the SCB.

Results of the optimization with the hierarchical GAs provide the converged design parameter values of the SCB:

$$X_{\text{crest}} = 0.705;\ X_{\text{relative}} = 0.0869;\ X_{\text{length}} = 0.254;\ \Delta Yh = 0.0299.$$

The corresponding Mach number distribution is shown in Figure 39. It can be seen that the shock is weakened using the SCB, and that the shock is less prominent than in the single GA case.

The final design parameter values are listed in Table 3. The hierarchical approach reduces the drag in this test case by 40.7%, compared to only 26% achieved with the single GA. The results show a better efficiency and accuracy of the hierarchical GAs approach compared to the single-population GA approach. This difference can be explained with two remarks. First, implementing cheaper computational models on the intermediate levels of HGAs can reduce CPU time significantly. This flexibility in choosing different models is the underlying idea behind the hierarchical topology approach. Second, the different exploration and exploitation levels can feed the optimization procedure with new genetic material enriching the quality of candidate solutions at the exploitation upper level. Of course such a fast genetic improvement is more difficult to achieve with a single-population GA limited to a single model and uniform selection – crossover –

TABLE 3 SCB design parameters obtained by the single and hierarchical GAs.

Bump design	X_{crest}	X_{relative}	X_{length}	ΔYh	Drag	Drag red. (%)
Baseline	–	–	–	–	0.02135	–
GA	0.691	0.0774	0.201	0.0296	0.01580	26.0
HGAs	0.705	0.0869	0.254	0.0299	0.01266	40.7

mutation parameters over the whole population. The results are coherent in both test case runs, confirming the validity of the hierarchical approach.

More detailed results can be found in Paper [PVI]. Other results about this optimization test case can be found in the Finnish design test case database [48]. In this study, the computation with different Euler flow analyzers focuses only on the wave drag reduction. Navier-Stokes modeling with meshless techniques is under investigation to do more realistic viscous corrections for optimization, in particular to replace the HGAs' top level modeling.

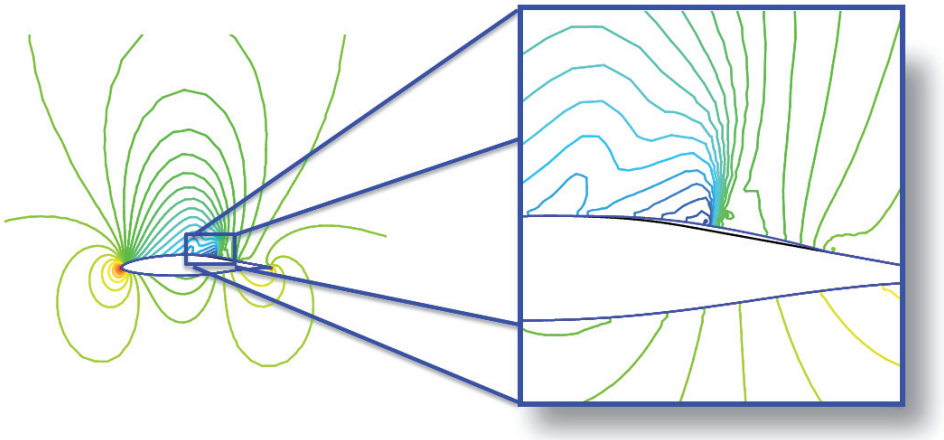

FIGURE 38 Mach number distribution in the flow field for the optimized airfoil using GA.

