Efficient Hybrid-Game Strategies Coupled to Evolutionary Algorithms for Robust Multidisciplinary Design Optimization in Aerospace Engineering

D. S. Lee, L. F. Gonzalez, *Member, IEEE,* J. Périaux, and K. Srinivas

***Abstract*—A number of game strategies have been developed in past decades and used in the fields of economics, engineering, computer science, and biology due to their efficiency in solving design optimization problems. In addition, research in multiobjective and multidisciplinary design optimization has focused on developing a robust and efficient optimization method so it can produce a set of high quality solutions with less computational time. In this paper, two optimization techniques are considered; the first optimization method uses multifidelity hierarchical Pareto-optimality. The second optimization method uses the combination of game strategies Nash-equilibrium and Pareto-optimality. This paper shows how game strategies can be coupled to multiobjective evolutionary algorithms and robust design techniques to produce a set of high quality solutions. Numerical results obtained from both optimization methods are compared in terms of computational expense and model quality. The benefits of using Hybrid and non-Hybrid-Game strategies are demonstrated.**

***Index Terms*—Evolutionary optimization, game strategies, Nash-equilibrium, Pareto front, robust design, shape optimization, uncertainties.**

I. Introduction

THERE IS an increased complexity in optimizing aerospace designs due to advent of new technologies. Research in multiobjective (MO) and multidisciplinary design optimization (MDO) therefore faces the need for developing robust and efficient optimization methods and produce higher quality designs without paying expensive computational cost. The recent optimization problems use robust design techniques to produce high quality designs; however, this dramatically increases the computational expense [1]–[3]. One alternative method can be the use of game strategies to save the computational cost. Nash and Pareto strategies are game theory tools which can be used to save central processing unit (CPU) usage and to produce high quality solutions due to their efficiency in design optimization.

This paper considers the application of [4], where Lee *et al.* studied multiobjective and robust multidisciplinary design optimization of unmanned combat air vehicle (UCAV) using hierarchical asynchronous parallel multiobjective evolutionary algorithm (HAPMOEA) [5]. Numerical results from [4] show that the robust design technique produces high quality solutions which have higher aerodynamic performance with lower sensitivity when compared to the baseline design while avoiding the over-optimized solutions. However, it can be seen that the use of a robust design technique takes high computational cost. This paper therefore introduces a new optimization method coupled to an evolutionary algorithm to save the computational cost; the method is a dynamic combination of the Nash-equilibrium [6] and Pareto-optimality approaches [7] and is denoted by Hybrid-Game. HAPMOEA uses three hierarchical layers with seven populations (Pareto-Games) which are divided by multifidelity conditions. The Hybrid-Game consists of one Pareto-Player and several Nash-Players providing dynamic elite information to the Pareto algorithm and hence it can produce a Nash-equilibrium and Pareto nondominated solutions simultaneously [8]. It is shown in this paper how a Nash-Game acts as a pre-conditioner of the Pareto algorithm to speed up the capture of the Pareto front. This new approach is implemented successfully to solve complex robust MO/MDO problems which require expensive computational cost.

Numerical results obtained by both optimization methods for the detailed design of an unmanned aerial system (UAS) blended wing under uncertainties, are compared in terms of computational expense and quality in the design. The benefits of using game strategies coupled with evolutionary algorithms are clearly demonstrated and illustrate the potential of the method as a future tool to be used in an advanced industrial design environment.

The rest of this paper is organized as follows. Section II describes both methodologies and algorithms of HAPMOEA and Hybrid-Game. Mathematical design problems are conducted as validation test cases for Hybrid-Game coupled

Manuscript received July 1, 2009; revised October 12, 2009, December 11, 2009, and January 25, 2010. Date of current version March 30, 2011.

D. S. Lee and J. Périaux are with the International Center for Numerical Methods in Engineering (CIMNE)/UPC, Barcelona 08034, Spain (e-mail: ds.chris.lee@gmail.com; jperiaux@gmail.com).

L. F. Gonzalez is with the School of Engineering System, Queensland University of Technology, Brisbane QLD 4072, Australia (e-mail: felipe.gonzalez@qut.edu.au).

K. Srinivas is with the School of Aerospace Mechanical Mechatronic Engineering (AMME), University of Sydney, NSW 2006, Australia (e-mail: k.srinivas@usyd.edui.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2010.2043364

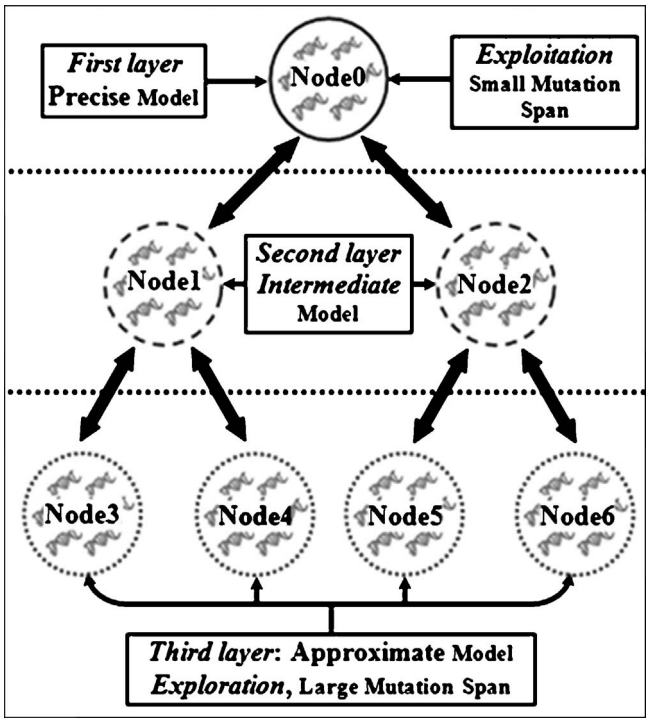


Fig. 1. Topology of HAPMOEA.

to HAPMOEA in Section III. Section IV describes analysis tools for aerodynamics and electromagnetics. The real-world design problems are conducted in Section V. A discussion and a conclusion are presented in Sections VI and VII.

II. Methodology

Both methods HAPMOEA and Hybrid-Game have the same features of multiobjective evolutionary algorithms (MOEA). HAPMOEA uses the hierarchical multipopulation Pareto-optimality approach while both concepts of Nash-equilibrium and Pareto-optimality are implemented for the Hybrid-Game. Both HAPMOEA and Hybrid-Game have capabilities of solving robust/uncertainty design problems.

A. Multiobjective Evolutionary Algorithms (MOEA)

Both HAPMOEA and Hybrid-Game optimization approaches use a MOEA with several analysis tools [5]. The core of stochastic method is based on evolution strategies [9], [10] which incorporate the concepts of covariance matrix adaptation [11], [12], distance dependent mutation [10], and the asynchronous parallel computation [13], [14]. The methods couple the MOEA, analysis tools, and a statistical design tool to evaluate uncertainty in the design.

B. Hierarchical Multifidelity/Population Topology

A hierarchical multifidelity/population topology [15] uses three layers (HAPMOEA-L3) as shown in Fig. 1. Reference [16] shows that the use of hierarchical multifidelity populations makes faster convergence when compared to single population EA.

The optimizer has capabilities to handle multifidelity/physics models for the solution. There are seven different populations in HAPMOEA-L3; the first layer (one high-fidelity population: Node0) concentrates on the refinement of solutions, while the third layer (four less-fidelity populations: Node3–Node6) uses approximate model. Therefore, the populations at the third layer are entirely devoted to exploration. The second layer (two intermediate-fidelity populations: Node1 and Node2) compromises solutions from between exploration (third layer) and exploitation (first layer). Details of hierarchical setting can be found in [18]; the less fidelity is the use of less resolution of mesh condition which produces less than 5% accuracy error.

As an example, if the problem considers six design variables (DV1–DV6) each Pareto-Game at each layer has the same fitness/objective function and considers whole design variable span (DV1–DV6). There is migration operation at every generation; individual migrates up and down from the third to the first layer and from the first to the third layer during the optimization. The topology of HAPMOEA is normally fixed for the multiobjective, multidisciplinary design. In addition, HAPMOEA uses the well-known concept of Pareto-optimality [7], [20]. Details of HAPMOEA can be found in [5].

C. Nash-Game

Nash-equilibrium is a result of a game based on symmetric information exchanged between different players. Each player is in charge of one objective, has its own strategy set, and its own criterion. During the game, each player looks for the best strategy in its search space in order to improve its own objective criterion while design variables from other players' criteria are fixed. In other words, Nash-Game will decompose a problem into several simpler problems corresponding to the number of Nash-Players. The Nash-equilibrium is reached after a series of strategies tried by players in a rational set until no players can improve its score/objective values by changing its own best strategy. For instance, if the problem considers the objective function as $f = \min(xy)$ as illustrated in Fig. 2.

The design variable x corresponds to the first criterion and y to the second one. The first player P_1 is assigned for the optimization of x and the optimization of y to P_2. P_1 optimizes $f = \min(xy*)$ with respect to the first criterion by modifying x, while $y*$ is fixed by P_2. Symmetrically, P_2 optimizes $f = \min(x*y)$ with respect to the second criterion by modifying y while $x*$ is fixed by P_1. The Nash-equilibrium will be reached when both players P_1 and P_2 cannot improve their objective functions $f = \min(xy*)$ and $f = \min(x*y)$, respectively, i.e., $f = \min(x*y*) = f = \min(x*y)$ and $f = \min(x*y)$. It can be seen that the Nash-Game decomposes a problem $[f = \min(xy)]$ into two simpler problems, in this case two Nash-Players; P_1 $[f = \min(x*y)]$ and P_2 $[f = \min(xy*)]$ to create a competitive design environment for Nash-Game.

In this paper, Nash-Game is used to decompose complex design problems and also to be performed as a dynamic pre-conditioner incorporated to Pareto-optimality. These characteristics of Nash-Game will accelerate the multiobjective optimization process by capturing local minima.

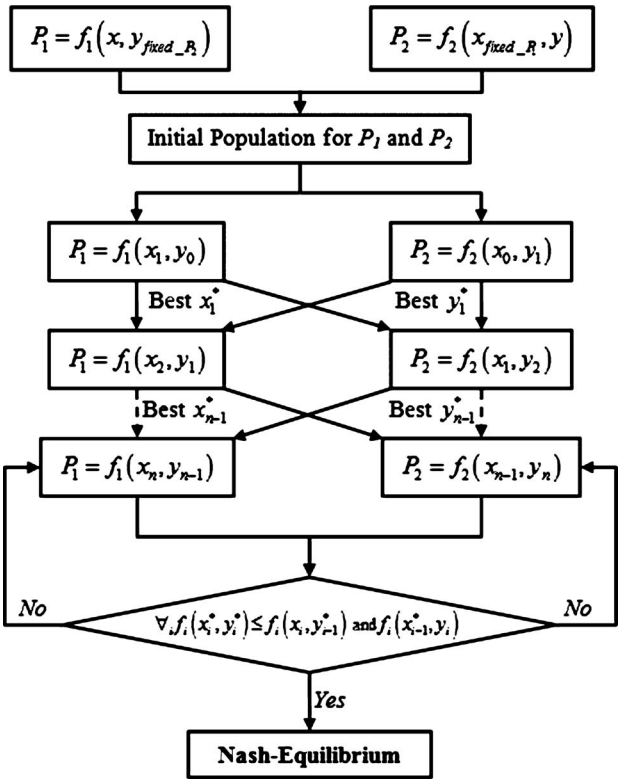


Fig. 2. Nash-Game.

D. Hybrid-Game (Hybrid-Nash)

The Hybrid-Game uses the dynamic concepts of Nash-Game and Pareto-optimality and hence it can simultaneously produce Nash-equilibrium and a set of Pareto nondominated solutions [8]. The reason for implementing of Nash-Game is to speed up in order to search one of the global solutions. The global solution or elite design from Nash-Game will be seeded to a Pareto-Game at every generation. This mechanism increases diversity of Pareto-Game during the optimization process. Each Nash-Player has its design criteria using own optimization strategy. The example shape of hybrid Nash-HAPEA topology is a top view of trigonal pyramid as shown in Fig. 3.

It can be seen that the optimizer consists of three Nash-Players with one Pareto-Player in the middle. Each Nash-Player is located in a symmetrical array at 60° (line 1, line 2, and line 3). Each Nash-Player can have a single or two hierarchical sub-players. As an example, if the problem considers six design variables (DV1–DV6). The distributions of design variable are: Nash-Player 1 (black circle) only considers black square design components (DV1, DV4), DV2 and DV5 are considered by Nash-Player 2 (blue circle) while Nash-Player 3 considers DV3 and DV6. The Pareto-Player considers whole design variable span (DV1 to DV6). It can be noticed that the sum of Nash-Players design variables is the same as the number of design variables for the Pareto-Player. This is because a set of elite designs (DV1–DV6) obtained by Nash-Game will be seeded to the population of Pareto-Player. In this example, Nash-Game decomposes the problem into three simpler problems corresponding to Nash-Player 1, Nash-Player 2, and Nash-Player 3 to become a pre-conditioner of Pareto-Player.

