Science in China Series E: Technological Sciences

Springer

www.scichina.com
tech.scichina.com
www.springerlink.com

Distributed optimization using virtual and real game strategies for multi-criterion aerodynamic design

TANG ZhiLi[1], BAI Wen[2†] & DONG Jun[3]

[1] College of Aerospace Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China;
[2] Chinese Aeronautics Computation Technique Research Institute, Xi'an 710068, China;
[3] Chinese Aerodynamic Research Institute on Aeronautics, Shenyang 110034, China

This paper introduces the virtual and real game concepts to investigate multi-criterion optimization for optimum shape design in aerodynamics. The constrained adjoint methodology is used as the basic optimizer. Furthermore, the above is combined with the virtual and real game strategies to treat single-point/multi-point airfoil optimization. In a symmetric Nash Game, each optimizer attempts to optimize one's own target with exchange of symmetric information with others. A Nash equilibrium is just the compromised solution among the multiple criteria. Several kinds of airfoil splitting and design cases are shown for the utility of virtual and real game strategies in aerodynamic design. Successful design results confirm the validity and efficiency of the present design method.

multi-criterion optimization, Nash equilibrium, virtual game, real game, aerodynamics, distributed computing

1 Introduction

Engineering design by its very nature is multi-objective, and often requires tradeoffs between disparate and conflicting objectives since improving one worsens the other[1,2]. So, the multi- criterion optimization can be described as a methodology for the design of systems where the interactions between several criteria must be considered, and where the designer is free to significantly affect the system performance in more than one objective. The pervasiveness of these tradeoffs in engineering design has given rise to a rich and vast array of methods and approaches for multi-objective and multi-criterion optimization. Examples include the weighted sum and compromise programming approaches[1], genetic algorithm-based approaches[2,3], game theory based approaches[4−9], Pareto point approximation methods[10], and some 'brute force' approaches such as

Received January 3, 2008; accepted April 15, 2008; published online September 10, 2008
doi: 10.1007/s11431-008-0150-5
†Corresponding author (email: wenbai@263.net)
Supported by the National Natural Science Foundation of China (Grant No.10372040)

parameter space investigation[11].

A straightforward approach to solve the multi-objective optimization problem is to aggregate the different criteria with different weights, and then minimize the resulting function. Three drawbacks exist in this method: (1) not all the solutions are found； (2) in a 'penalty-function' approach, the weights assigned to some criteria may not be suitable and the resulting function may lack significance； (3) the linear combination is meaningless when multiple objectives are conflicting. Many researchers have studied the limitations of the weighted sum approaches to capture the Pareto set in non-convex problems[12]. Messac et al.[13] derived quantitative conditions for determining whether or not a Pareto point can be captured with a given objective function formulation. Das and Dennis[14] also examined the drawbacks of using weighted sums to find the Pareto set during multi-criterion optimization, noting that an evenly distributed set of weights fails to produce an even distribution of points in the Pareto set. Their observations led to the development of the Normal-Boundary Intersection (NBI) method to parameterize the Pareto set and generate an evenly distributed set of points in the Pareto set using an evenly distributed set of parameters[15].

Nowadays, Evolutionary Algorithms (EAs) benefit from its robustness in capturing convex, non-convex, discrete or discontinuous Pareto fronts of multi-objective optimization problems. So EAs and Pareto front concept are used more and more in solving practical design problems in the industry[3,16−19], and this new combination of EAs with Pareto front concept is called multi-objective evolutionary algorithms (MOEAs). But this evolutionary process is time consuming since a large number of evaluations of the objective functions, of the order of thousands, are necessary to obtain an acceptable solution.

Wilson et al.[10] proposed a method that employs design of experiments (e.g., central composite designs, orthogonal arrays, and Latin hypercubes) and surrogate approximations (e.g., response surfaces and kriging models) to facilitate exploring and capturing the Pareto frontier. The surrogate approximations are used in lieu of the computationally expensive analysis to explore the multi-objective design space and identify a rich set of potential points along the Pareto frontier. Candidate points can then be used to obtain the actual (or near actual) Pareto frontier from the original analysis codes after identifying good designs for the multiple competing objectives.

In refs. [20, 21], we introduced a novel approach combining adjoint-variable technique with a formulation derived from game theory[5] to treat multi-point airfoil optimization problems. In a symmetric Nash game, each player attempts to optimize one's own target with exchange of symmetric information with the others[5]. A Nash equilibrium is reached when each player, constrained by the strategies of the others, cannot further improve one's own target[6,7]. Here, we follow the main idea developed in refs. [20, 21] and extend it into more practical multi-objective aerodynamic designs and provide detailed numerical implementation. Several kinds of airfoil splitting and design cases are shown for the virtual and real game strategies. Successful design results confirm the validity and efficiency of the present design methods. A method based on optimal control theory[22−25], derived from the continuous Euler equations, to evaluate the gradient is used as the basic optimizer. As mentioned in refs. [23, 24], only relatively few constraints can be implemented in the adjoint method, but optimization with constraints is practical in aerodynamic design, such as minimization of the drag under fixed lift mode. Recently, we have developed the constrained adjoint method under the framework of control theory. All the constraints were satisfied implicitly and automatically in the design, and the boundary condition of adjoint equation and gradient expression of the constrained optimization problem were presented in details in ref. [26]. Here this constraint implementation methodology is used in game strategies to treat constrained multi-objective optimization problem.

2 General formulation of constrained adjoint approach for aerodynamic optimum-shape design

In the practical aerodynamic design, a design usually has to satisfy a number of constraints. Here, we summarize the major conclusions of the constrained adjoint approach developed in ref. [26], where the constraint was satisfied implicitly and automatically.

A typical shape optimization problem can be stated as follows:

Let Ω be a subspace of $\Re^2$ or $\Re^3$. Find the shape of Γ_c, a boundary of Ω (see Figure 1(a)), controlled by design variables b, such that the functional

$$\begin{cases} J = J(w,b) = \int_{\Gamma} \Phi(w,b) \mathrm{d}\Gamma = \langle \Phi(w,b), 1 \rangle_{\Gamma}, \\ \text{subject to} \qquad H(w,b)\big|_{\Gamma_c} = H^* \end{cases} \tag{1}$$

computed on a boundary Γ (where Γ and Γ_c can be different) is minimized (or maximized). The state variables w are given over the domain Ω by the PDE (governing equation of the flow field):

$$R(w,b) = 0 \quad \text{in } \Omega \tag{2}$$

with the boundary condition: $B(w,b)\big|_{\Gamma_c} = 0$.

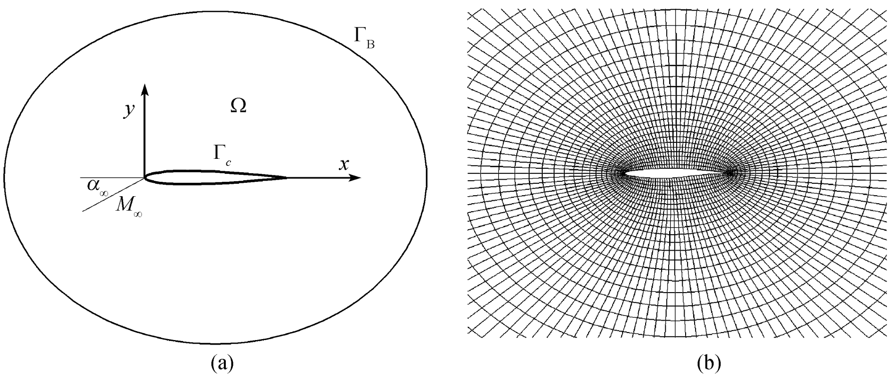

Figure 1 (a) Computational domain; (b) partial view of O-type mesh.