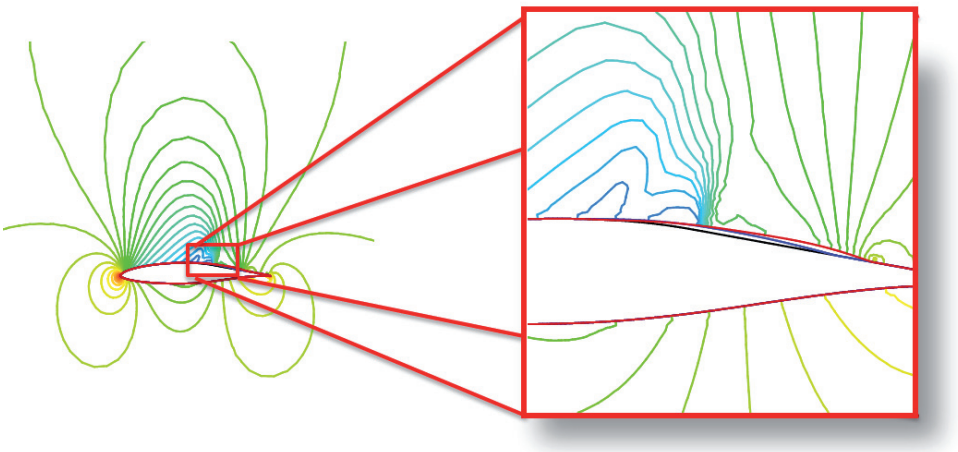

FIGURE 39 Mach number distribution in the flow field for the optimized airfoil using HGAs.

6 CONCLUSIONS AND FUTURE WORK

In this research, evolutionary design optimization methods coupled with Nash games and hybridized mesh/meshless methods are developed and evaluated on CFD problems. The three main pillars of this research are EAs, game strategies and mesh/meshless methods with two different cloud movement techniques.

A meshless algorithm using dynamic clouds of points is implemented to simulate rigid moving boundary problems, which are the main ingredients to solve optimization problems. A new fast meshless method using second and fourth order artificial dissipation developed by the author is an important ingredient bringing very useful flexibility and efficiency to the design procedure.

After reviewing different classes of evolutionary algorithms, a single GA is used to reconstruct, via prescribed surface pressures, the targeted position of an oscillating airfoil with the help of dynamic clouds of points based on a Delaunay graph mapping strategy. Numerical results qualify the flexibility of our methods for position reconstruction of aerodynamic bodies. Then, cooperative Pareto games, competitive Nash games and hybridized Pareto-Nash games are reviewed and associated to multi-objective optimization. Using the distributed and parallel computing potentiality of these games, different optimization problems are solved using various mesh, meshless and hybridized mesh/meshless methods coupled to different optimizers like GAs, Nash GAs and HGAs.

Based on numerous numerical experiments achieved on many model CFD problems, the benefit of using different mesh/meshless Euler flow analyzers coupled to different evolutionary optimizers is clearly demonstrated in this research. FVM mesh-based methods, the WLS meshless methods and hybridized mesh /meshless methods have been validated on simple model simulation and optimization test cases. Finally, the HGAs optimizer described in Chapter 3 is used for a real life optimization problem from a Finnish design test case database. Drag reduction achieved by the shape optimization of a 2D shock control bump installed on the upper surface of an RAE5243 airfoil, operating in transonic Euler flow conditions under a fixed lift constraint. The precise model of the top layer was an accurate fast meshless method. The intermediate model used a hybridized mesh/meshless method and the coarse model used a fast but less ac-

curate FVM Euler flow analyzer. Numerical results show explicitly the gain of efficiency and accuracy of the HGAs over a single population GA.

In addition to many results illustrating this research, the adaptive meshless method, without adding points in the computational domain, remains a major avenue to provide a higher quality of design optimization. Keeping in mind the important value of mesh/meshless adaption in multi-objective or multidisciplinary design, the above methods will be applied to more complex geometries and realistic flow models. These models include viscous effects such as boundary layers and turbulent Navier-Stokes flows. The targeted applications in the future include, among others, optimization of design parameters of a wind turbine and also the shape of paper machine box, requiring both multi-disciplinary/multi-physics simulation and optimization for innovative design.

YHTEENVETO (FINNISH SUMMARY)

Oheinen tutkimus käsittelee virtausdynamiikkaan liittyviä optimointiongelmia sekä laskennallisen älykkyyden käyttöä niiden ratkaisemisessa. Virtauksen simuloinnissa käytetään Eulerin virtausmallia, jonka analysointi toteutetaan tehokkaiden verkollisten ja verkottomien menetelmien yhdistelmien avulla. Optimoinnissa puolestaan hyödynnetään peliteorian työkaluja ja geneettisiä algoritmeja.