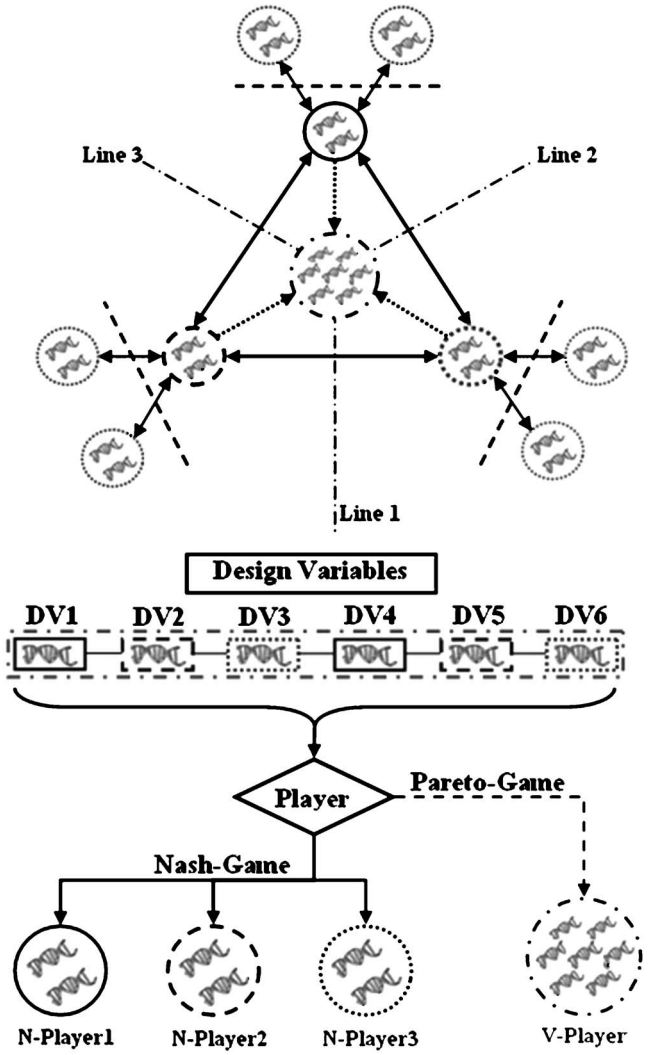


Fig. 3. Topology of Hybrid-Game.

The Nash-Game will decompose the problem into several single-objective design problems if the problem considers a multiobjective design. Also, Nash-Game will decompose the problem into single-disciplinary design problems if the problem considers the multidisciplinary/multiphysics design.

The topology of hybrid Nash-HAPEA is flexible; if there are four Nash-Players, then the shape will be a quadrangular pyramid.

E. Robust/Uncertainty Design

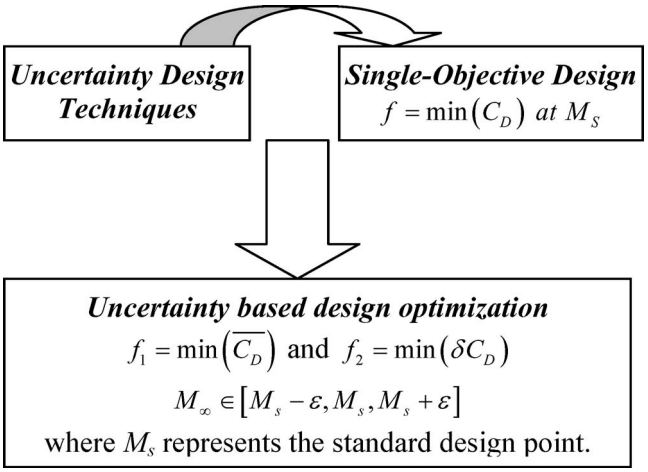
A robust design uncertainty technique developed by Taguchi [17] is considered to improve design quality of the physical model. The robust design approach is defined by using two statistical sampling formulas *mean* (1) and *variance* (2)

$$\overline{f} = \frac{1}{K}\sum_{j=1}^{K} f_j \tag{1}$$

$$\delta f = \frac{1}{K-1}\left(\sum_{j=1}^{K} \left| f_j - \overline{f} \right|\right) \tag{2}$$

where K represents the number of off-design conditions.

The values obtained by mean and variance represent the model quality in terms of the magnitude of performance and stability/sensitivity at a set of variable design conditions. For instance, when *uncertainty* is applied to single-objective problem such as minimization of drag [$f = \min(C_D)$], the problem can be modified as an uncertainty based multiobjective design problem as follows:



1) Apply K number of off-design conditions with the step size ϵ in operating condition M_∞; Mach number in standard flight condition (M_S) becomes a vector of flight conditions $M_{\infty_K} \in [M_S - \varepsilon, M_S, M_S + \varepsilon]$.
2) Split the objective/fitness function into *mean* [$\overline{C_D}$: (3)] and *variance* of drag coefficient [δC_D: (4)]

$$\overline{C_D} = \frac{1}{K}\sum_{i=1}^{K} C_{Di} \tag{3}$$

$$\delta C_D = \frac{1}{K-1}\left(\sum_{i=1}^{K} |C_{Di} - \overline{C_D}|\right) \tag{4}$$

where K represents the number of uncertainty conditions.

Consequently, the major role of uncertainty technique is to improve C_D quality with low drag coefficient and drag sensitivity in uncertain flight conditions by computing mean and variance of criteria. Additional details on the uncertainty based technique can be found in [1], [2], [4].

F. Algorithms for HAMOEA and Hybrid-Game

The algorithms for HAPMOEA and Hybrid-Game are shown in Fig. 4(a) and (b) where it is assumed that the problem considers the objective function $f = \min(x_1 x_2 x_3)$.

1) *HAPMOEA-L3 [Fig. 4(a)]:* The method has eight main steps as follows.

1) Step 1: Define population size and number of generation for hierarchical topology (Node0 to Node6), number of design variables (x_1, x_2, x_3) and their design bounds, model fidelity [Layer1 (Node0): precise, Layer2 (Node1, Node2): intermediate, Layer3 (Node3 to Node6): least precise].
2) Step 2: Initialize seven random populations on each Node0 to Node6.
   while termination condition (generation or elapsed time or pre-defined fitness value).
3) Step 3: Generate offspring using selection, mutation, or recombination operations.
4) Step 4: Evaluate offspring corresponding to fitness functions.
   a) Step 4-1: Evaluate offspring on each node using precise, compromise, least precise model.
5) Step 5: Sort each population for each node based on its fitness.
6) Step 6: Replace best individuals into the nondominated population of each node.
   end while (termination condition is reached)
7) Step 7: Analysis of final results; Pareto-optimal front obtained by Node0 at first layer (precise model).
8) Step 8: Conduct post-processing of results; if the problem considers aerodynamic wing design for instance, Mach sweep will be plotted for each objective (C_D, C_L, *L/D*).

2) *Hybrid-Game [Fig. 4(b)]:* The method has eight main steps as follows.

1) Step 1: Define population size and number of generation for Nash-Players (N-Player 1, N-Player 2, N-Player 3) and Pareto-Player (P-Player), number of design variables (x_1, x_2, x_3) and their design bounds. Splitting of the design variables for each player (N-Player 1: x_1, N-Player 2: x_2, N-Player 3: x_3, P-Player: x_1, x_2, x_3).
2) Step 2: Initialize random population for each player.
   while termination condition (generation or elapsed time or pre-defined fitness value).
3) Step 3: Generate offspring using selection, mutation, or recombination operations.
4) Step 4: Evaluate offspring in each Pareto and Nash-Player.
   a) Step 4-1: Evaluate offspring in Nash-Game.
      i) N-Player 1: use x_1 with design variables x_2, x_3 fixed by N-Player 2 and N-Player 3.
      ii) N-Player 2: use x_2 with design variables x_1, x_3 fixed by N-Player 1 and N-Player 3.
      iii) N-Player 3: use x_3 with design variables x_1, x_2 fixed by N-Player 1 and N-Player 2.
   b) Step 4-2: Evaluate offspring in P-Player.
      if (the first offspring at each generation is considered)
      i) P-Player: seed elite design (x_1*, x_2*, x_3*) obtained by each Nash-Player in Step4-1.
      else
      ii) P-Player: use x_1, x_2, x_3 obtained by mutation or recombination operation as default.
5) Step 5: Sort each population for each player based on its fitness.
6) Step 6: Replace the nondominated individuals into each player population.
   end while
7) Step 7: Analysis of final results.
   a) P-Player: Pareto-optimal front obtained by Pareto-Player.

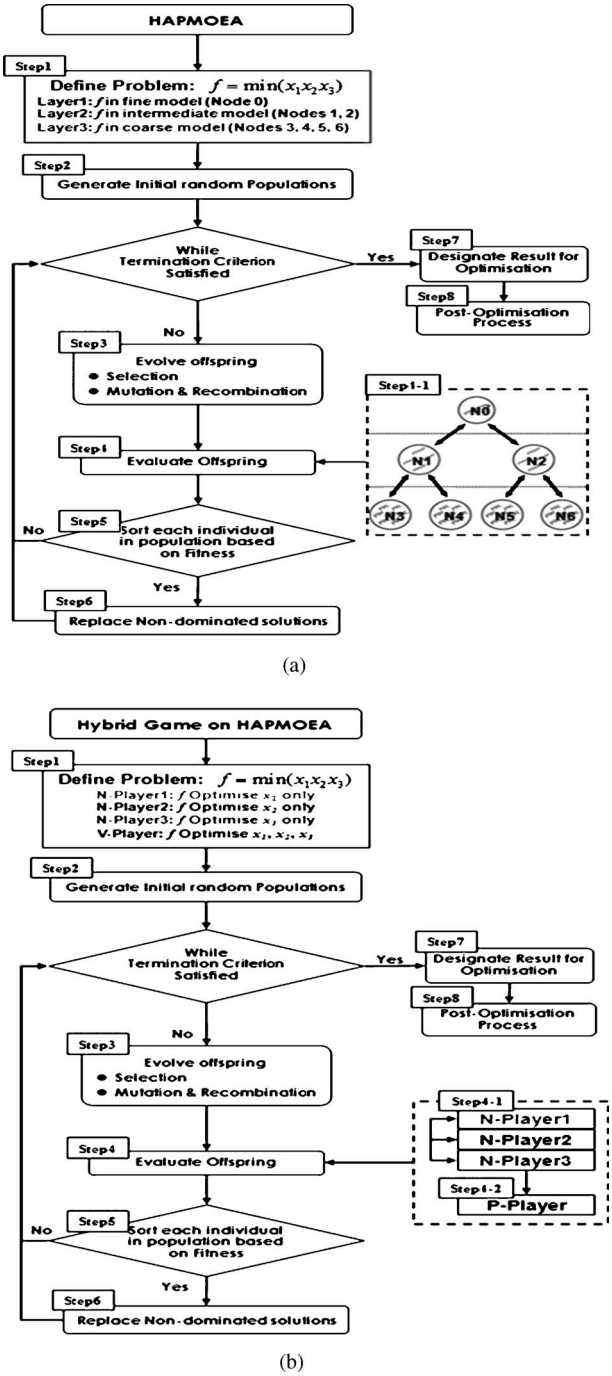


Fig. 4. (a) Algorithm of HAPMOEA-L3. (b) Algorithm of Hybrid-Game.

b) Nash-Game: Plot Nash-equilibrium obtained by N-Player 1, N-Player 2, N-Player 3.

8) Step 8: Conduct post-processing of results; if the problem considers aerodynamic wing design for instance, Mach sweep will be plotted for each objective (C_D, C_L, *L/D*).

III. Mathematical-Benchmark Validation of Hybrid-Game (Hybrid-Nash)

The HAPMOEA-L3 approach has been tested for a number of multiobjective test problems [18], [19]. In this section, the Hybrid-Game on the HAPMOEA approach described in the previous section is verified though three multiobjective mathematical test cases, including nonuniformly distributed nonconvex, discontinuous, and a nonlinear goal programming of mechanical design problem. In addition, the Pareto convergences obtained by NSGA-II and Hybrid-Game on NSGA-II are compared. For these mathematical problems, Hybrid-Game on NSGA-II employs two Nash-Players (Nash-Player 1 and Nash-Player 2) and one Pareto-Player (NSGA-II). Pareto-Player (NSGA-II) will minimize all fitness functions (f_1 and f_2) while Nash-Game decomposes this multiobjective problem into two single-objective problems; Nash-Player 1 minimizes fitness function 1 (f_1) while Nash-Player 2 minimizes fitness function 2 (f_2) with fixed elite design obtained by Nash-Player 1. Each Nash-Player will take into account all constraints since a set of elite designs should satisfy all constraints to be seeded to Pareto-Player. Nash-Game has the same size population as Pareto-Player and will run for the same generation/function evaluations as the Pareto-Player. Details of Hybrid-Game setup for mathematical benchmark are shown in Table I.

TABLE I
Hybrid-Game Setting on NSGA-II for Mathematical-Benchmark

Description	Hybrid-Game			NSGA-II
	Pareto-P	Nash-P1	Nash-P2	
Fitness	f_1 and f_2	f_1	f_2	f_1 and f_2
Constraints				
(Section-B)	C_1 and C_2	C_1 and C_2	C_1 and C_2	C_1 and C_2
(Section-C)	C_1–C_4	C_1–C_4	C_1–C_4	C_1–C_4
DVs				
(Section-A)	x_1 and x_2	x_1	x_2 with x_1^*	x_1 and x_2
(Section-B)	x_1 and x_2	x_1	x_2	x_1 and x_2
(Section-C)	h, b, l, t	h, b with l^*, t^*	l, t with h^*, b^*	h, b, l, t
Generation				
(Section-A)	50	50	50	50
(Section-B)	100	100	100	100
(Section-C)	50	50	50	50

Note: DVs represents design variables and * indicates fixed elite design variable obtained by the other Nash-Player. For constraints, Pareto-Player and Nash-Players consider same constraints since the elite design variables obtained by Nash-Players will be seeded to the Pareto-Player. If the fitness values of f_1 or f_2 are not satisfied, the constraints in Section B and C will trigger the penalty functions.