In general, $H(w,b)$ is in the integral form in aerodynamics, such as lift or drag. Let us assume $H(w,b) = \int_{\Gamma_c} \varphi(w,b) \mathrm{d}\Gamma_c = \langle \varphi(w,b), 1 \rangle_{\Gamma_c}$. Here, the governing equation is the two-dimensional Euler equations. According to the control theory, two adjoint variables (Lagrangian multipliers) Ψ and χ are introduced . Then the optimization problem becomes

$$J = \langle \Phi(w,b), 1 \rangle_{\Gamma} - \langle \Psi, R(w,b) \rangle_{\Omega} + \langle \chi, B(w,b) \rangle_{\Gamma_c} + \frac{\Theta}{2}\left(H(w,b) - H^*\right)^2, \tag{3}$$

where Θ is an arbitrary positive number and $\langle \cdot, \cdot \rangle$ denotes an interior product. In order to eliminate the multiple flow field evaluations, control theory requires to solve the following adjoint equations:

$$\begin{cases} L^T(\Psi)=0 & \text{in } \Omega, \\ L_n^T \Psi = \partial_w \Phi(w,b) & \text{on } \Gamma_b, \\ L_n^T \Psi = \chi \partial_w B(w,b) + \partial_w \Phi(w,b) + \Theta\left(H(w,b)-H^*\right)\partial_w \varphi(w,b) & \text{on } \Gamma_c, \end{cases} \tag{4}$$

where L is the linearized operator of R, and L_n is the spatial operator L projected on the boundaries. (n_x, n_y) is the outward unit vector norm to the surface. Then, the gradient can be computed as follows:

$$G_{\mathrm{rad}}^T = \left\langle \chi \partial_b B(w,b) + \Theta\left(H(w,b)-H^*\right)\partial_b \varphi(w,b), 1\right\rangle_{\Gamma_c} + \left\langle \partial_b \Phi(w,b), 1\right\rangle_{\Gamma} - \left\langle \Psi, \frac{\partial R}{\partial b}\right\rangle_{\Omega} \tag{5}$$

which is the function of state variables w, co-state variables Ψ, χ and design variables b. Once the gradient is established, any descent procedure can be used to obtain the design improvement, e.g. the steepest descent method.

3 Nash game strategy for multi-criteria optimization problems

Nash Equilibrium is a steady state solution concept which is a collection of strategies by various players such that no player can improve his outcome by changing only his own strategy. In this paper, we use the theory of Nash equilibrium to study the multi-objective optimum design in aerodynamics, especially in the designs where multiple critera are mutually conflicting.

3.1 Definition of Nash Equilibrium

Suppose that the multi-objective optimization problem can be stated as follows:

$$\text{Minimize (or Maximize) } J_i(x), \quad i=1,\dots,N, \tag{6}$$

where J_i are the cost functions, N is the number of objectives, x is a vector whose K components are the design or decision variables.

For an optimization problem with N objectives defined in formulation (6), a Nash strategy consists in having N players with each optimizing his own criterion. However, each player has to optimize its criterion given that all the other criteria are fixed by the rest of the players. When no player can further improve his criterion, the system has reached a state of equilibrium called Nash Equilibrium.

Let X_i be the search space for the i-th criterion, $X_i \subset X = X_1 \otimes, \dots, \otimes X_i, \dots, \otimes X_N$, an action $x_i \in X_i$. A strategy pair $\left(x_1^*, x_2^*, \dots, x_N^*\right) \in X$ is said to be a Nash Equilibrium if and only if $\forall i$

$$J_i\left(x_1^*, \dots, x_i^*, \dots, x_N^*\right) = \inf_{x_i \in X_i} J_i\left(x_1^*, \dots, x_{i-1}^*, x_i, x_{i+1}^*, \dots, x_N^*\right). \tag{7}$$

Alternatively,

$x^* = \left(x_1^*, x_2^*, \dots, x_N^*\right)$ is said to be a Nash equilibrium if and only if $\forall i$, $\forall x_i$,

$$J_i\left(x_1^*, \dots, x_{i-1}^*, x_i^*, x_{i+1}^*, \dots, x_N^*\right) \leqslant J_i\left(x_1^*, \dots, x_{i-1}^*, x_i, x_{i+1}^*, \dots, x_N^*\right). \tag{8}$$

This alternative formulation of the definition points us to a (not necessary efficient) method of finding Nash Equilibrium: First calculate the best choice for each player at the present step, then exchange their decisions to repeat the decision making procedure[7].

3.2 Algorithm implementation

The following step consists of merging adjoint method and Nash strategy in order to give a complete description on the procedure for building the Nash equilibrium between the confronted criteria in multi-objective aerodynamics optimization. Here, players are optimizers who are competitive with the others. So, at first we split the design variables into as many subsets as the targets. Each subset is associated with a player. The splitting of design variables depends on the physics of aerodynamics. Then allocate design targets to the players. According to the definitions in (7) and (8), each player optimizes his own criterion by modifying his own subset (design variables) while keeping the other subsets unchanged with exchange of information with others[4,20,21].

Let us consider N players optimizing a set of N objectives $\left(J_1, J_2, ...J_N\right)$. The optimization variables are distributed among the players in such a way that each player handles a subset of the set of optimization variables. Let $\left(x_1, ...x_N\right)$ be the optimization variables (each x_i can be a vector or scalar variable), where $x \in X$ and $X = X_1 \otimes X_2 \otimes ... \otimes X_N$, $x_i \in X_i$. We further assume that all the targets are minimization problem for convenience. Player i is responsible for J_i by modifying x_i, so the design problems can be explained as follows:

$$\text{Player } i: \min_{x_i \in X_i} J_i\left(x_1, x_2 ... x_N\right), \quad i = 1, 2, ...N, \tag{9}$$

where x_i is the free design variable of cost function J_i, all x_k, $k \neq i$ are fixed in Player i and come from the result of Player k.

The Nash/Adjoint will then work by using a same starting point, say $x^0 = \left(x_1^0, ..., x_N^0\right)$. The first player will optimize x_1 using criterion J_1 while the other variables are fixed by the other players. The second player will optimize x_2 using criterion J_2 while the other variables are fixed by the other players, and so on. Each player sends his best choice to the others at the end of every Nash design cycle. Say, the starting point at m step is the $x^{m-1} = \left(x_1^{m-1}, ..., x_N^{m-1}\right)$, where x_i^{m-1} is the best design found by Player i at $m-1$ step.

Then Player i optimizes x_i starting from x_i^{m-1} by using x_k^{m-1}, $k = 1, 2, ..., N$, $k \neq i$, and the best solution of player i at step m is

$$J_i\left(x_1^{m-1}, ..., x_{i-1}^{m-1}, x_i^m, x_{i+1}^{m-1}, ..., x_N^{m-1}\right) = \inf_{x_i \in X_i} J_i\left(x_1^{m-1}, ..., x_{i-1}^{m-1}, x_i, x_{i+1}^{m-1}, ..., x_N^{m-1}\right). \tag{10}$$

At the end of the Nash design cycle, each player sends his best solution to produce the best global solution of m step $x^m = \left(x_1^m, ..., x_N^m\right)$. Nash Equilibrium is reached when no player can further improve his criterion.

In general, for any $i, j \in \left[1, N\right]$ if $J_i = J_j = J$ or deal with a subset of J' such that $\sum_{i=1}^N J_i' = J$ wher $J_i' > 0$, we call the above game the Virtual Game; if $\exists i, j \in \left[1, N\right]$ such that $J_i \neq J_j$ when $i \neq j$, we call the above game the Real Game. With this definition, the game theory can also be used to capture the optimum solution for a single point design problem using a virtual game approach. This potential ability is further demonstrated below by numerical experiments. Lastly, it is observed that the traditional penalty method is of delicate use in multi-objective design with conflicting targets. The present approach provides an easy way to organize the concurrency of criteria in a multi-objective design, as demonstrated below in the practical test cases.

4 Numerical implementation

4.1 Implementation of adjoint method in Nash game strategies

Different from the global gradient computation in single-objective optimizations, the Nash game needs the partial gradient of the cost function with respect to the partial design variables. In section 2, we gave the global gradient computation by Adjoint method, and the partial gradient is just the global gradient projected onto the corresponding subspace. The projection matrix is the same as that of the global design variable space projected onto the subspace of the partial design variables.