Työssä esitetään ja toteutetaan useita eri diskretointimenetelmiä Eulerin yhtälöiden ratkaisemiseksi kahdessa ulottuvuudessa. Näitä ovat tilavuusmenetelmä, Taylorin sarjaan perustuva verkoton menetelmä sekä verkollisten ja verkottomien menetelmien yhdistelmä- eli hybridimenetelmä. Työssä on toteutettu nopea verkoton menetelmä, jossa käytetään uusia keinotekoiseen dissipaatioon liittyviä parametreja. Menetelmää on testattu useilla eri testitapauksilla, ja numeeriset tulokset ilmentävät nopean verkottoman menetelmän joustavuutta, tarkkuutta sekä tehokkuutta aerodynamiikan suunnittelutehtävissä. Tekijä on kehittänyt nopeaa dynaamisen laskentapilvimenetelmän, joka perustuu Delaunayn graafin kuvautumiseen. Tätä menetelmää käytetään virtauksen analysointiin muuttuvissa geometrioissa ja muodon optimointiin.

Kuluneen vuosikymmenen aikana evoluutiopohjaiset algoritmit ovat jatkuvasti kasvattaneet suosiotaan teollisuuden monitavoitteisten ja monitieteellisten optimointiongelmien ratkaisemisessa. Deterministisiin menetelmiin verrattuina evoluutiopohjaiset algoritmit eivät vaadi tietoa gradienteista, toimivat myös tapauksissa joissa objektifunktio ei ole differentioituva, ja ovat resistantteja kohinalle. Tässä työssä yksinkertaiset geneettiset algoritmit yhdistetään Nashin peliteorioihin, ja hierarkkisia geneettisiä algoritmeja tutkitaan soveltaen niitä muodon optimoinnissa. Edellä mainittujen menetelmien suorituskykyä ja laskentatarkkuutta evaluoidaan akateemisilla testiesimerkeillä.

Lopuksi käsitellään realistista siipiprofiilin ilmanvastuskertoimen pienentämisongelmaa, jossa rajoitteena on kiinnitetty profiilin nostovoimakerroin. Tapausta, jossa standardiin RAE5243-laminaariprofiiliin halutaan muotoilla shokkiaaltoja kontrolloiva kohouma, tarkastellaan numeerisesti käyttäen yllä mainittuja menetelmiä. Ongelmaan liittyvä optimointitehtävä ratkaistaan käyttäen nopeaa verkotonta menetelmää, verkollisen ja verkottoman menetelmän hybridiä sekä tilavuusmenetelmää ja soveltaen hierarkkisia geneettisiä algoritmeja. Kattavien laskennallisten tulosten myötä todetaan, että adaptiiviset verkollisten ja verkottomien menetelmien hybridit yhdistettyinä evoluutiopohjaisiin algoritmeihin ja peliteorian työkaluihin ovat hyödyllisiä välineitä monimutkaisten virtausdynamiikan suunnittelu- ja optimointitehtävien käsittelemisessä.

REFERENCES

[1] A. Anguloa, L. Pérez Pozob, and F. Perazzo. A posteriori error estimator and an adaptive technique in meshless finite points method. *Engineering Analysis with Boundary Elements*, 33(11):1322–1338, 2009.

[2] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG)approach in computational mechanics. *Computational Mechanics*, 22:117–27, 1998.

[3] J. T. Batina. A gridless Euler/Navier–Stokes solution algorithm for complex two–dimensional applications. *NASA-TM-107631*, 1992.

[4] T. Belytschko, Y. Y. Lu, and L. Gu. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–56, 1994.

[5] J. Blazek. *Computational fluid dynamics: principles and applications*. Elsevier Science Ltd publisher, 2001.

[6] H. Q. Chen. An implicit gridless method and its applications. *Acta Aerodynamica Sinica*, 20(2):133–140, 2002.

[7] H. Q. Chen. Implicit gridless method for Euler equations. *Chinese Journal of Computational Physics*, 20(1):2003, 9–13.

[8] H. Q. Chen and C. Shu. An efficient implicit mesh–free method to solve two–dimensional compressible Euler equaitons. *International Journal of Modern Physics C*, 16(3):439–454, 2005.

[9] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi–Objective Problems*. Springer, 1 edition, 2002.

[10] S. S. Davis. NACA64A010 oscillatory and transient pitching. Technical report, AGARD Report 702 Dataset 2, 1982.

[11] K. Deb. Multi–objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal*, 7(3):205–230, 1999.

[12] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, 2001.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non–dominated sorting genetic algorithm for multi–objective optimization: NSGA–II. pages 849–858. Springer, 2000.