A. Nonuniformly Distributed Nonconvex Design

This problem defined in [20] considers a nonuniformly distributed nonconvex problem. It is an extended version of a nonlinear problem where the objective is to minimize (5) and (6). Random solutions are shown in Fig. 5(a)

$$f_1(x_1) = 1 - \exp(-4x_1)\sin^4(5\pi x_1) \tag{5}$$

$$f_2(x_1, x_2) = g(x_2) \cdot h(f_1(x_1), g(x_2)) \tag{6}$$

where $0 \le x_1, x_2 \le 1$

$$g(x_2) = \begin{cases} 4 - 3\exp\left(-\left(\frac{x_2 - 0.2}{0.02}\right)^2\right), & \text{if } 0 \le x_2 \le 0.4 \\ 4 - 3\exp\left(-\left(\frac{x_2 - 0.7}{0.2}\right)^2\right), & \text{if } 0.4 \le x_2 \le 1 \end{cases}$$

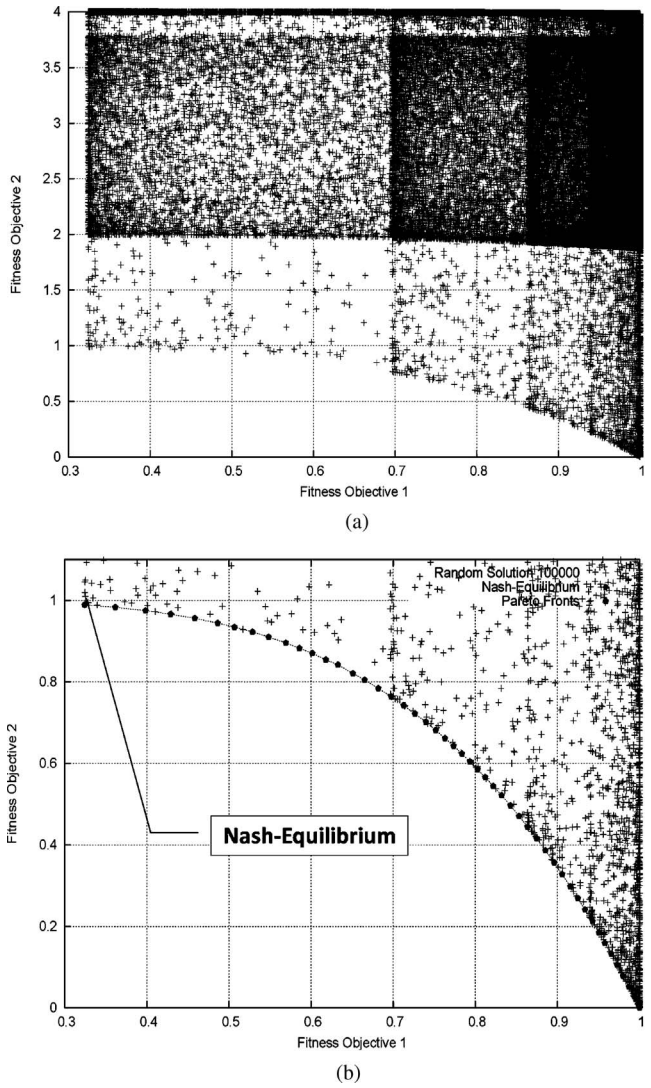


Fig. 5. (a) Random solutions (Section-A). (b) True Pareto front obtained by Hybrid-Game on HAPMOEA (Section-A).

$$h(f_1, g) = \begin{cases} 1 - \left(\frac{f_1}{g}\right)^{\alpha}, & \text{if } f_1 \leq g \\ 0, & \text{otherwise } \alpha = 4. \end{cases}$$

The Hybrid-Game on HAPMOEA was allowed to run for 15 000 function evaluations and it successfully produces true Pareto-optimal fronts as shown in Fig. 5(b).

Fig. 6(a) and (b) shows the initial population obtained by NSGA-II and Hybrid-Game on NSGA-II. It can be seen that NSGA-II found nine Pareto members with better fitness values for the objective 1 when compared to Pareto-Player in Hybrid-Game which found seven Pareto members. One thing should be noticed here is that the elite design obtained by the Nash-Players of Hybrid-Game is located almost near the global solutions as shown in Fig. 6(b). This elite design will be seeded to the population of Pareto-Player where Pareto members 1 to 5 are dominated by the elite design obtained by Nash-Players. In other words, the elite design of Nash-Game will become Pareto member 1 in the population of Pareto-optimality in the following generation. The next individuals in the population of Pareto-Player will be located around this elite design.

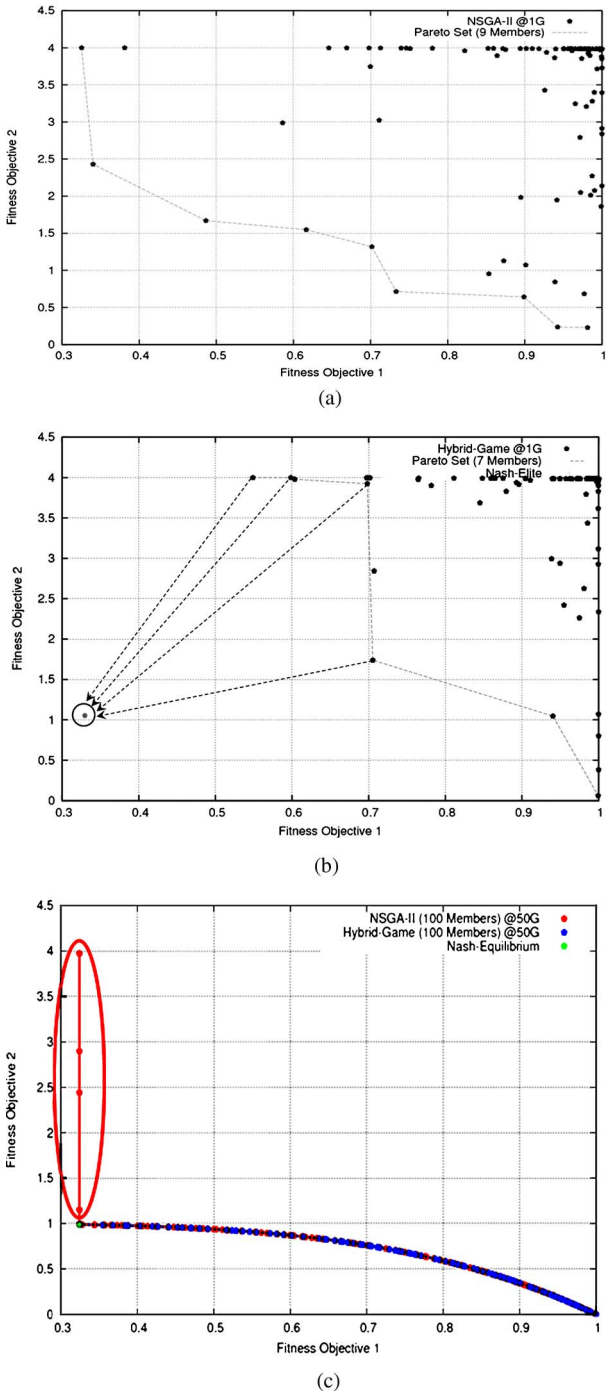


Fig. 6. (a) Initial population obtained by NSGA-II. (b) Initial population obtained by Hybrid-Game on NSGA-II. (c) Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 50 generations (Section-A).

Fig. 6(c) compares the convergence obtained by NSGA-II and Hybrid-Game on NSGA-II. The optimization is stopped after 50 generations with a population size of 100. It can be seen that the Hybrid-Game helps NSGA-II to find true Pareto front faster while the NSGA-II without Hybrid-Game needs more function evaluations for the Pareto members as marked by the red circle. Numerical results clearly show the benefits of using Hybrid-Game.

B. Discontinuous Multiobjective (TNK) Design

The problem TNK proposed in [21] considers minimization of (7) and (8). Random solutions are shown in Fig. 7(a)

$$f_1(x_1) = x_1 \tag{7}$$

$$f_2(x_2) = x_2 \tag{8}$$

subject to

$$C_1(x_1, x_2) = -x_1^2 - x_2^2 + 1 + 0.1\cos\left(16\arctan\frac{x_1}{x_2}\right) \le 0$$

$$C_2(x_1, x_2) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \le 0.5$$

where $0 \le x_1, x_2 \le \pi$.

The Hybrid-Game on HAPMOEA was allowed to run for 30 000 function evaluations and it successfully produces true Pareto-optimal fronts as shown in Fig. 7(b).

Fig. 7(c) compares the convergence obtained by NSGA-II and Hybrid-Game on NSGA-II. The optimization is stopped after 100 generations with a population size of 100. It can be seen that the NSGA-II need more function evaluations to find Pareto members in the Section-A marked with red square while the Hybrid-Game produces a true Pareto front.

C. Nonlinear Goal Programming Design in Mechanical Problem

The problem is a well-known mechanical design optimization problem [22]. A beam needs to carry a certain load F after welding a beam to another beam as shown in Fig. 8. This problem desires to find four optimal design parameters including the thickness of beam (b), width of the beam (t), length of weld (l), and weld thickness (h). The length of the overhang beam is 14 in and the force ($F = 6000$ lb) is applied at the end of overhang beam.

The goal programming objective is to minimize the cost and deflection of beam. The goals are shown in (9) and (10) with four constraints (C_1, C_2, C_3, C_4,). The first constraint is to make sure that the shear stress developed at the support position is smaller than the allowable shear strength (13 600 psi). The second is that the normal stress developed at the support location is to be smaller than the allowable yield strength (30 000 psi). The third is that the thickness of the beam is not smaller than the weld thickness from a practical standpoint. The fourth is the allowable buckling load along t direction is more than the applied load F. The goal functions are converted to objective/fitness functions as indicated in (11) and (12). Random solutions are shown in Fig. 9(a). The goals are

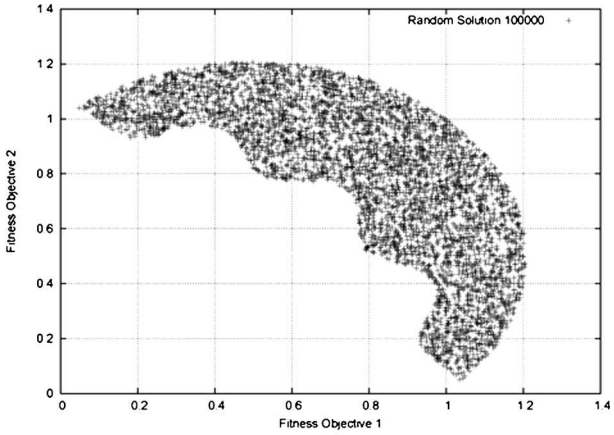
$$goal_1(f_1(h, b, l, t) = 1.10471h^2l + 0.04811tb(14.0 + l) \le 5.0) \tag{9}$$

$$goal_2(f_2(h, b, l, t) = \frac{2.1952}{t^3b} \le 0.001) \tag{10}$$

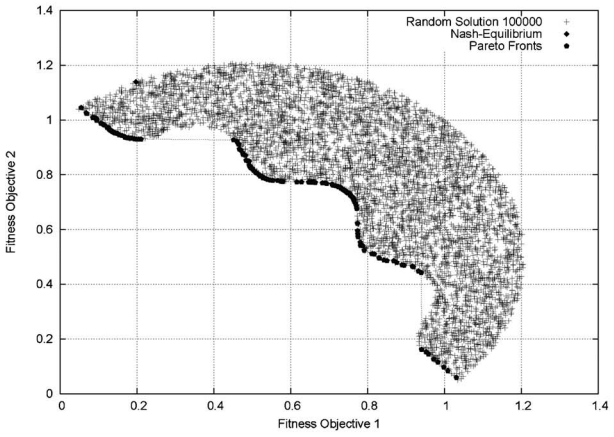
subject to

$$C_1(\tau) = 13\,600 - \tau(h, l, t) \ge 0$$

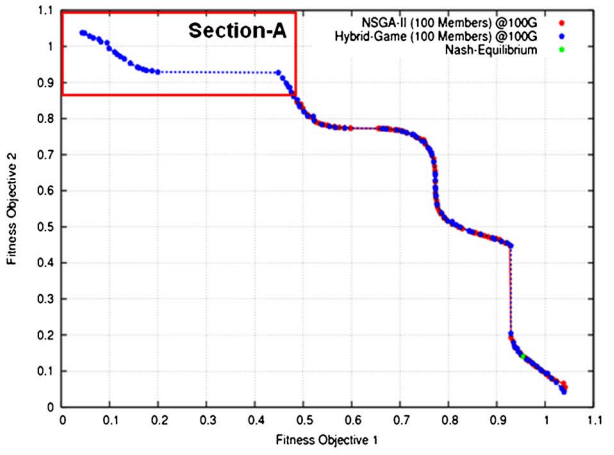
$$C_2(\sigma) = 30\,000 - \sigma(b, t) \ge 0$$