For example, the n-dimensional global design variable space is X and its m-dimensional subspace is X', where $m \leqslant n$. The projection matrix from X to X' is $A_{m\times n}$, where $A_{m\times n}$ satisfies $X' = A_{m\times n} X$. Then the relation between global gradient G_{rad} and partial gradient G'_{rad} is (suppose cost function is J)

$$G'_{\text{rad}} = A_{m\times n} G_{\text{rad}}, \qquad G'_{\text{rad}} = \frac{\delta J}{\delta x'},\ x' \in X', \qquad G_{\text{rad}} = \frac{\delta J}{\delta x},\ x \in X \quad X' \subset X. \tag{11}$$

4.2 Distributed computing and numerical procedure

4.2.1 Distributed computing. One important concern related to the multi-objective optimum aerodynamic shape design is the computational effort demand because we have to analyze multiple flow fields and adjoint fields at each design point. Distributed computing on multiple processors is the underlying technology which makes an integrated design system possible by providing the computational resource necessary to achieve acceptable execution time. In a symmetric Nash game, the players make simultaneous and independent decisions, no player is informed of the choice of another prior to making a decision. Moreover, each player must be concerned only with its instantaneous payoff and ignore the effects of his current action on the other players' future behavior. Hence, each player (optimizer) can make his decision in a separate processor. After the number of design cycles is equal to the frequency of information exchange (i.e. K in Figure 2), each player has optimized the design variables corresponding to his own criterion, and is prepared to exchange information with others. The process is continued until no player can improve his outcome by changing only his own strategy. Then we say the equilibrium point of the game has been reached. This equilibrium point is the solution of the multi-objective optimization problem and the design procedure stops.

4.2.2 Numerical procedure. Suppose for the dual criterion optimization problem, we have two design points and two cost functions J_1 and J_2, and both are minimization problems. So we have two players, Player1 is responsible for J_1, and Player2 is responsible for J_2. A numerical procedure of Nash/Adjoint method is described below:

Step 0 (Initialization). Specify Nash strategy and split the design variables X according to the physics of the optimization problem and the flow field characteristics. For example, $X = X_1 \otimes X_2$, and Nash strategy is as follows:

$$\text{Player1: } \min_{X_1} J_1 = J_1(X_1, X_2), \quad \text{Player2: } \min_{X_2} J_2 = J_2(X_1, X_2).$$

Optimization starts from an initial guess $X^{\text{old}} = X_1^{\text{old}} \otimes X_2^{\text{old}}$.

Do loop for Nash strategy cycles.

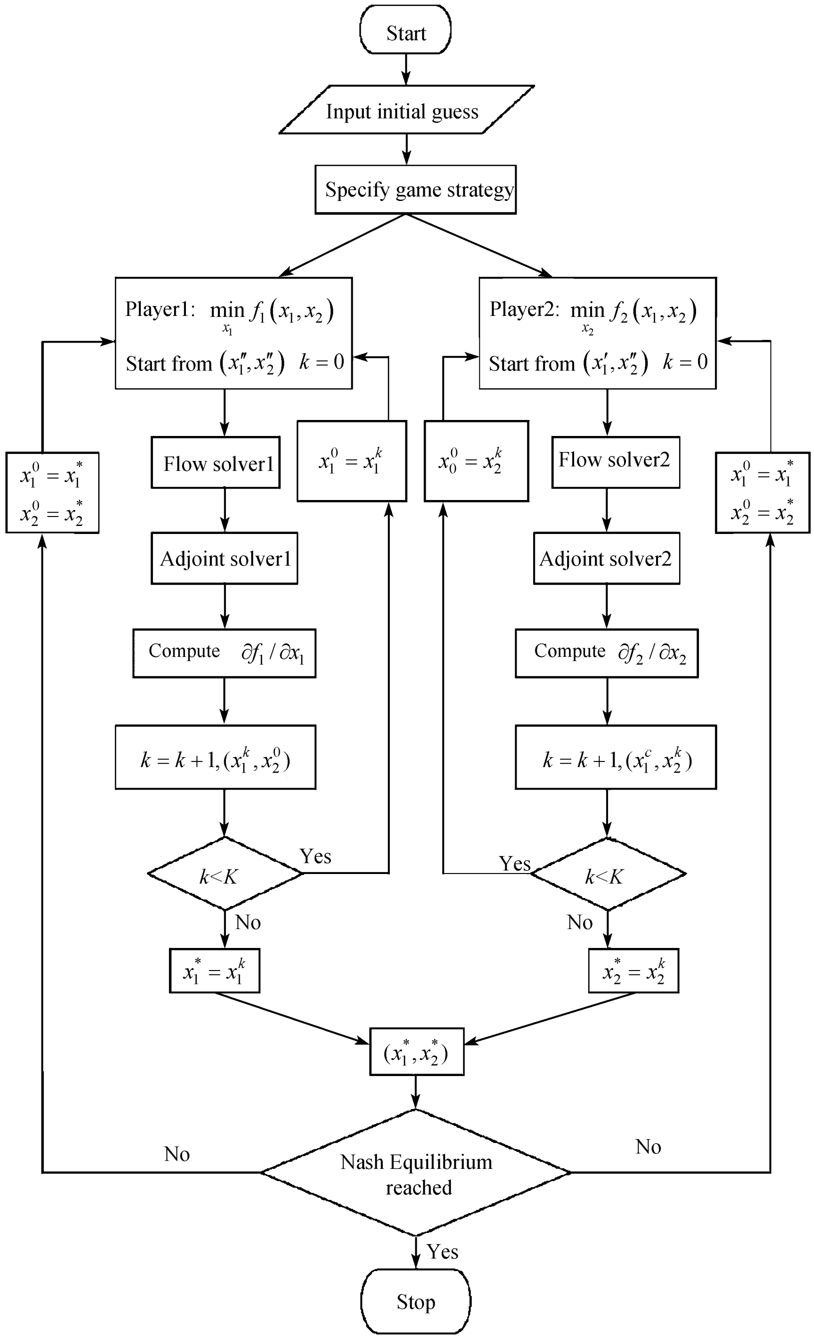

Figure 2 Flowchart of numerical procedure for Nash/Adjoint strategies in distributed computing environment.

Load each player's optimization task on an independent processor, and run at its own design point. For player *i*, adjust X_i to minimize J_i , $i = 1, 2$, simultaneously.

Do loop for adjoint optimization iterations, from $k = 1, 2, ..., K$.

Step 1. Run the CFD solver, outputs being C_l , C_d and the flow field variables.

Step 2. Run the adjoint solver.

Step 3. Compute the global gradient.

Step 4. Project the global gradient onto the corresponding subspace.

Step 5. Modify the partial design variables to get a tentative updated aerodynamic shape.

End of Adjoint iterations.

Get new partial design variables X_i^{new}.

Step 6. Exchange information symmetrically, and construct new global design variables $X^{\text{new}} = X_1^{\text{new}} \otimes X_2^{\text{new}}$, then update the global aerodynamic shape.

End of Nash cycles

Step 7. If the Nash equilibrium is reached, then stop; otherwise, $X^{\text{old}} = X^{\text{new}}$, and go to the beginning of Nash cycle.

Technical details of the above procedure (discrete state equations, discrete adjoint equations and discrete computation of the gradient can be found in refs. [21, 27]. The flowchart of this procedure with two players is shown in Figure 2.

5 Optimization examples and results

In accordance with the above design optimization methods described in sections 2, 3 and 4, the following experiments were performed to confirm the validity and efficiency of the methods. All the flow fields were analyzed on a 128×38 mesh with 4864 cells, which has an *O*-topology illustrated in Figure 1(b). The surface of airfoil was parameterized using 28 Hicks-Henne bell-shaped functions, 14 of them for the upper surface and 14 for lower surface (see details in ref. [21]).

5.1 Single-objective airfoil reconstruction in transonic regime via a virtual game strategy

Here, we introduce the concept of virtual game to treat a single point inverse design problem, and regard the single-objective design problem as a special case of multi-objective optimization, where the multiple cost functions are exactly the same. The cost function for an airfoil reconstruction design is defined in eqs. (12) and (13) via Virtual Games with 2 and 3 players. The low drag airfoil in transonic regime (defined in ref. [3]) is chosen as the target. The design conditions are $M_\infty = 0.75$, $\alpha = 1°$. Starting with the initial geometry of a NACA0012 airfoil, the shape evolves with iterations to reconstruct the target pressure distribution. Figures 3 and 4 show the splitting and optimization strategies of virtual games with 2 and 3 players, respectively. Each player optimizes the same cost function but only modifies the partial portion of shape as follows:

$$\begin{cases} \text{Player1}: \min\limits_{b^u_{i\in[1,6]},\, b^l_{i\in[1,6]}} \; J = \int_{\Gamma_c} (p - p_d)^2 \,\mathrm{d}s \quad \text{at } M_\infty = 0.75,\ \alpha = 1°, \\ \text{Player2}: \min\limits_{b^u_{i\in[7,14]},\, b^l_{i\in[7,14]}} \; J = \int_{\Gamma_c} (p - p_d)^2 \,\mathrm{d}s \quad \text{at } M_\infty = 0.75,\ \alpha = 1°, \end{cases} \tag{12}$$

$$\begin{cases} \text{Player1}: \min\limits_{b^u_{i\in[1,3]},\, b^l_{i\in[1,3]}} \; J = \int_{\Gamma_c} (p - p_d)^2 \,\mathrm{d}s \quad \text{at } M_\infty = 0.75,\ \alpha = 1°, \\ \text{Player2}: \min\limits_{b^u_{i\in[4,10]},\, b^l_{i\in[4,10]}} \; J = \int_{\Gamma_c} (p - p_d)^2 \,\mathrm{d}s \quad \text{at } M_\infty = 0.75,\ \alpha = 1°, \\ \text{Player3}: \min\limits_{b^u_{i\in[11,14]},\, b^l_{i\in[11,14]}} \; J = \int_{\Gamma_c} (p - p_d)^2 \,\mathrm{d}s \quad \text{at } M_\infty = 0.75,\ \alpha = 1°, \end{cases} \tag{13}$$

where b_i^u , b_i^l , $i=1,...,14$ are design variables of upper and lower surfaces of airfoil respectively. The resulting designed airfoils of the two games are in good agreement with target (see Figure 5), but the efficiency is quite different: the calculation is more efficient with a larger number players (see Figure 6).