[14] C. Farhat, C. Degand, B. Koobus, and et al. An improved method of spring analogy for dynamic unstructured fluid meshes. *AIAA Paper*, 2070, 1998.

[15] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non–spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–87, 1977.

[16] S. K. Godunov. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Mat. Sb. (N.S.)*, 47(89):271–306, 1959.

[17] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing, 1989.

[18] P. Hartmut, B. Wolfgang, and P. Marco. *Bézier and B-Spline techniques*. Springer, 2002.

[19] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.

[20] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation*, IEEE World Congress on Computational Intelligence, 1994.

[21] A. T. Hsu and J. K. Lytle. A simple algebraic grid adaptation scheme with applications to two– and three–dimensional flow problems. *AIAA Paper*, 1984, 1989.

[22] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, Sep. 1988.

[23] A. Jameson and D. Mavriplis. Finite volume solution of the two–dimensional Euler equations on a regular triangular mesh. *AIAA Journal*, 24:611–618, 1986.

[24] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time–stepping schemes. In *AIAA 14th Fluid and Plasma Dynamics Conference*, 1981.

[25] D. J. Kirshman and F. Liu. Cartesian grid solution of the Euler equations using a gridless boundary condition treatment. *AIAA Paper*, 3974:2003, 2003.

[26] J. R. Koza. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical report, Stanford University, Stanford, 1990.

[27] D. S. Lee. *Uncertainty Based Multi-objective and Multidisciplinary Design Optimisation in Aerospace Engineering*. PhD thesis, The University of Sydney, 2008.

[28] D. S. Lee, L. F. Gonzalez, J. Périaux, and G. Bugeda. Design optimization using advanced artificial intelligent system coupled to hybrid–game strategies. In *Proceedings of the 3rd International Workshop on Artificial Intelligence in Science and Technology (AISAT 2009)*. University of Tasmania, 2009.

[29] D. S. Lee, J. Periaux, L. F. Gonzalez, and G. Bugeda. Multi-objective design optimization of morphing UAV aerofoil/wing using hybridised MOGA. In *2012 IEEE World Congress on Computational Intelligence*, 2012.

[30] D. S. Lee, J. Periaux, L. F. Gonzalez, K. Srinivas, and E. Onãte. Active flow control bump design using hybrid Nash–game coupled to evolutionary algorithms. In J. C. F. Pereira and A. Sequeira, editors, *V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, 2010.

[31] B. Van Leer. Towards the ultimate conservative difference scheme. v. a second–order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.

[32] J. Leskinen and J. Périaux. A new distributed optimization approach for solving CFD design problems using Nash game coalition and Evolutionary Algorithms. To appear in Domain Decomposition Methods in Science and Engineering, R. Bank and M. Holst and O. Widlund and J.C. Xu (Eds.), 2012.

[33] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11:83–95, 1980.

[34] W. K. Liu, Y. Chen, S. Jun, J. S. Chen, T. Belytschko, C. Pan, R. A. Uras, and C. Chang. Overview and applications of the reproducing kernel particle method. *Archives of Computational Methods in Engineering*, 3:3–80, 1996.

[35] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1181–106, 1995.

[36] X. Q. Liu, N. Qin, and H. Xia. Fast dynamic grid deformation based on Delaunay graph mapping. *Journal of Computational Physics*, 211(2):405–423, 2006.

[37] Y. Lu, T. Belytschko, and L. GuBelytschko. A new implementaion of the element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–56, 1994.

[38] L. Lucy. A numerical approach to testing the fission hypothesis. *Astronomical Journal*, 82:1013–24, 1977.

[39] H. Luo and J. D. Baumy. A hybrid cartesian grid and gridless method for compressible flow. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.

[40] Z. H. Ma, H. Q. Chen, and X. J. Wu. A gridless–finite volume hybrid algorithm for Euler equations. *Chinese Journal of Aeronautics*, 19(4):286–294, 2006.

[41] Z. Michalewicz. *Genetic Algorithm + Data structures = Evolution Programs*. New York: Springer–Verlag, 1992.

[42] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Springer, 1999.

[43] M. Mitchell. *An Introduction To Genetic Algorithms*. The MIT Press, 1977.

[44] K. Morinishi. Effective accuracy and conservation consistency of gridless type solver. In *Proceedings of the First International Conference on Computational Fluid Dynamics*, 2000.