(a)



(b)



(c)

Fig. 7. (a) Random solutions (Section-B). (b) True Pareto front obtained by Hybrid-Game on HAPMOEA (Section-B). (c) Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 100 generations (Section-B).

$$C_3(h, b) = b - h \ge 0$$

$$C_4(P_c) = P_c(t, b) - 6000 \ge 0$$

where $0.125 \le h, b \le 5.0,\ 0.1 \le l, t \le 10.0$

$$\tau(h, l, t) = \sqrt{\tau'^2 + \tau''^2 + l\tau'\tau''/\sqrt{0.25\left(l^2 + (h + t)^2\right)}}$$

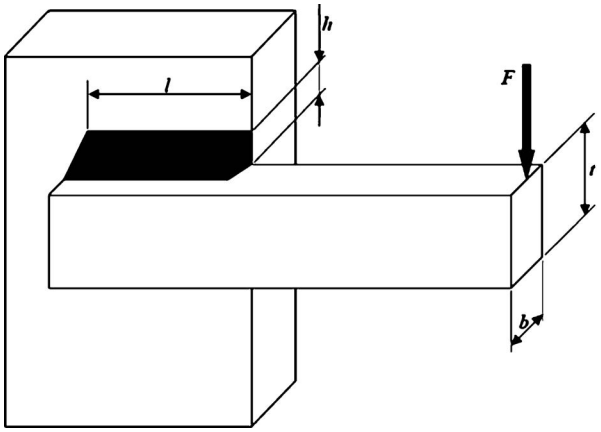


Fig. 8. Welded beam.

$$\tau' = \frac{6000}{\sqrt{2}hl}$$

$$\tau''(h,l,t) = \frac{6000\,(14+0.5l)\sqrt{0.25\left(l^2+(h+t)^2\right)}}{2\left[0.707hl\left(l^2/12+0.25\,(h+t)^2\right)\right]}$$

$$\sigma(b,t) = \frac{504\,000}{t^2 b}$$

$$P_c(t,b) = 64\,746.022\,(1-0.0282346t)\,tb^3.$$

The objective/fitness functions from goal programming (9) and (10) can be written now as (11) and (12)

$$fitness_1\,\langle f_1(h,b,l,t)-5\rangle \tag{11}$$

$$fitness_2\,\langle f_2(h,b,l,t)-0.001\rangle\,. \tag{12}$$

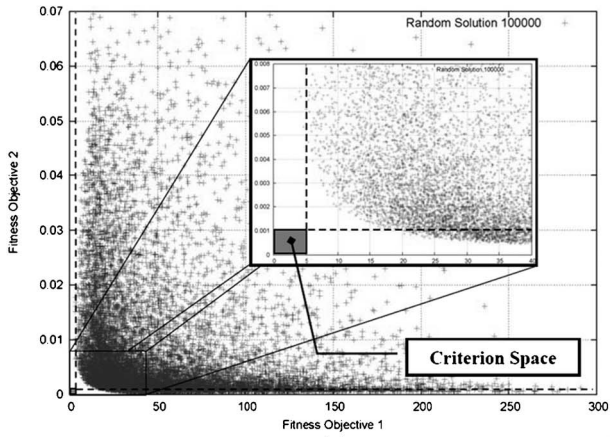
The Hybrid-Game on HAPMOEA was allowed to run for 50 000 function evaluations and it successfully produces true Pareto-optimal fronts as shown in Fig. 9(b).

Fig. 9(c) compares the convergence obtained by NSGA-II and Hybrid-Game on NSGA-II without goal programming. The optimization is stopped after 50 generations with a population size of 100. It can be seen that the NSGA-II need more function evaluations to be convergent while the Hybrid-Game produces a true Pareto front.

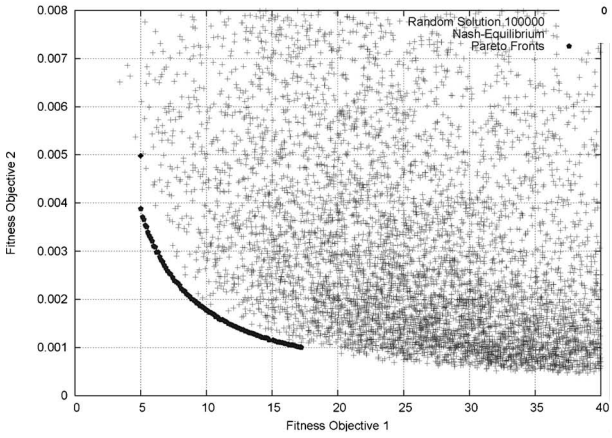
To conclude the comparison between NSGA-II and Hybrid-Game on NSGA-II, the Hybrid-Game accelerates the searching speed of NSGA-II to capture the true Pareto front for nonuniformly distributed nonconvex, discontinuous, and mechanical design problems. In addition, the elite design obtained by Nash-Players is better than the solutions obtained by Pareto-Player at the beginning of optimization due to the decomposition of the multiobjective design problem into two single-objective problems by Nash-Game.

D. Discussion on Hybrid-Game (Pareto-Optimality+Dynamic Nash-Game)

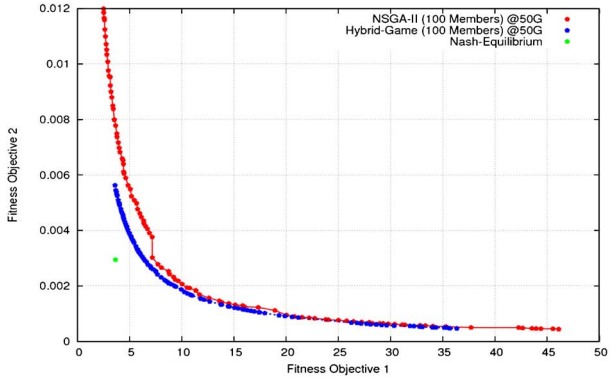
To summarize validation test cases, the solutions in the Pareto nondominated front obtained by Pareto-Player (P-Player) may not be good as Nash-solution at the beginning



(a)



(b)

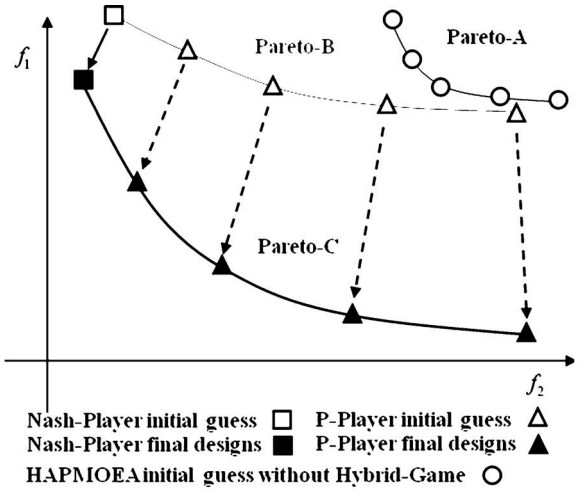


(c)

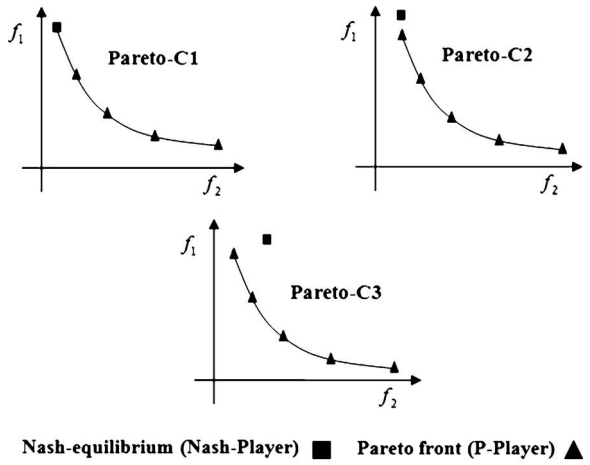
Fig. 9. (a) Random solutions (Section-C). (b) True Pareto front obtained by Hybrid-Game on HAPMOEA (Section-C). (c) Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 50 generations (Section-C).

of the optimization. In other words, an elite solution from Nash-Players is not enough to produce all good nondominated solutions, however, P-Player benefits from the use of the elite designs obtained by Nash-Players at these initial stages. For instance, Fig. 10(a) shows the progress of Pareto front for a two-objective design problem. There are two different initial guesses with and without Nash-Players. To produce *Pareto-C*, P-Player still needs to search (hidden line arrows) with improvement of Nash-solution (line arrow).

The important discussion point is that the Nash-equilibrium can be within the Pareto nondominated solutions obtained by



(a)



(b)

Fig. 10. (a) Comparison of Nash-equilibrium and Pareto-optimal front. *Pareto-A* can be initial guess of P-Player without Nash-Players (HAPMOEA). *Pareto-B* can be produced by P-Player with Nash-Players (Hybrid-Game). *Pareto-C* is the final nondominated solutions. (b) Comparison of initial guess obtained by HAPMOEA and Hybrid-Game.

P-Player as shown *Pareto-C1* or *Pareto-C2* in Fig. 10(b). A Nash-equilibrium can be one of the nondominated solutions since the elite designs obtained by the Nash-Players are seeded to P-Player population if the Nash-solution is better or nondominated by nondominated solutions from P-Player. A Nash-solution can be located like *Pareto-C3* when the Nash-solution is not better than the solutions from P-Player.

Reference [23] shows another validation of Hybrid-Game for a real-world design problem. Lee, Gonzalez, Périaux, and Srinivas considered the reconstruction design for 3-D ONERA M6 wing using HAPMOEA and multifidelity Hybrid-Game, and compared the optimization efficiency and solution quality. Numerical results obtained by [23] show that Hybrid-Game is 75% more efficient when compared to HAPMOEA for the reconstruction design problem. Reference [24] shows that the Hybrid-Game can also be implemented to the nondominated sorting genetic algorithm II (NSGA-II) and comparison between Hybrid-Game and NSGA-II. The optimization efficiency of NSGA-II can be improved by 80% using Hybrid-Game for mission path planning system design problems.

It may not be a fair comparison between Hybrid-Game on NSGA-II and NSGA-II itself due to the different size of population, however, it can be seen that the Hybrid-Game helps NSGA-II to converge faster than NSGA-II itself. The Nash algorithm running in parallel with Pareto optimizer operates, in numerical analysis terminology, as a pre-conditioner for the Pareto optimizer. Broadly speaking, for many difficult problems a Pareto-only game optimizer will require many generations or function evaluations to reach convergence.

The diversity introduced by elite information from the Nash-Players will speed up the convergence of the Pareto optimizer (NSGA-II or others). In other words, the elite Nash information will speed up the convergence of the NSGA-II.

It should also be remembered that Nash-solutions may not be very far from nondominated solutions and that a Nash-Game is much cheaper to compute when compared to a Pareto-Game, accelerating therefore the convergence to the optimal Pareto front. Moreover, the Nash-Player pre-conditioners are run in parallel with the Pareto-Player; this additional "Nash time" is not sequentially added to the performance evaluation of the global optimization.

This paper focuses on comparing the optimization efficiency and solution quality of multifidelity/population HAPMOEA and Hybrid-Game for solving the uncertainty based multidisciplinary design problem.

IV. Analysis Tools

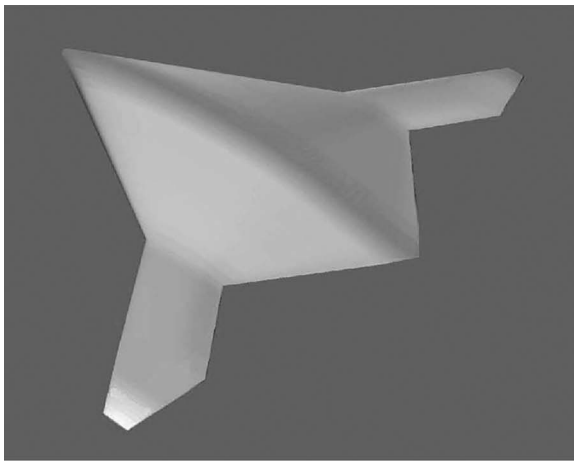
In this sequel, two analysis tools are considered for robust MDO. For aerodynamic analysis, both FLO22 and FRICTION software are utilized to compute aerodynamic characteristics on 3-D wing while POFACETs is used to estimate radar cross section (RCS) on the UAS.

A. Aerodynamic Analysis Tools

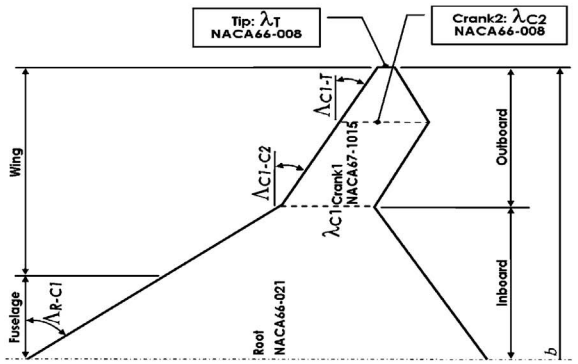
In this paper, the potential flow solver that is used has capabilities for analysing inviscid, isentropic, transonic shocked flow past 3-D swept wing configurations [25]. Friction drag is externally computed by utilising the program FRICTION code [26] which provides an estimation of the laminar and turbulent skin friction suitable for use in aircraft preliminary design. Details on the validation of the potential flow solver can be found in [27] where it is shown that the results obtained by the potential flow solver are in good agreement with experimental data.