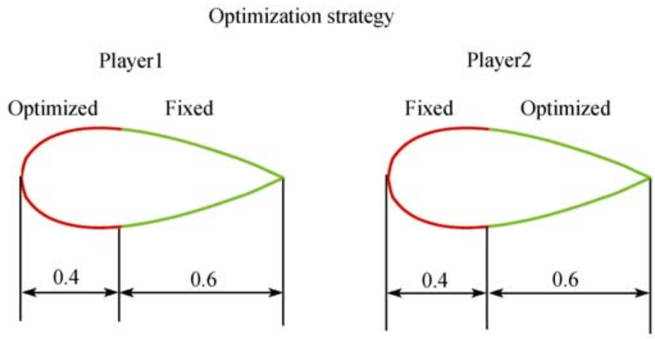

Figure 3 Splitting and optimization strategies (2 players). Exchange of information: every 10-10 design iterations.

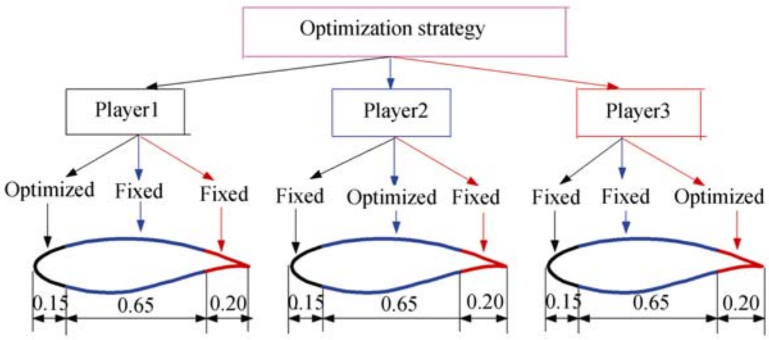

Figure 4 Splitting and optimization strategies (3 players). Exchange of information: every 5-5-5 design iterations.

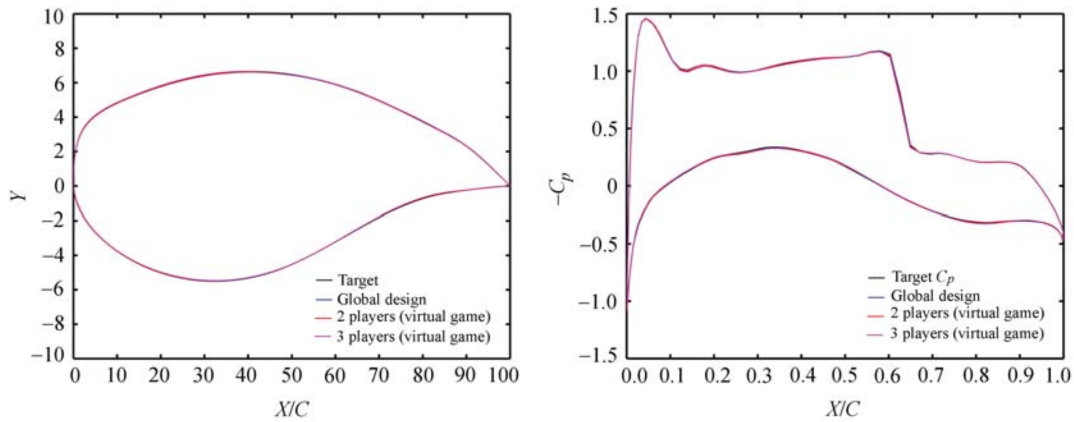

Figure 5 Comparison of designed results with target.

5.2 Single point airfoil drag reduction via virtual game strategy

Here, the concept of virtual game strategy is introduced to deal with the single point optimization problem defined as

$$\begin{cases} \min J = \dfrac{\Omega_1}{2} \oint\limits_c \left(p - p_{\text{initial}} \right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad\qquad C_l = C_l^{\text{initial}}, \end{cases} \tag{14}$$

where the weighting functions $\Omega_1 = 0.1$, and $\Omega_2 = 2.9$. The initial profile is *RAE*2822 airfoil, p_{initial} is the pressure distribution on initial airfoil. Design conditions are $M_\infty = 0.73$, $\alpha = 2.0°$. Three design strategies and their results are presented as follows.

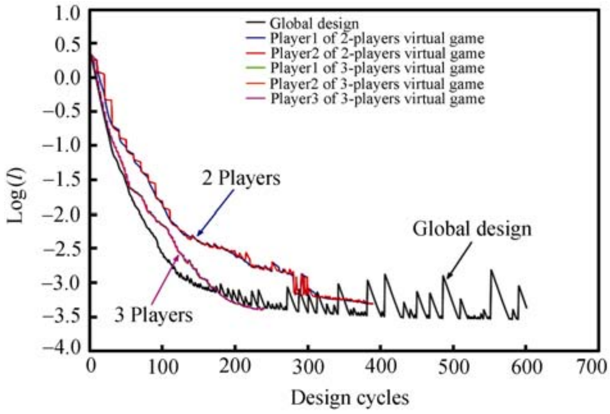

Figure 6 Convergence histories.

5.2.1 Results of front/rear splitting with two players. The splitting and optimization strategies in this experiment are illustrated in Figure 7. In mathematics, we define

$$\begin{cases} \text{Player1:} \begin{cases} \min\limits_{\Gamma_c^{\text{blue}}} J\left(\Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2} \oint\limits_c \left(p - p_{\text{initial}} \right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad\qquad C_l = C_l^{\text{initial}}, \end{cases} \\ \text{Player2:} \begin{cases} \min\limits_{\Gamma_c^{\text{red}}} J\left(\Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2} \oint\limits_c \left(p - p_{\text{initial}} \right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad\qquad C_l = C_l^{\text{initial}}, \end{cases} \end{cases} \tag{15}$$

where Player1 and Player2 optimize the same cost function (defined in (15)), but modify the different portion of design variables. Design variables are split into two sets because there are two players in the game. The first set is responsible for the modification of front portion (20% of chord length from leading edge) of the airfoil by Player1, the second set is responsible for the modification of rear

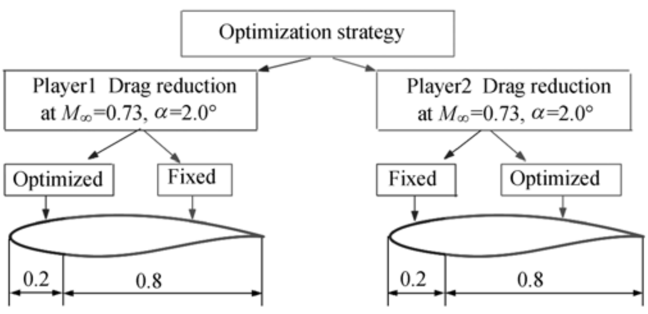

Figure 7 Splitting and optimization strategies. Exchange of information: every 3-3 design iterations.

portion (80% of chord length from the trailing edge) of the airfoil by Player2. Each player makes his decision on a separate processor simultaneously, meanwhile keeps the opponent's design variables unchanged, then exchanges the information every 3-3 design iterations. A Nash equilibrium is reached after 80 Nash cycles, see Figure 8(b). Figure 8(a) shows the results of this design optimization. The numerical results show significant improvements in C_d, the resulting C_d decreased by 67.4% from 0.0092 to 0.0030 after 240 design cycles, while the lift slightly increased by 3.33%.