[45] K. Morinishi. An implicit gridless type solver for the Navier-Stokes equations. *Computational Fluid Dynamics Journal*, Special Issue:551–560, 2001.

[46] H. Mühlenbein. Evolution in time and space – the parallel genetic algorithm. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, 1991.

[47] J. Nash. Non-cooperative games. *The Annals of Mathematics*, 2(54):286–295, 1951.

[48] University of Jyväskylä. Design test case database. URL: <http://jucri.jyu.fi>. Accessed: 12/11/2012.

[49] E. Onãte, S. Idelsohn, O. Zienkiewics, and R. Taylor. A finite point methods in computational mechanics, application to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 1996.

[50] E. Onãte, S. Idelsohn, O. Zienkiewics, R. Taylor, and C. Sacco. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 139:315–46, 1996.

[51] S. J. Osher. *Shock modelling in aeronautics, in Numerical Methods in Fluid Dynamics*. Academic Press, 1983.

[52] V. Pareto. *Cours D'Economie Politique*, volume I–II. F. Rouge, Lausanne, 1897.

[53] F. Perazzoa, R. Löhnerb, and L. Perez-Pozo. Adaptive methodology for meshless finite point method. *Advances in Engineering Software*, 39(3):156–166, 2008.

[54] J. Périaux, H. Q. Chen, B. Mantel, M. Sefrioui, and H. T. Sui. Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. *Finite Elements in Analysis and Design*, 37(5):417–429, 2001.

[55] J. Périaux, L. F. Gonzalez, and D. S. Lee. Introduction to optimization and multidisciplinary design in aeronautics and turbomachinery. Von Karman Institute for Fluid Dynamics, May 2010. Lecture Series.

[56] J. Périaux, L. F. Gonzalez, and D. S. Lee. *Evolutionary Optimisation and Game Strategies for Advanced Multi physics Design in AerospaceEngineering*. Springer, 2013. To appear.

[57] T. H. Pulliam. Artificial dissipation models for the Euler equations. *AIAA Paper*, 24(12), 1986.

[58] T. H. Pulliam and J. L. Steger. Recent improvements in efficiency, accuracy, and convergence for implicit approximate factorization algorithms. *AIAA Journal*, 85–0360, 1985.

[59] N. Qin, W. S. Wong, and A. Le Moigne. Three-dimensional contour bumps for transonic wing drag reduction. *Journal of Aerospace Engineering*, 222:619–629, 2008.

[60] N. Qin, Y. Zhu, and S. T. Shaw. Numerical study of active shock control for transonic aerodynamics. *International Journal of Numerical Methods Heat Fluid Flow*, 14(4):444–466, 2004.

[61] I. Rechenberg. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (in German)*. Frommann-Holzboog Verlag, Stuttgart-Bad Cannstatt, 1973.

[62] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[63] J. D. Schaffer. Multiple objective optimization with vector evaluated Genetic Algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, number 8, pages 93–100, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc.

[64] M. Sefrioui and J. Périaux. A hierarchical genetic algorithm using multiple models for optimization. *Parallel Problem Solving from Nature PPSN VI*, pages 879–888, 2000.

[65] S. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *tionary Computation*, 2:221–248, 1994.

[66] R. Storn and K. Price. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, 1995.

[67] Z. L. Tang, J. Periaux, and J. Dong. Constraints handling in nash /adjoint optimization methods for multi objective aerodynamic design. *Journal of Computational Physics*, 2012. under revision.

[68] T. Varis and T. Tuovinen. Open benchmark database for multidisciplinary optimization problems. In *11th International Conference on Modeling and Applied Simulation*, Austria, 2012.

[69] J. von Neumann and M. Oskar. Theory of games and economic behavior. *Princeton University Press*, 1944.

[70] G. Wang, Y. D. Sun, and Z. Y. Ye. Gridless solution method for two-dimensional unsteady flow. *Chinese Journal of Aeronautics*, 18(1):8–14, 2005.

[71] X. Zhou and H. Q. Xu. Gridless method for unsteady flows involving moving discrete points and its applications. *Engineering Applications of Computational Fluid Mechanics*, 4(2):276–286, 2010.

[72] Z. Q. Zhu, P. Wang, and X. B. Lu. Adaptive muhigrid solution of the 2D–Euler equations on an unstructured grid. *Acta Mechanica*, 144:43–56, 2000.