B. Electromagnetic Analysis Tools

POFACETs [28], developed at the Naval Postgraduate School, is a numerical implementation of a physical optics approximation for predicting the RCS of complex 3-D objects. The software calculates the mono-static or bi-static RCS values of the object for radar frequency and illumination parameters specified by the user and displays plots for the model geometry and its RCS. Details of POFACETs can be found in [1], [28].



(a)



(b)

Fig. 11. (a) Baseline design (3-D view). (b) Baseline UCAV configuration.

V. Real-World Design Problems

In this section, the Hybrid-Game is used to show the benefit of using Nash-Game and Pareto-Game simultaneously. To do so, results obtained by Hybrid-Game will be compared to the results obtained by HAPMOEA in terms of solution quality and computational expense. The test is extended work of [4], [30] for fast convergence in complex MO/MDO detailed design problems.

A. Formulation of Design Problem

The type of vehicle considered in this section is a joint unmanned combat air vehicle (J-UCAV) that is similar in shape to Northrop Grumman X-47B [29]. The baseline UCAV is shown in Fig. 11(a) and (b).

The wing planform shape is assumed to be an arrow shape with jagged trailing edge. The aircraft's maximum gross weight is approximately 46 396 lb (21 045 kg) and empty weight is 37 379 lb (16 955 kg). The design parameters for the baseline wing configuration are illustrated in Fig. 11(b) and Table II. In this test case, the fuselage is assumed from 0 to 25% of the half span. The crank positions are at 46.4% and 75.5% of the half span. The inboard and outboard sweep angles are 55° and 29°. Inboard and outboard taper ratios are 20% and 2% of the root chord, respectively.

It is assumed that the baseline design contains three types of airfoils at root, crank1, crank2, and tip section; NACA 66-021 and NACA 67-1015 are located at inboard, and two NACA 67-008 are placed at the outboard sections. These airfoils are shown in Fig. 12. The maximum thickness at the root section is 21% of the root chord that is about 3% thicker than that of the X-47B to increase avionics, fuel capacity, and missile payloads.

TABLE II
Baseline UCAV Wing Configurations

AR	b (m)	Λ_{R-C1}	Λ_{C1-C2}	Λ_{C1-T}	λ_{C1}	λ_{C2}	λ_T	Γ
4.4	18.9	55°	29°	29°	20	20	2	0°

Note: Taper ratio (λ) is $\%C_{\text{Root}}$.

AR: aspect ratio, b: span length, Λ: sweep angle, λ: taper ratio, Γ: dihedral angle.

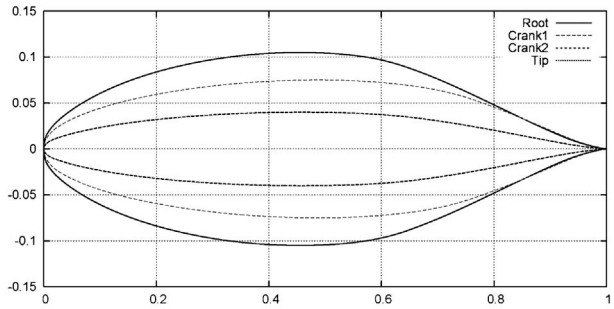


Fig. 12. Baseline UCAV wing airfoil sections.

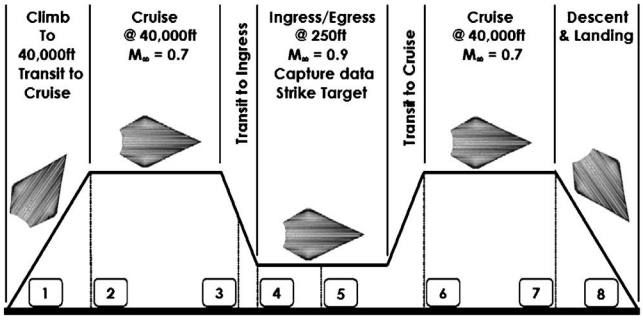
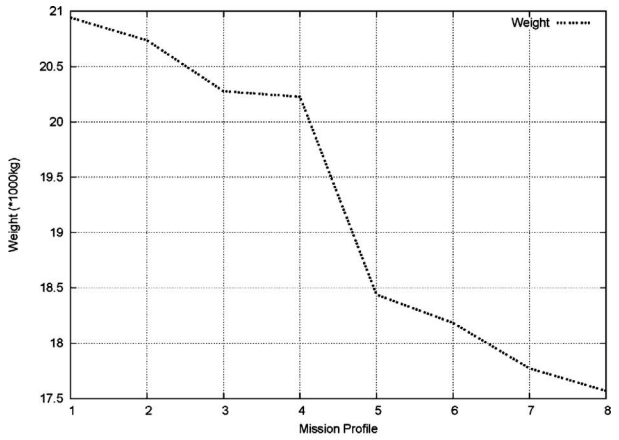


Fig. 13. Mission profile of baseline UCAV. Sector 1: T/O climb, Sector 2: cruise, Sector 3: transition dash, Sector 4: ingress, Sector 5: target strike, Sector 6: return cruise, Sector 7: end return cruise, Sector 8: decent and land.

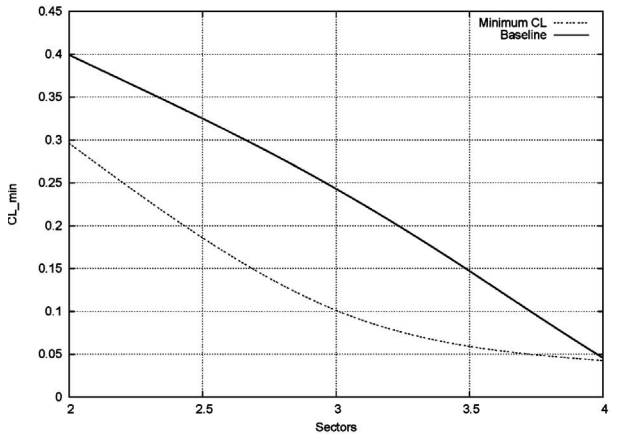
The mission profile consists of reconnaissance, intelligence, surveillance, and target acquisition (RISTA) as illustrated in Fig. 13. The mission is divided into eight Sectors.

Fig. 14(a) shows the weight distribution along the mission profile (Sector 1–Sector 8). The weight between Sector 4 and Sector 5 is significantly reduced since 80% of ammunitions weight is used for target strike.

In this case, the critical sectors are Sector 2 to Sector 4. The minimum lift coefficients ($C_{L_{\text{Minimum}}}$) for these sectors are shown in Fig. 14(b). The baseline design produces 30% higher lift coefficient in Sector 2 when compared to $C_{L_{\text{Minimum}}}$ while its lift coefficient is only 7% higher in Sector 4. The aim of this optimization is the improvement of aerodynamic performance in Sector 4 while maintaining aerodynamic performance in Sector 2.



(a)



(b)

Fig. 14. (a) Weight distribution along the mission. (b) $C_{L_{\text{Min}}}$ for Sector 2 to Sector 4.

TABLE III
WING PLANFORM DESIGN BOUNDS

S_1	S_2	S_3	Λ_{R-C1}	Λ_{C1-C2}	Λ_{C1-T}	λ_{C1}	λ_{C2}
50.46	10.09	5.05	49.5°	25°	25°	15	15
63.92	16.82	10.09	60.5°	35°	35°	45	45

Note: Taper ratio (λ) is $\%C_{\text{Root}}$, area (S) is in m^2 and one geometrical constraint is applied $\lambda_{C2} \leq \lambda_{C1}$.

B. Representation of Design Variables

The aerofoil geometry is represented using Bézier curves with a combination of a mean line and thickness distribution control points. The upper and lower bounds for mean and thickness control points at root, crank1, crank2 and tip sections are illustrated in Fig. 15(a)–(d). Each aerofoil considers 17 control points.

The wing planform shape is parameterized by considering the variables described in Fig. 16. The design bounds are shown in Table III where three wing section areas, three sweep angles, and two taper ratios are considered. This leads to different span length (b) and aspect ratio (AR). One constraint is that the taper ratio at crank2 is not higher than the taper ratio at crank1, i.e., ($\lambda_{C2} \leq \lambda_{C1}$).

TABLE IV
DISTRIBUTIONS OF DESIGN VARIABLES

Design Variables	Hybrid-Game on HAPMOEA						HAPMOEA-L3
	NP1	*NP2*	*NP3*	*NP4*	*NP5*	*PP*	
A_{Root}	√					√	√
A_{Crank1}		√				√	√
A_{Crank2}			√			√	√
A_{Tip}				√		√	√
Wing					√	√	√

Note: Design variable A_{Root} indicates aerofoil at root section and NP_i represents the ith Nash-Player and PP indicates the Pareto-Player.

C. Hybrid-Game (Pareto+Nash) Setup

The Hybrid-Game employs five Nash-Players and one Pareto-Player as shown in Table IV. The Pareto-Player of Hybrid-Game solely considers all 76 design variables for the shape of aerofoil sections and wing planform. Aerofoil sections at root, crank1, crank2 and tip are optimized by Nash-Players 1 to 4 while Nash-Player 5 optimizes wing planform only. In other words, each Nash-Player from 1 to 4 will optimize 17 aerofoil design variables while Nash-Player 5 will consider eight wing planform design variables. In contrast, each node (Node0-6) of HAPMOEA will consider all 76 design variables, including aerofoil sections and wing planform.

D. Uncertainty Based Multidisciplinary Design Optimization of UCAV

1) *Problem Definition:* This test case considers the multidisciplinary design optimization of UCAV when there is uncertainty in the operating conditions and a robust design technique is required. This problem is selected to show the benefits of using the Hybrid-Game (Nash+Pareto) method since the addition of uncertainty increases the computational cost considerably. The objectives are to maximize Aerodynamic Quality (AQ) while minimizing electro-magnetic quality (EQ) to maximize the survivability of the UCAV. AQ is defined by fitness functions 1 (mean) and 2 (variance) that represent an aerodynamic performance and sensitivity corresponding to the five variability of flight conditions including Mach, angle of attacks and altitude. EQ is expressed using one normalized equation; fitness functions 3 which represents the magnitude and sensitivity of RCS for a given UAV shape at five variability radar frequencies. UAV will have less chance to be detected by enemy radar systems if the value of EQ is low. In other words, UAV will be stealthier. The fitness functions for Pareto-Player and Nash-Players are indicated in Table V.

The possible uncertainty flight conditions (five Mach numbers, angle of attacks and altitudes) and five radar frequencies are

$$M_{\infty i} \in [0.75, 0.775, M_S = 0.80, 0.825, 0.85]$$

$$\alpha_{\infty i} \in [4.662, 3.968, \alpha_S = 3.275°, 2.581, 1.887]$$

$$ALT_{\infty i} \in [30\,062, 25\,093, ALT_S = 20\,125\,\text{ft}, 15\,156, 10\,187]$$

$$F_{\infty i} \in [1.0, 1.25, F_S = 1.5\ \text{GHz}, 1.75, 2.0]\,.$$

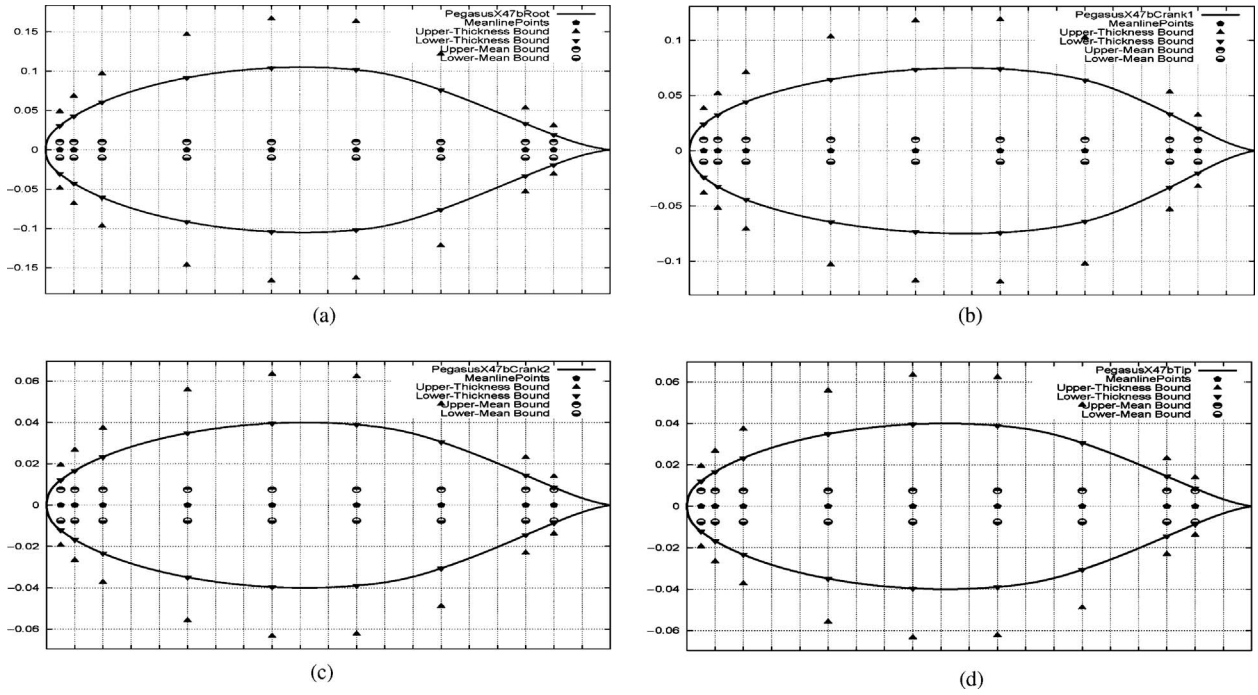


Fig. 15. (a) Control points at root section. (b) Control points at crank1 section. (c) Control points at crank2 section. (d) Control points at tip section.