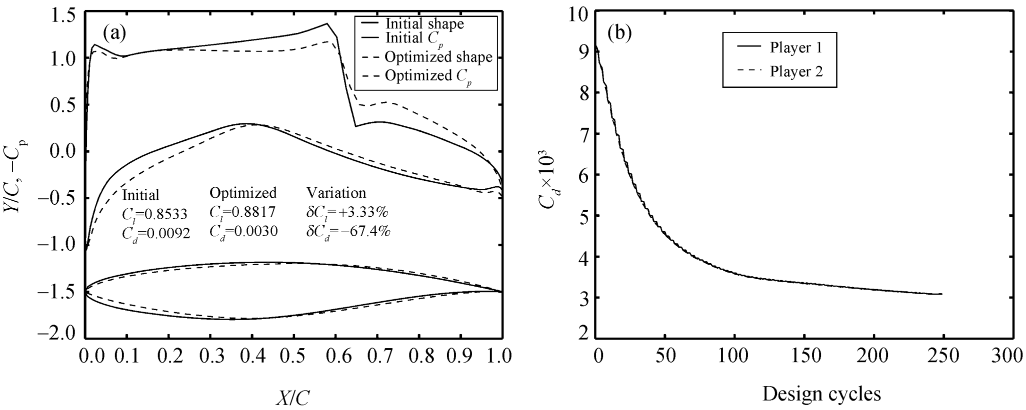

Figure 8 (a) Comparison of final results with the initial ones; (b) Nash Equilibrium establishment procedure.

5.2.2 Results of front/middle/rear splitting with three players. The splitting and optimization strategies of this experiment is illustrated in Figure 9. Mathematically, we define

$$\begin{cases} \text{Player1}: \begin{cases} \min\limits_{\Gamma_c^{\text{black}}} J\left(\Gamma_c^{\text{black}}, \Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2}\oint\limits_c \left(p - p_{\text{initial}}\right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad C_l = C_l^{\text{initial}}, \end{cases} \\ \text{Player2}: \begin{cases} \min\limits_{\Gamma_c^{\text{blue}}} J\left(\Gamma_c^{\text{black}}, \Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2}\oint\limits_c \left(p - p_{\text{initial}}\right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad C_l = C_l^{\text{initial}}, \end{cases} \\ \text{Player3}: \begin{cases} \min\limits_{\Gamma_c^{\text{red}}} J\left(\Gamma_c^{\text{black}}, \Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2}\oint\limits_c \left(p - p_{\text{initial}}\right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad C_l = C_l^{\text{initial}}. \end{cases} \end{cases} \tag{16}$$

There are three players in the game, so the design variables are split into three sets. The three players modify different portions of the airfoil (see Figure 9). The frequency of information exchange is every 4-4-1 design cycles. Figure 10 shows the results of design and Nash equilibrium establishment procedure. The numerical results show significant improvements in C_d, the resulting C_d decreased by 66.3% from 0.0092 to 0.0031 after 31 Nash cycles, while the lift slightly increased by 3.16%.

5.2.3 Results of alternation splitting with two players. In this case, we use the alternation splitting illustrated in Figure 11. The mathematical definition of optimization strategy is the same as in eq. (15), but Γ_c^{blue} and Γ_c^{red} are corresponding to the portions marked in Figure 11. Player1 works with the design variables with odd number and Player2 works with the even numbered

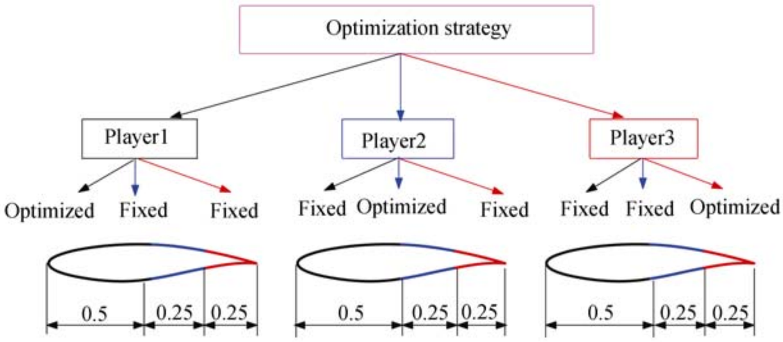

Figure 9 Splitting and optimization strategies. Exchange of information: every 4-4-1design iterations.

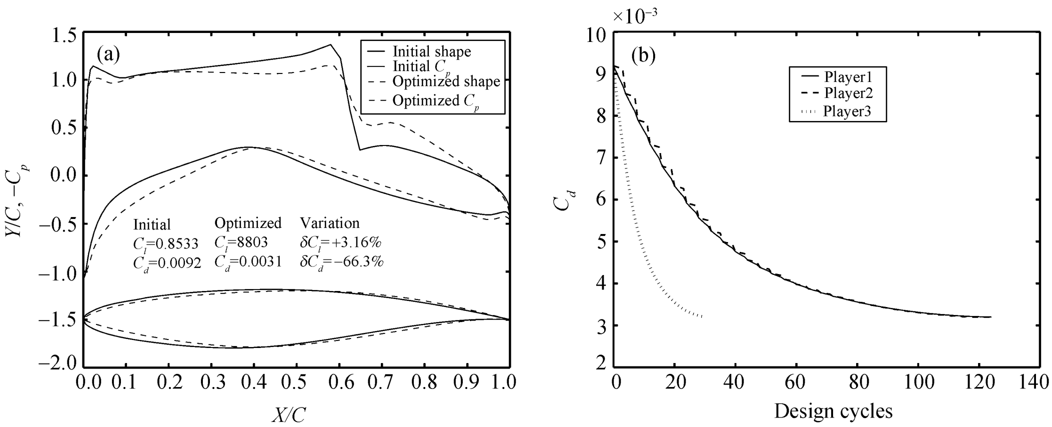

Figure 10 (a) Comparison of final results with the initial ones; (b) Nash equilibrium establishment procedure.

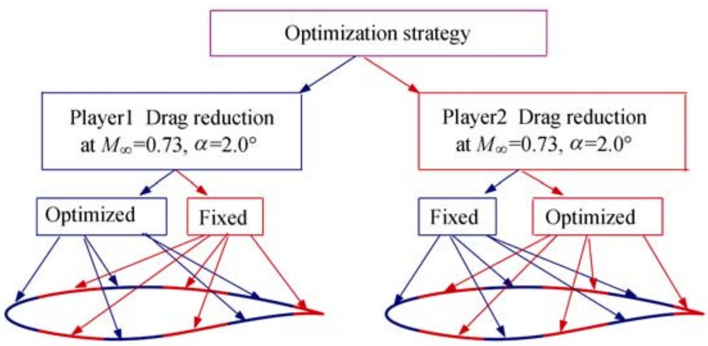

Figure 11 Splitting and optimization strategies. Exchange of information: every 3-3 design iterations.

design variables. Exchange of information is at every 3-3 design iterations. Figure 12 shows the results and Nash equilibrium establishment procedure. The numerical results also show remarkable improvements in C_d, the resulting C_d decreased by 66.3% from 0.0092 to 0.0031 after 82 Nash cycles, while the lift slightly increased by 3.81%.

If the number of players is increased, e.g. 4 or 5, we can find that the optimization is more efficient with the increase of the sub-tasks by splitting the design task into multiple sub-tasks and dispatching each sub-task to a separate processor (see Figure 13). This is an important merit of the Virtual Game strategy in distributed computing environment for single-point optimization problem.