TABLE V
FITNESS FUNCTIONS FOR HYBRID-GAME

Player	Fitness Function	Optimization Criteria
PP	$f_{1\_PP} = \min\left(1\Big/\left(\overline{L/D}\right)\right)$ $f_{2\_PP} = \min\left(\delta\left(L/D\right)\right)$ $f_{3\_PP} = \min(EQ)$	Optimize wing planform and aerofoil sections at root, crank1, crank2 and tip to maximize $\overline{L/D}$, and minimize $\delta\left(L/D\right)$ and EQ.
NP1	$f_{1\_NP1} = \min\left(1/AQ\right)$	Optimize root aerofoil section only to maximize AQ with fixed aerofoil sections (crank1, crnak2, tip) and wing planform.
NP2	$f_{1\_NP2} = \min\left(1/AQ\right)$	Optimize crank1 aerofoil section only to maximize AQ with fixed aerofoil sections (root, crnak2, tip) and wing planform.
NP3	$f_{1\_NP3} = \min\left(1/AQ\right)$	Optimize crank2 aerofoil section only to maximize AQ with fixed aerofoil sections (root, crank1, tip) and wing planform.
NP4	$f_{1\_NP4} = \min\left(1/AQ\right)$	Optimize tip aerofoil section only to maximize AQ with fixed aerofoil sections (root, crank1, crank2) and wing planform.
NP5	$f_{1\_NP5} = \min(EQ)$	Optimize wing planform shape only to minimize EQ with fixed aerofoil sections (root, crank1, crank2, tip).

Note: $AQ = \frac{\overline{L/D}}{\delta\left(L/D\right)}$ and $EQ = \delta RCS_{Mono\&Bi} + \overline{RCS_{Mono\&Bi}}$ $\overline{RCS}_{Mono\&Bi} = \frac{1}{2}\left(\overline{RCS}_{Mono} + \overline{RCS}_{Bi}\right)$ and

$\delta RCS_{Mono\&Bi} = \frac{1}{2}\left(\delta RCS_{Mono} + \delta RCS_{Bi}\right)$

Mono-static : $\theta = [0° : 3° : 360°]$ and $\phi = [0° : 0° : 0°]$
Bi-static : $\theta_0 = 135°$ and $\phi_0 = 0°$
$\theta = [0° : 3° : 360°]$ and $\phi = [0° : 0° : 0°]$

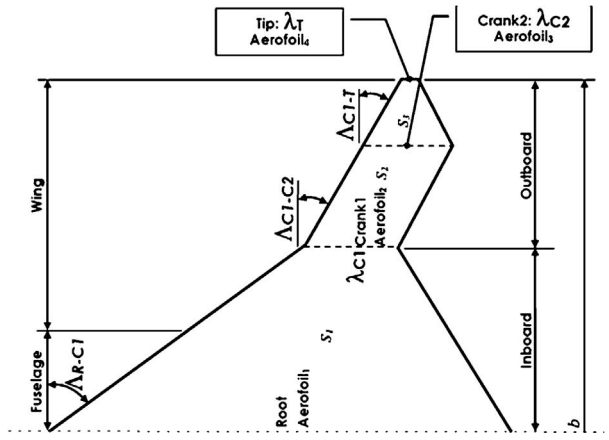
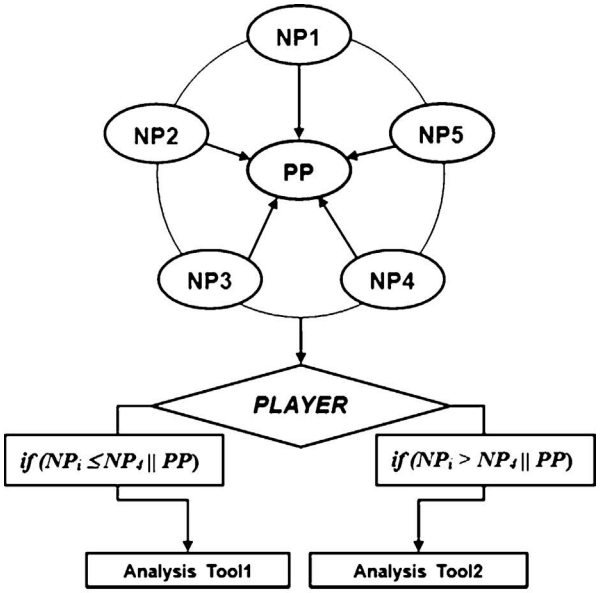


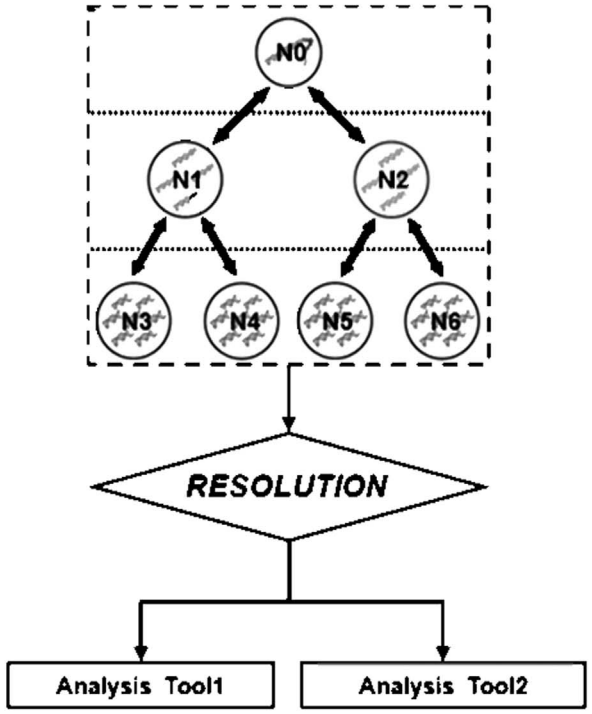
Fig. 16. Wing planform design variables.

The uncertainty flight conditions are taken from Sector 2.5 (middle of cruise) to Sector 3.5 (right after high transition dash/before target acquisition) as shown in Fig. 13. Since there is a dramatic change between Sector 2 and Sector 4, where the Mach and altitude number changes from 0.7 to 0.9 and from 41 000 ft to 250 ft, respectively. In other words, the changes of flight conditions will lead to dramatic change (fluctuation) in aerodynamic performance which may cause structural or flight control failure. In addition, the altitude change means that the enemy radar systems are also changed from Mono-static to Bi-static with higher radar frequencies. This is the reason that the range from Sector 2.5 to Sector 3.5 is chosen to prevent the aerodynamic and electromagnetic fluctuation.

The fitness/objective functions of Nash-Players (1 to 5) are defined using variance to mean ratio (VMR) to minimize the number of Nash-Players. Otherwise, Hybrid-Game will



(a)



(b)

Fig. 17. (a) Evaluation mechanism of Hybrid-Game. (b) Evaluation mechanism of HAPMOEA-L3.

use ten Nash-Players if the objective functions of *AQ* and *EQ* are defined by a separated mean and variance formulas. VMR is a statistical formula to minimize variance value while maximizing mean value of objective.

The aerodynamic or electro-magnetic analysis tools used in this multidisciplinary design optimization will be determined by the objective of each player for Hybrid-Game [Fig. 17(a)]. It can be seen that Nash-Players (1 to 4) in Hybrid-Game use aerodynamic analysis tools only to maximize *AQ* while Nash-Player 5 in Hybrid-Game uses electromagnetic analysis tool only to minimize *EQ*. Pareto-Player in Hybrid-Game uses both aerodynamic and electromagnetic analysis tools for both *AQ* and *EQ*.

In contrast, the HAPMOEA-L3 uses both aerodynamic and electromagnetic analysis tools as shown in Fig. 17(b). This is because that each node in HAPMOEA considers both *AQ* and *EQ* in multifidelity model. Therefore, Hybrid-Game will have more chance to evaluate candidates.

2) *Interpretation of Numerical Results:* Both HAPMOEA and Hybrid-Game use two 2.4 GHz processors. The HAPMOEA algorithm was allowed to run approximately for 540 function evaluations and took 200 h while Hybrid-Game algorithm was run approximately for 400 function evaluations and took 60 h which is 30% of the computation cost of HAPMOEA.

The Pareto fronts obtained by HAPMOEA and Hybrid-Game are compared to the baseline design in Fig. 18(a)–(d).

It can be seen that Hybrid-Game produces similar solutions when compared to HAPMOEA. The black inverse triangle (Pareto member 1) represents the best solution for fitness function1. The red square (Pareto member 9) represents the best solution for fitness function 2. The blue triangle (Pareto member 10) indicates the best solution for the third fitness. The green square (Pareto member 4 and 5) indicates the compromised solution for Hybrid-Game. It can be seen all Pareto members produce higher lift to drag ratio (fitness 1) with low sensitivity (fitness 2) and also their wing planform shapes have lower *EQ* (fitness 3).

Table VI compares the mean and variance of lift to drag ratio and RCS quality of Pareto members (1, 8, 10) from HAPMOEA, Pareto members (1, 5, 9, 10) from Hybrid-Game and the baseline design. Even though the Hybrid-Game ran only 30% of HAPMOEA computational time, it produces similar nondominated solutions when compared to Pareto nondominated solutions obtained by HAPMOEA.

Fig. 19(a) and (b) compares the wing planform shape corresponding to Pareto nondominated solutions obtained by HAPMOEA and Hybrid-Game, and the baseline design. It can be seen that the Hybrid-Game has more variety on wing planform shapes. This may be due to the evaluation mechanism of Hybrid-Game that allows Pareto-Player and Nash-Player 5 to have a detailed design after running more function evaluations.

The Sector sweep is plotted with the lift and drag coefficient obtained by HAPMOEA, Hybrid-Game and the baseline design as shown in Fig. 20(a) and (b). The range of sector sweep is $M_\infty \in [0.7{:}0.9]$, $\alpha \in [6.05°{:}0.5°]$ and altitude $(ft) \in [40\,000{:}250]$. It can be seen that the Hybrid-Game produces a set of comparable solutions to HAPMOEA even though Hybrid-Game ran only 30% of HAPMOEA computational time. The best solution for objective 1 (BO1) from Hybrid-Game has higher C_L values while Pareto member 1 (BO1) from HAPMOEA produces a lower drag along the sector sweep.

Table VII compares the quality of drag coefficient obtained by HAPMOEA, Hybrid-Game and the baseline design using the uncertainty mean and variance statistical formulas. It can

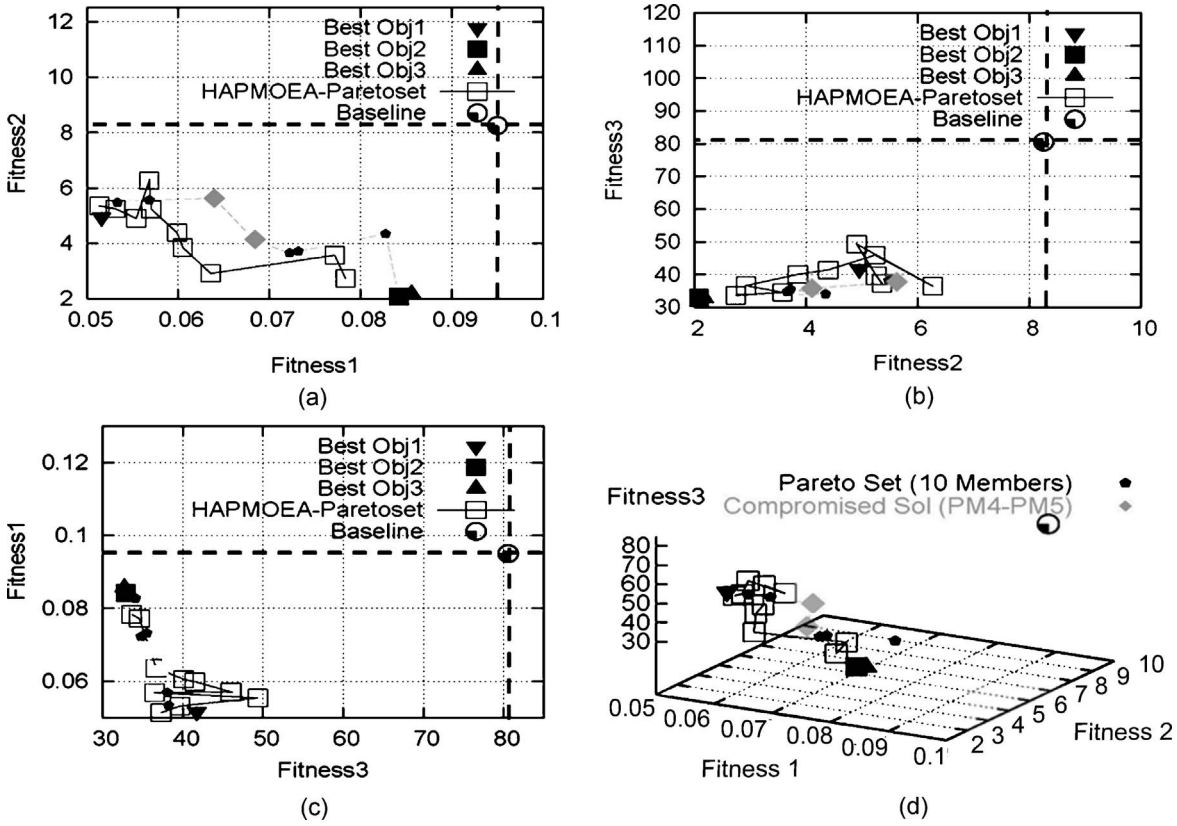


Fig. 18. (a) Fitness 2 versus Fitness 1. (b) Fitness 3 versus Fitness 2. (c) Fitness 1 versus Fitness 3. (d) Pareto nondominated solutions obtained by HAPMOEA and Hybrid-Game.

TABLE VI
COMPARISON OF FITNESS VALUES OBTAINED BY HAPMOEA AND HYBRID-GAME

Objective	HAPMOEA (200 h)			Hybrid-Game (60 h)			
	PM1 (BO1)	*PM8 (CS)*	*PM10 (BO2* and *3)*	*PM1 (BO1)*	*PM5 (CS)*	*PM9 (BO2)*	*PM10 (BO3)*
Fitness11 $\Big/ \left(\overline{L/D}\right)$	0.051 (−46%)	0.063 (−34%)	0.078 (−18%)	0.051 (−46%)	0.068 (−28%)	0.084 (−12%)	0.085 (−11%)
Fitness2 $\delta\left(L/D\right)$	5.35 (−35%)	2.91 (−65%)	2.73 (−67%)	4.94 (−40%)	4.10 (−50%)	2.07 (−75%)	2.17 (−74%)
Fitness3 EQ	37.29 (−53%)	36.67 (−54%)	33.62 (−58%)	41.74 (−48%)	35.48 (−56%)	32.89 (−59%)	32.69 (−60%)

Note: The fitness values 1, 2, and 3 of the baseline model are 0.095, 8.25, and 80.58, respectively. BO_i represents the best objective solution for the ith fitness function. *CS* indicates the compromised solution.

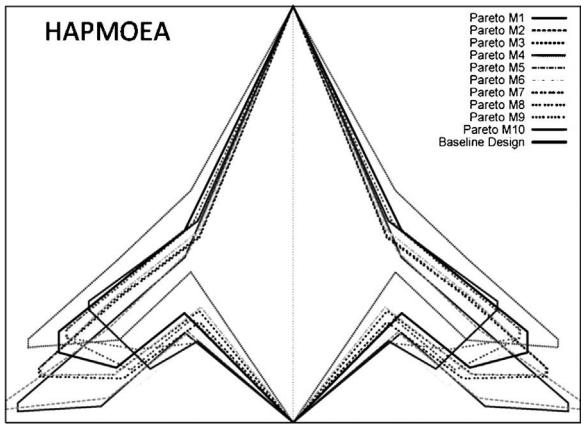
be seen that Pareto member 1 from HAPMOEA produces lower drag at (Sector 2:Sector 4) when compared to Hybrid-Game while Pareto member 9 from Hybrid-Game produces a robust design with lower sensitivity in drag.

Fig. 21 compares the lift to drag ratio distribution obtained by Pareto members (1, 8, and 10) from HAPMOEA, Pareto members (1, 4, 5, and 10) from Hybrid-Game and the baseline design. Pareto member 1 (BO1) from both HAPMOEA and Hybrid-Game is not only similarly distributed but also produces higher lift to drag ratios than others along the Sector sweep while Pareto member 9 (BO2) from Hybrid-Game produces lower sensitivity in Mach, angle of attack and altitude. It can be seen that all Pareto-solutions from HAPMOEA and Hybrid-Game have a less fluctuation (stable motions) from Sector 2.5 to Sector 3.5 due to the

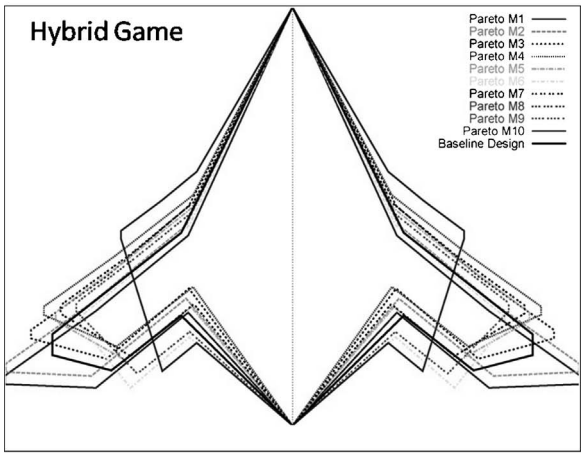
TABLE VII
COMPARISON OF DRAG QUALITY OBTAINED BY HAPMOEA AND HYBRID-GAME

Drag Quality	HAPMOEA (200 h)			Hybrid-Game (60 h)			
	PM1 (BO1)	*PM8 (CS)*	*PM10 (BO2* and *3)*	*PM1 (BO1)*	*PM5 (CS)*	*PM9 (BO2)*	*PM10 (BO3)*
$\overline{C_D}$	0.012 (−52%)	0.014 (−44%)	0.015 (−40%)	0.013 (−48%)	0.015 (−40%)	0.014 (−44%)	0.015 (−40%)
δC_D ($\times 10^{-6}$)	7.92	6.48	3.83	8.50	6.65	3.69	3.74

Note: The $\overline{C_D}$ and δC_D of baseline model are 0.025 and 5.49 $\times$ 10^{-6}, respectively. Quality is represented by mean (magnitude of performance) and variance (sensitivity/stability).

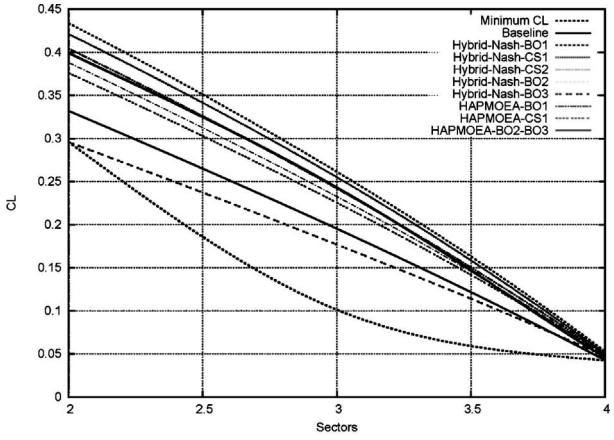


(a)

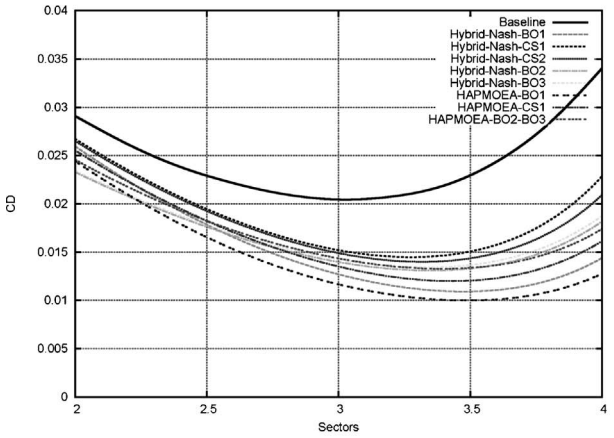


(b)

Fig. 19. (a) Wing planform shapes obtained by HAPMOEA-L3. (b) Wing planform shapes obtained by Hybrid-Game.



(a)



(b)

Fig. 20. (a) C_L versus Sectors. (b) C_D versus Sectors.

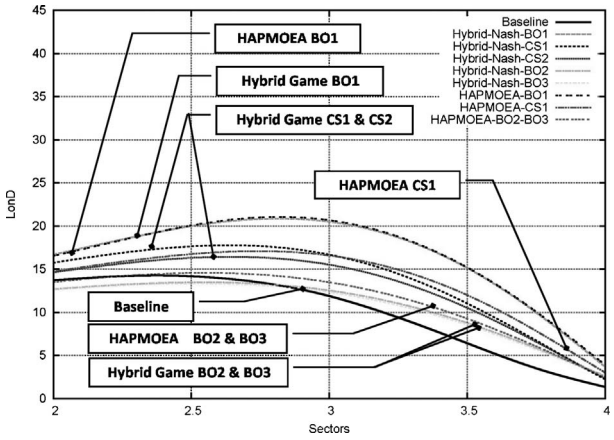


Fig. 21. L/D versus sectors.

consideration of uncertainty design during the optimization process.

Fig. 22(a) compares the mono-static RCS obtained by Pareto members 8 (CS) and 10 (BO3) from HAPMOEA, Pareto members 5 (CS) and 10 (BO3) from Hybrid-Game and the baseline design at the standard design frequency 1.5 GHz. It can be seen that Pareto members 8 and 10 from HAPMOEA produce 9% and 20% lower RCS while Pareto members 5 and 10 from Hybrid-Game produce 13% and 26% lower RCS when compared to the baseline design. Fig. 22(b) illustrates a frequency sweep $F_{\infty i} \in$ [1.0, 1.25, $F_S = 1.5$ GHz, 1.75, 2.0] corresponding to mono-static RCS analysis. The variance value (0.024) for Pareto member 5 (CS) from Hybrid-Game is lower than others while the baseline design value highly fluctuates along the frequency sweep.

Fig. 23(a) compares the bi-static RCS obtained by Pareto members [8 (CS) and 10 (BO3)], Pareto members [5 (CS) and 10 (BO3)] from Hybrid-Game and the baseline design at the standard design frequency (1.5 GHz). It can be seen that Pareto members 8 (CS) and 10 (BO3) from HAPMOEA produce 9% and 12% lower RCS while Pareto members 5 (CS) and 10 (BO3) from HAPMOEA produce 11% and 15% lower RCS when compared to the baseline design. Fig. 23(b) illustrates a frequency sweep $F_{\infty i} \in$[1.0, 1.25, $F_S = 1.5$ GHz, 1.75, 2.0] corresponding to bi-static RCS analysis. The variance value (0.09) for Pareto member 10 (BO3) from HAPMOEA is lower than other Pareto members.

The top, side, front, and 3-D view of compromised model from HAPMOEA (Pareto member 8) and Hybrid-Game (Pareto member 5) are shown in Fig. 24(a) and (b), respectively. Even though the Hybrid-Game spent less computational

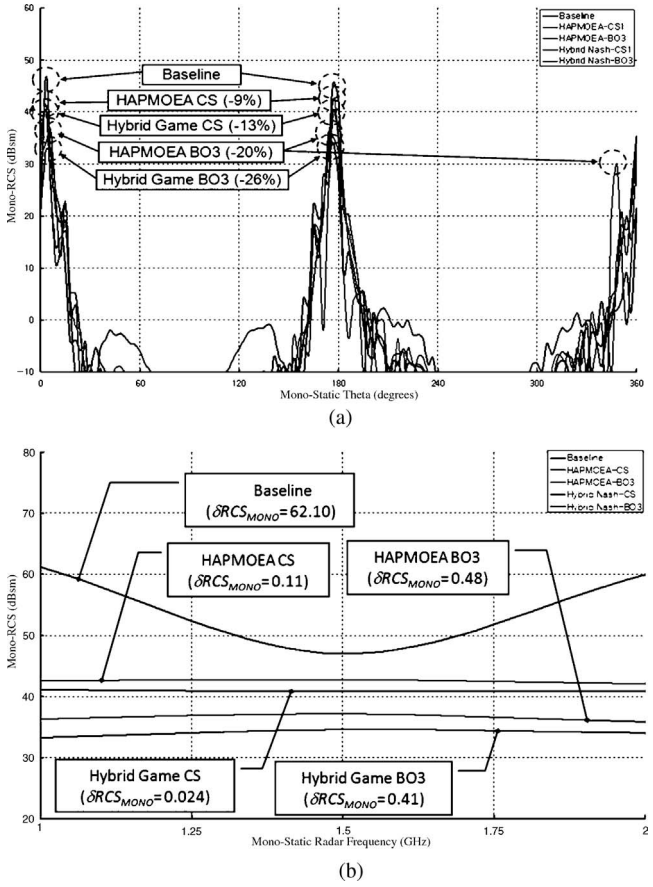


Fig. 22. (a) $RCS_{\text{Mono-Static}}$ at $F = 1.5$ GHz. (b) $RCS_{\text{Mono-Static}}$ sweep at $F \in [1.0, 1.25, F_s = 1.5\text{ GHz}, 1.75, 2.0]$.