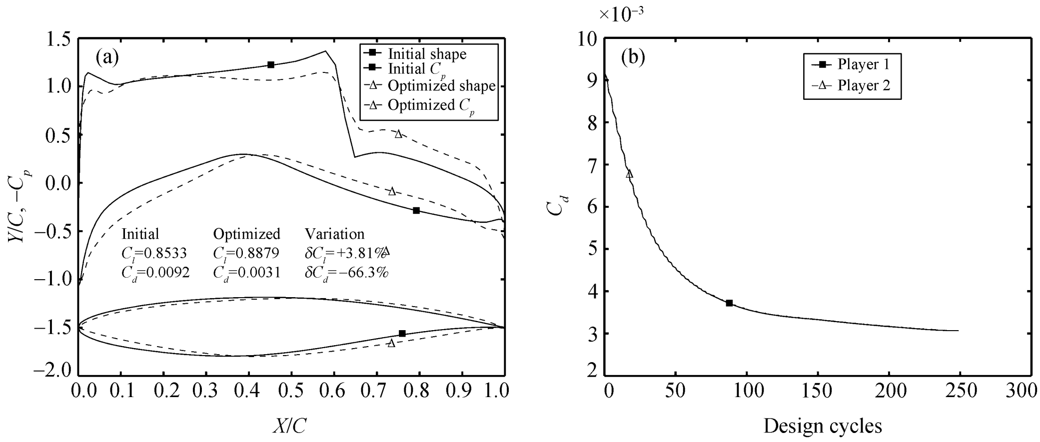

Figure 12 (a) Comparison of final results with the initial ones; (b) Nash Equilibrium establishment procedure.

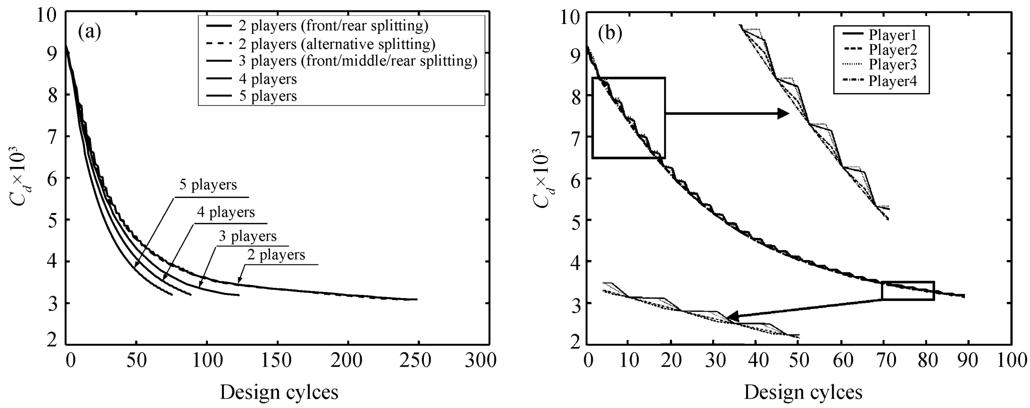

Figure 13 (a) Comparison of drag histories computed by different numbers of players with different splitting; (b) convergence behavior of each player in the virtual Nash game strategies with 4 players.

5.3 Two-point airfoil drag reduction via real game strategy

We attempt to reduce the drag of an airfoil at two transonic points. This multi-objective optimization problem is defined as follows:

$$\begin{cases} \left.\begin{array}{ll} \min J_1 = \dfrac{\Omega_1}{2}\oint_c (p - p_{d1})^2\,\mathrm{d}s + \Omega_2 C_{d1} & \\ \text{subject to} \qquad C_l = C_{l\ \text{initial}\,1} & \end{array}\right\} \text{at } M_\infty = 0.70,\ \alpha = 5.0°, \\ \left.\begin{array}{ll} \min J_2 = \dfrac{\Omega_1}{2}\oint_c (p - p_{d2})^2\,\mathrm{d}s + \Omega_2 C_{d2} & \\ \text{subject to} \qquad C_l = C_{l\ \text{initial}\,2} & \end{array}\right\} \text{at } M_\infty = 0.75,\ \alpha = 2.0°, \end{cases} \tag{17}$$

where the initial airfoil is still *RAE*2822, and the two weighting functions still are $\Omega_1 = 0.1$, and $\Omega_2 = 2.9$. Firstly, we evaluate a transonic Euler flow at $M_\infty = 0.70$ and $\alpha = 5.0°$ on *RAE*2822, and calculate the pressure p_{d1}; secondly, we evaluate the other transonic Euler flow at $M_\infty =$ 0.75 and $\alpha = 2.0°$, and calculate the pressure p_{d2}. We try to modify the *RAE*2822 airfoil to reduce both drags simultaneously under fixed lift mode in the two design conditions. Two optimization strategies and their results are presented as follows.

5.3.1 Results of front/rear splitting with two players. In this case, Player1 and Player2 minimize simultaneously the airfoil drag at the two independent transonic points. Each player works at one design point by modifying the different portion of design variables illustrated in Figure 14. Mathematically, we define

$$\begin{cases} \begin{array}{c} \text{Player1}: \\ M_\infty = 0.70 \\ \alpha = 5.0^\circ \end{array} \begin{cases} \min\limits_{\Gamma_c^{\text{blue}}} J\left(\Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2} \oint\limits_c \left(p - p_{d1}\right)^2 \mathrm{d}s + \Omega_2 C_{d1}, \\ \text{subject to} \qquad C_l = C_{l\ \text{initial}\,1}, \end{cases} \\ \begin{array}{c} \text{Player2}: \\ M_\infty = 0.75 \\ \alpha = 2.0^\circ \end{array} \begin{cases} \min\limits_{\Gamma_c^{\text{red}}} J\left(\Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2} \oint\limits_c \left(p - p_{d2}\right)^2 \mathrm{d}s + \Omega_2 C_{d2}, \\ \text{subject to} \qquad C_l = C_{l\ \text{initial}\,2}. \end{cases} \end{cases} \tag{18}$$

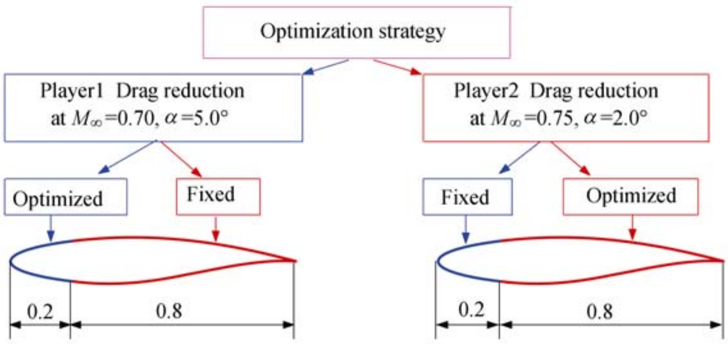

Figure 14 Splitting and optimization strategies. Exchange of information: every 2-2 design iterations.

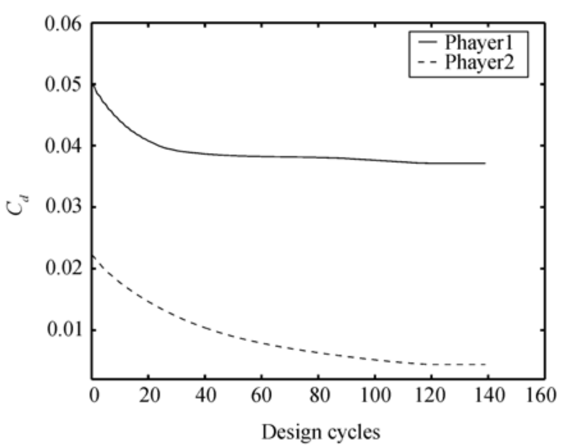

Figure 15 Convergence history of the design procedure and the Nash Equilibrium establishment.

The frequency of information exchange is every 2-2 design iterations. Nash equilibrium is reached when no player can improve his outcome by changing only his own design variables (see Figure 15). Figure 16 shows the results of this case, which indicate that Euler optimization procedure produces the weaker shock at both design points. The numerical results show the significant improvement in C_d at each design point. The resulting C_d of Player1 is decreased by 26.0% from 0.0501 to 0.0371 after 140 design cycles while the lift is slightly decreased by 7.51%; C_d of Player2 decreased by 80.3% from 0.0223 to 0.0044 while the lift slightly increased by 1.03%.

5.3.2 Results of alternation splitting with two players. Different from the above design case, the alternation splitting is used in this experiment (see Figure 17). The mathematical definition of optimization strategy is the same as in eq. (18), but Γ_c^{blue} and Γ_c^{red} correspond to the portions marked in Figure 17. That is to say, Player1 works with the odd numbered design variables, Player2

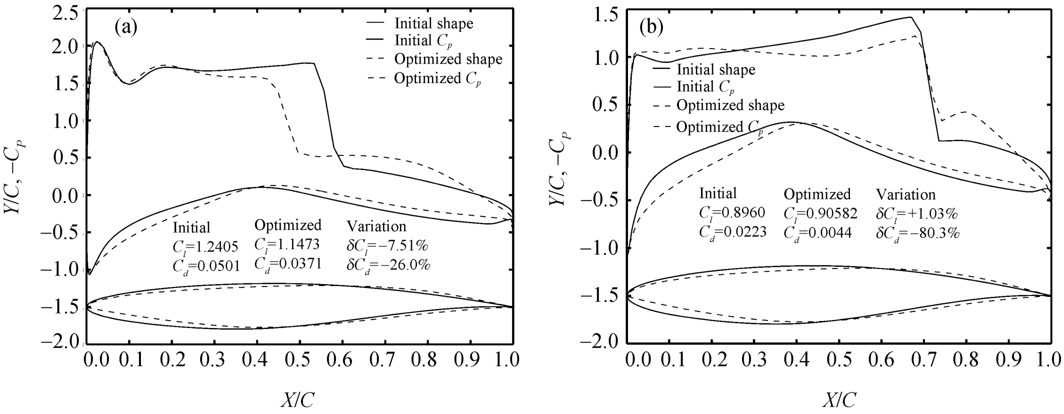

Figure 16 Comparison of final results with the initial ones at two design points.

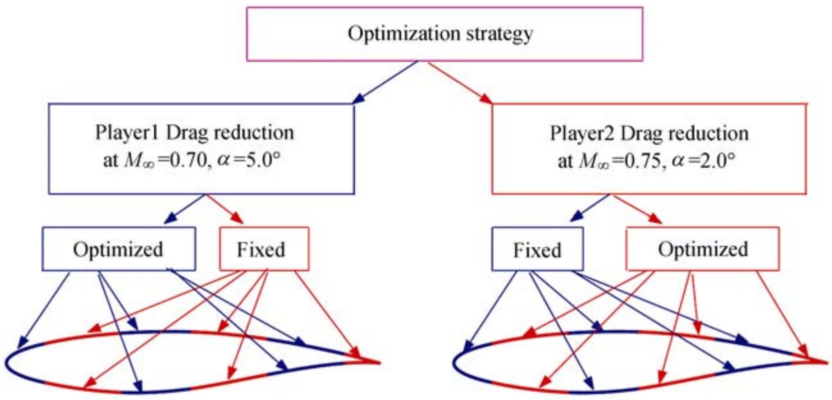

Figure 17 Splitting and optimization strategies. Exchange of information: every 2-2 design iterations.

works with the even numbered design variables. The frequency of information exchange is still every 2-2 design iterations. Figure 18 gives the procedure of Nash equilibrium establishment. Figure 19 shows the results of this optimization. Numerical results show significant improvements in C_d at each design point. The resulting C_d at first design point is decreased by 32.9% from 0.0501 to 0.0336 by Player1 after 140 design cycles while the lift slightly is decreased by 4.33%, and the resulting C_d at second design point decreased by 65.6% from 0.0223 to 0.0077 by Player2 while the lift slightly decreased by 4.07%. The numerical results in this section is different from that in the above subsection. This is due to the fact that different Nash equilibrium can be reached by different territory splitting in real game strategy.

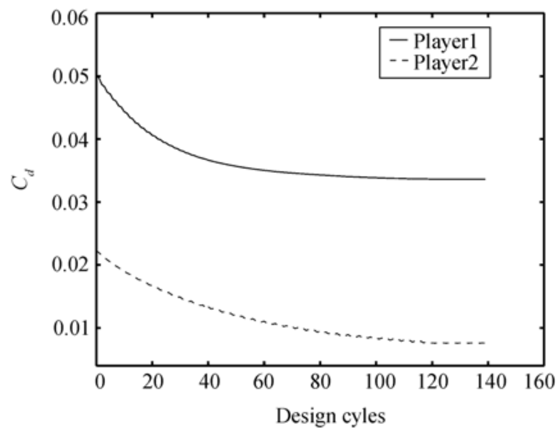

Figure 18 Convergence history of the design procedure and the Nash Equilibrium establishment.

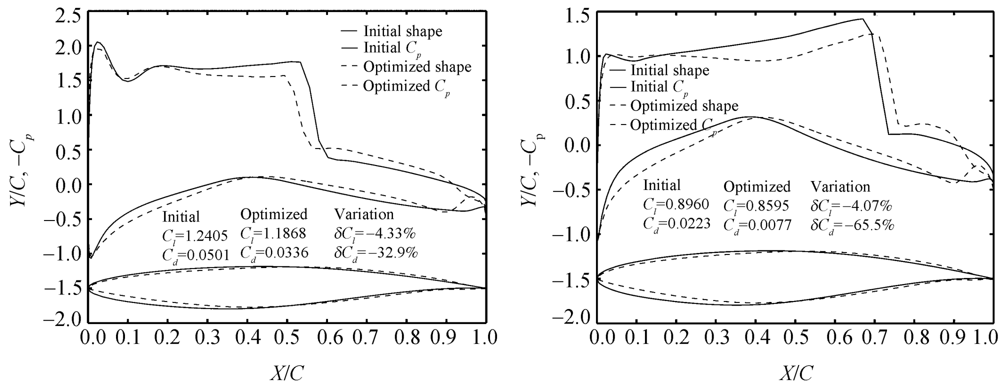

Figure 19 Comparison of final results with the initial ones at two design points.

5.4 Two-point airfoil lift maximization/drag minimization design via real game strategy

In this section, we will treat the confronted multi-objective optimum design problem to show further the flexibility and efficiency of the game theory, where the traditional multi-objective optimization methods exhibit their inherent shortcoming because it is difficult to organize the design problem under conflict. The test case is defined as follows:

$$\begin{cases} \max J_1 = C_l & \text{at } M_\infty = 0.35,\ \alpha = 8.0°, \\ \left.\begin{aligned} &\min J_2 = \frac{\Omega_1}{2}\oint_c \left(p - p_d\right)^2 \mathrm{d}s + \Omega_2 C_d \\ &\text{subject to} \qquad C_l = C_{l\ \text{initial}} \end{aligned}\right\} & \text{at } M_\infty = 0.75,\ \alpha = 2.0°, \end{cases} \tag{19}$$

where $\Omega_1 = 0.1$ and $\Omega_2 = 2.9$, the initial airfoil is still *RAE*2822. p_d is the pressure distribution on *RAE*2822 at $M_\infty = 0.75$ and $\alpha = 2.0°$. Design aims at making a proper modification on initial shape and achieving a better lift performance in subsonic flow, with shock wave drag in transonic flow as low as possible.

The splitting and optimization strategies are illustrated in Figure 20. The mathematical definition of the strategy is

$$\begin{cases} \text{Player1}: \max\limits_{\Gamma_c^{\text{blue}}} J_1\left(\Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = C_l \quad \text{at } M_\infty = 0.35, \alpha = 8.0°, \\ \begin{matrix}\text{Player2}: \\ M_\infty = 0.75 \\ \alpha = 2.0°\end{matrix} \begin{cases} \min\limits_{\Gamma_c^{\text{red}}} J_2\left(\Gamma_c^{\text{blue}}, \Gamma_c^{\text{red}}\right) = \dfrac{\Omega_1}{2}\oint_c \left(p - p_d\right)^2 \mathrm{d}s + \Omega_2 C_d, \\ \text{subject to} \qquad C_l = C_{l\ \text{initial}}. \end{cases} \end{cases} \tag{20}$$

Player1 and Player2 maximize lift/minimize drag of an airfoil simultaneously at the two different design points by modifying the separate portions of airfoil. Each player works on a separate processor independently. After 3 design iterations, players exchange their latest design variables to construct a new airfoil. Repeat this process until no player can improve his outcome by changing only his own design variables. The numerical results show remarkable improvement in C_l at subsonic design point and significant improvement in C_d at transonic design point. Because the resulting C_l of Player1 is increased by 6.38% from 1.1436 to 1.2166 after 80 design iterations, meanwhile, the resulting C_d of Player2 is decreased by 34.5% from 0.0223 to 0.0146 (see Figure 21). Figure 22 gives the histories of lift and drag at each design point, which indicate that the Nash

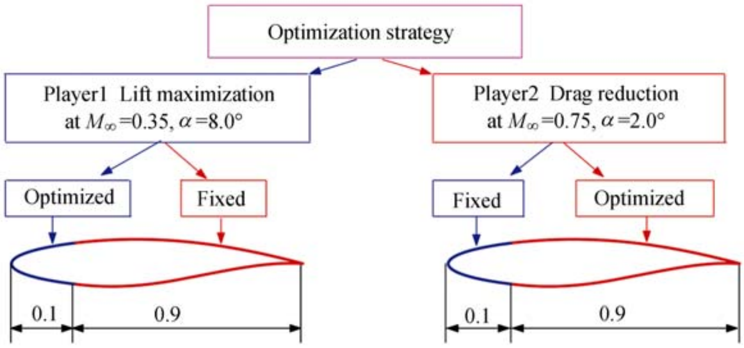

Figure 20 Splitting and optimization strategies. Exchange of information: every 3-3 design iterations.

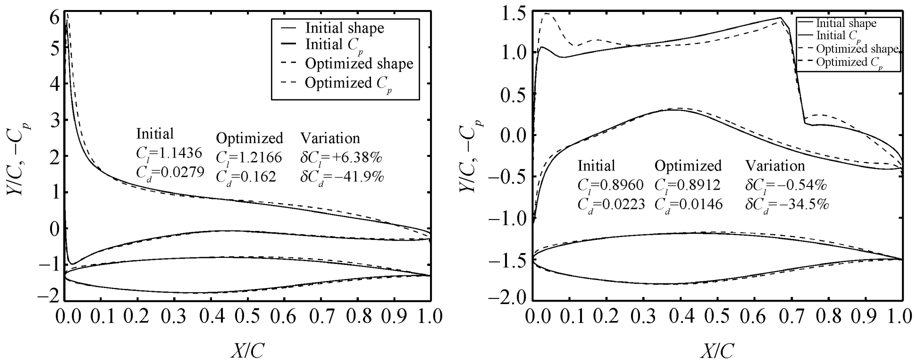

Figure 21 Comparison of final results with the initial ones for two design points.

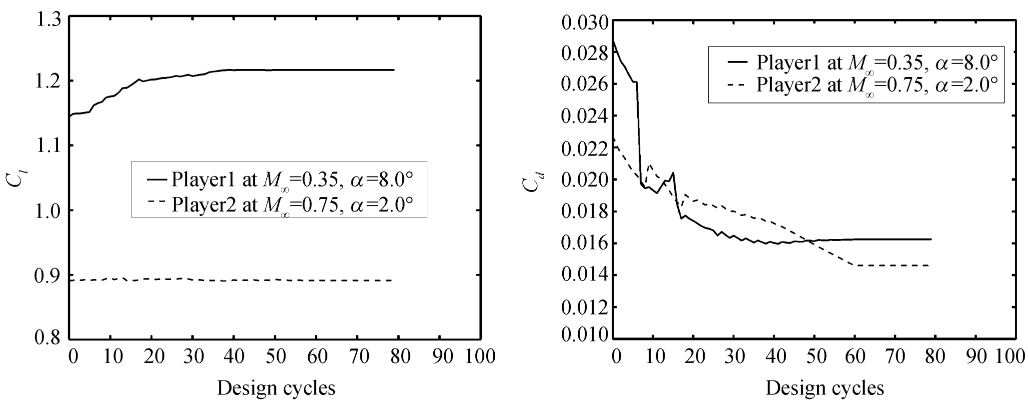

Figure 22 Convergence history of the optimization procedure.

Equilibrium is reached because all of them do not change any more information as the number of design iteration increases.

6 Conclusion

In this paper, a novel approach described in ref. [21] is used successfully in more practical aerodynamic optimization to treat multi-point design problems. A Nash Equilibrium is reached when each player, constrained by the strategies of the others, cannot improve further his own target. Specific virtual and real symmetric Nash games are introduced and implemented to set up an optimization strategy for design under conflict.

For the single point design, the optimization process can be accelerated by splitting the design task into multiple sub-tasks and distributing each subtask to a separate processor. Moreover, the calculation is more efficient with a larger number of sub-tasks. This is an important merit of the game approach in parallel computing.

1 Cohon J L. Multi-objective Programming and Planning. New York: Academic Press, 1978
2 Deb K. Multi Objective Optimization Using Evolutionary Algorithms. New York: John Wiley, 2001
3 Marco N, Desideri J A, Lanteri S. Multi-objective optimization in CFD by genetic algorithms. INRIA Report No. 3686, 1999
4 Tang Z L, Desideri J A, Periaux J. Distributed optimization using virtual and real game strategies for aerodynamic multi-objective design. In: Zeiton V E, Periaux J, Marini M, eds. WEHSFF Aerospace Applications from High Subsonic to Hypersonic Regime, Barcelona: CIMNE, 2002
5 Nash J F. Non-cooperative games. Ann Math, 1951, 54(2): 286－295
6 Basar T, Olsder G J. Dynamic Non-cooperative Game Theory. Bodmin, Cornwall: Academic Press, 1995
7 Osborne M J, Rubinstein A. A Course in Game Theory. Massachusetts Institute of Technology, 1997
8 Ehtanzo H. Incentive strategies and equilibria for dynamic games with delayed information. J Optim Theor Appl, 1989, 63(3): 355－370[DOI]
9 Ehtanzo H. A cooperative incentive equilibrium for a resource management problem. J Economic Dyn Control, 1993, 17(4): 659－678[DOI]
10 Wilson B, Cappelleri D J, Simpson T W, et al. Efficient Pareto frontier exploration using surrogate approximations. Optim Eng, 2001, 2: 31－50[DOI]
11 Sobol I M. An efficient approach to multi-criterion optimization design problems. Surv Math Indus, 1992, .1: 259－ 281
12 Athan T W, Papalambros P Y. A note on weighted criteria methods for compromise solutions in multi-objective optimization. Eng Optim, 1996, 27(2): 155－176[DOI]
13 Messac A, Sundararaj J G, Tappeta R V, et al. The ability of objective functions to generate non-convex Pareto frontier. In: 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structure Dynamics, and Material Conference and Exhibition, AIAA, 1999, 1: 78－87
14 Das I, Dennis J E. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multi-criterion optimization problems. Struc Optim, 1997, 14(1): 63－69[DOI]
15 Das I, Dennis J E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multi-criterion optimization problems. SIAM J on Optim, 1998, 8(3): 631－657[DOI]
16 Periaux J, Sefrioui M, Stoufflet B, et al. Robust genetic algorithms for optimization problems in aerodynamic design. In: Galais M, Winter G, Periaux J, et al., eds. Genetic Algorithm in Engineering and Computer Science. New York: John Wiley, 1995. 371－396
17 Sefrioui M, Periaux J, Mantel B. RCS multi-objective optimization of scattered waves by active control elements using GA. In: Fourth International Conference on Control, Automation, Robotics and Vision, Singapore, 1996
18 Padovan L, Pediroda V, Poloni C. Multi objective robust design optimization of airfoils in transonic field (M.O.R.D.O). In: International Congress on Evolutionary Methods for Design Optimization and Control with Application to Industrial Problems, EUROGEN. Barcelona: CIMNE, 2003. 1－7
19 Gonzalez L F, Whitney E J, Srinivas K, et al. Multidisciplenary aircraft conceptual design optimization using a hierarchical asynchronous parallel evolutionary algorith (HAPEA). In: Proceedings of the Sixth International Conference on Adaptive Computing in Design and Manufacture (ACDM2004). Bristol: Springer Verlag, 2004. 1－9
20 Tang Z L. Multi-objective optimization strategies using adjoint method and game theory in aerodynamics. Acta Mech Sin, 2006, .22(4): 307－314[DOI]
21 Tang Z L, Desideri J A, Periaux J. Multi-criteria aerodynamic shape-design optimization and inverse problems using control theory and Nash Games. J Optim Theor Appl, 2007, 135(2): 599－622 [DOI]
22 Lions J L. Controle Optimal des Systemes Gouvernes par des Equations aux Derivees Partielles. Paris: Gauthier-Villars, 1969
23 Jameson A. Aerodynamic design via control theory. J Sci Comput, 1988, 3(3): 233－260 [DOI]
24 Reuther J, Jameson A. Control theory based airfoil design using the Euler equations. AIAA 94－4272, 1994
25 Pironneau O. Optimal shape design for elliptic systems. New York: Springer-Verlag, 1984
26 Tang Z L. Constrained optimal control theory: Application to aerodynamics design. Chin J Theor Appl Mech, 2007, 39(2): 273－277
27 Tang Z L. The research on optimum aerodynamic design using CFD and control theory. PhD Thesis. Nanjing: Nanjing University of Aeronautics & Astronautics, 2000