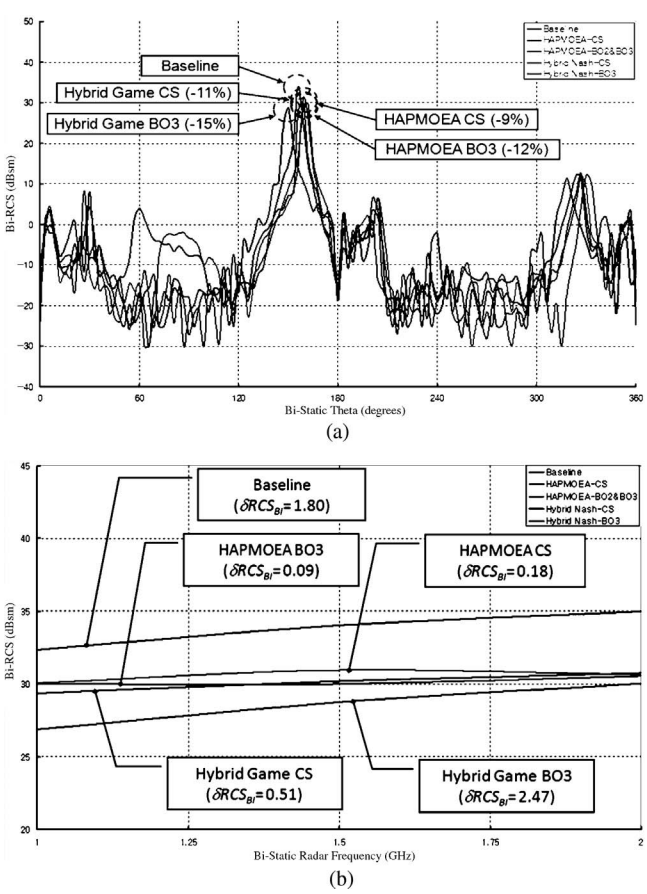


Fig. 23. (a) $RCS_{\text{Bi-Static}}$ at $F = 1.5\,\text{GHz}$. (b) $\text{RCS}_{\text{Bi-Static}}$ sweep at $F \in [1.0, 1.25, F_s = 1.5\,\text{GHz}, 1.75, 2.0]$.

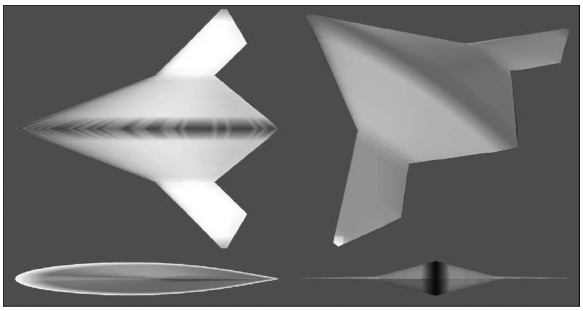
time when compared to HAPMOEA, both compromised solutions are geometrically similar.

Pareto members 8 and 10 from HAPMOEA and Pareto member 5 from Hybrid-Game can be selected as compromised solutions for further evaluation and are suitable for this RISTA stealth mission, since they have not only low observability (stealthy) at mono and bi-static radar system when compared to the baseline design but also have low sensitivity at a set of variable radar frequencies.

VI. Discussion

This paper explored the optimization methods; HAPMOEA and Hybrid-Game for robust multidisciplinary design optimization. The numerical results still give us discussion points and possible research avenues.

From a theoretical point of view, standard evolutionary algorithms cannot provide fast nondominated solutions on the Pareto front due to a tough competition between quite different chromosomes targeting different nondominated solutions. Generally, a large function evaluation is needed in the standard EA to increase the diversity and capture these nondominated Pareto-solutions. In such a situation, it is inevitable to introduce a new methodology to reduce the number of function evaluations and increasing efficiency which in turn makes evolutionary optimizer more complex. This is possible by



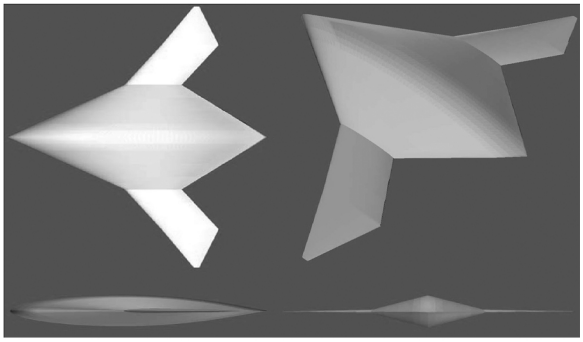


Fig. 24. (a) Pareto member 8 obtained by HAPMOEA. (b) Pareto member 5 obtained by Hybrid-Game.

introducing Nash-Game as a companion optimizer to help or guide the evolutionary optimizer to capture the Pareto front. As shown in the numerical results, the Nash-Game decomposes a complex multiobjective problem into several single-objective problems that leads to the nondominated solutions on a Pareto front are well distributed and have with quite different chromosomes, each of these nondominated solutions looks as a different species. Since the Nash optimizer is locally similar to a nondominated solution, elite information from crossover, mutations guide the Pareto optimizer to similar species, therefore reducing the number of function evaluation as compared with the standard EA and hence makes search much more efficient.

Concerning the next step of this research, the introduction of distributed optimization via game strategies is critical for large scale optimization problems. These large scale optimization problems justify the continuous effort to increase the efficiency of the optimizer by introducing new methods such as this hybrid approach. The Pareto front is now becoming an important design database in an industrial environment that offers tradeoffs and alternative solutions to the design engineer. Detailed design with complex 3-D nonlinear partial differential equations analyzers and complex 3-D geometries still has a long way to reach reasonable CPU time for computational industrial design optimization but the use of decentralized tasks coordinated via game coalition seems an interesting and important approach in parallel environments which are complemented with new advanced information technology tools to reduce design cycles.

VII. Conclusion

The optimization methods HAPMOEA and Hybrid-Game have been demonstrated and investigated. Both optimization methods find a set of useful Pareto nondominated solutions for robust multidisciplinary problems. It was also shown that the coupling of both methods with an uncertainty analysis produces higher and stable aerodynamic performance with lower and stable RCS/observability at variable flight conditions and radar frequencies. Hybrid-Game has superiority on both computational efficiency and solution quality when compared to HAPMOEA. Both methodologies couple a robust multidisciplinary evolutionary algorithm with software for aerodynamic and RCS analysis software. The results of the methods have shown the simultaneous improvement in UCAV aerodynamic performance and RCS in both mono and bi-static radar systems. Real-world design problems illustrated the applicability of methods. A family of Pareto-optimal designs obtained from optimization provided the designer with a selection to proceed into more detailed phases of the design process. The future work will focus on coupling Hybrid-Game with higher fidelity aerodynamic and electromagnetic analysis tools.

Acknowledgment

The authors wish to thank E. Whitney and M. Sefrioui for their discussions on asynchronous parallel and hierarchical EAs. They also wish to thank A. Jameson and W. H. Mason for providing access to FLO22 and FRICTION software, and F. Chatzigeorgiadis and D. C. Jenn for POFACETs software in this paper.

References

[1] D. S. Lee, L. F. Gonzalez, K. Srinivas, and J. Périaux, "Robust evolutionary algorithms for UAV/UCAV aerodynamic and RCS design optimization," *Int. J. Comput. Fluids*, vol. 37, no. 5, pp. 547–564, 2008.

[2] D. S. Lee, L. F. Gonzalez, K. Srinivas, and J. Périaux, "Robust design optimization using multiobjective evolutionary algorithms," *Int. J. Comput. Fluids*, vol. 37, no. 5, pp. 565–583, 2008.

[3] Z. Tang, J. Périaux, and J.-A. Désidéri, "Multicriteria robust design using adjoint methods and game strategies for solving drag optimization problems with uncertainties," in *Proc. East West High Speed Flow Fields Conf.*, Beijing, China, 2005, pp. 487–493.

[4] D. S. Lee, L. F. Gonzalez, J. Périaux, and K. Srinivas, *Evolutionary Optimization Methods with Uncertainty for Modern Multidisciplinary Design in Aerospace Engineering* (100 Volumes of 'Notes on Numerical Fluid Mechanics'). Heidelberg, Germany: Springer, 2009, ch. 3, pp. 271–284.

[5] D. S. Lee, L. F. Gonzalez, and E. J. Whitney, "Multiobjective, multidisciplinary multifidelity design tool: HAPMOEA – user guide, Appendix I," in *Uncertainty Based Multiobjective and Multidisciplinary Design Optimization in Aerospace Engineering*, D. S. Lee, Ed. Sydney, Australia: The University of Sydney, 2007.

[6] M. Sefrioui and J. Périaux, "Nash genetic algorithms: Examples and applications," in *Proc. Congr. Evol. Comput.*, San Diego, CA, 2000, pp. 509–516.

[7] K. Deb, "Multiobjective evolutionary algorithms: Introducing bias among Pareto-optimal solutions," KanGAL, Kanpur, India, Rep. 99002, 1999.

[8] D. S. Lee, *Uncertainty Based Multiobjective and Multidisciplinary Design Optimization in Aerospace Engineering*. Sydney, Australia: The University of Sydney, 2008, pp. 348–370.

[9] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA: MIT, 1994.

[10] Z. Michalewicz, "Artificial intelligence," in *Genetic Algorithms+Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1992.

[11] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.

[12] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, 2003.

[13] J. Wakunda and A. Zell, "Median-selection for parallel steady-state evolution strategies," in *Proc. 6th Parallel Problem Solving Nature (PPSN)*, Berlin, Germany, 2000, pp. 405–414.

[14] D. A. Van Veldhuizen, J. B. Zydallis, and G. B. Lamont, "Considerations in engineering parallel multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 144–217, Apr. 2003.

[15] M. Sefrioui and J. Périaux, "A hierarchical genetic algorithm using multiple models for optimization," in *Proc. 6th Parallel Problem Solving Nature (PPSN)*, 2000, pp. 879–888.

[16] D. S. Lee, *Uncertainty Based Multiobjective and Multidisciplinary Design Optimization in Aerospace Engineering*. Sydney, Australia: The University of Sydney, 2008, pp. 112–121.

[17] M. W. Trosset, "Managing uncertainty in robust design optimization," in *Lecture Note*. Williamsburg, VA: Department of Mathematics, College of William & Mary, 2004.

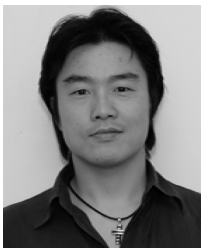
[18] D. S. Lee, L. F. Gonzalez, K. Srinivas, D. J. Auld, and K. C. Wong, "Aerodynamic shape optimization of unmanned aerial vehicles using hierarchical asynchronous parallel evolution evolutionary algorithm," *Int. J. Comput. Intell. Res.*, vol. 3, no. 3, pp. 231–252, 2007.

[19] L. F. Gonzalez, D. S. Lee, K. Srinivas, and K. C. Wong, "Single and multiobjective UAV aerofoil optimization via hierarchical asynchronous parallel evolutionary algorithm," *RAeS Aeronaut. J.*, vol. 110, no. 1112, pp. 659–672, 2006.

[20] K. Deb, "Multiobjective genetic algorithms: Problem difficulties and construction of test problems," *Evol. Comput. J.*, vol. 7, no. 3, pp. 205–230, 1999.

[21] K. Deb, "Nonlinear goal programming using multiobjective genetic algorithms," *J. Operational Res. Soc.*, vol. 52, no. 3, pp. 291–302, 2001.

[22] K. Deb, A. Pratap, and S. Moitra, "Mechanical component design for multiobjective using elitist nondominated sorting GA," KanGAL, Kanpur, India Rep. 200002, 2000.
[23] D. S. Lee, L. F. Gonzalez, K. Srinivas, and J. Périaux, "Multifidelity Nash-Game strategies for reconstruction design in aerospace engineering problems," in *Proc. 13th Australian Int. Aerospace Congr. (AIAC)*, Melbourne, Australia, 2009.
[24] D. S. Lee, J. Périaux, and L. F. Gonzalez, "UAS mission path planning system (MPPS) using hybrid-Game coupled to multiobjective optimizer (DETC2009-86749)," in *Proc. 2009 Design Eng. Tech. Conf. Comput. Inform. Eng. Conf. (ASME-IDETC/CIE)*, San Diego, CA, 2009, to be published.
[25] A. Jameson, D. A. Caughey, P. A. Newman, and R. M. Davis, "NYU transonic swept-wing computer program—FLO22," Langley Research Center, Hampton, VA, 1975.
[26] W. Mason, "Applied computational aerodynamics," in *Appendix D: Programs*. 1997.
[27] D. S. Lee, L. F. Gonzalez, K. Srinivas, D. J. Auld, and K. C. Wong, "Aerodynamic/RCS shape optimization of unmanned aerial vehicles using hierarchical asynchronous parallel evolutionary algorithms," in *Proc. 25th Appl. Aerodynamics Conf.*, San Francisco, CA, 2006.
[28] F. Chatzigeorgiadis and D. C. Jenn, "A MATLAB physical-optics RCS prediction code," *IEEE Antennas Propag. Mag.*, vol. 46, no. 4, pp. 136–139, Aug. 2004.
[29] Northrop Grumman Integrated Systems: X-47B UCAV [Online]. Available: http://www.is.northropgrumman.com/systems/nucasx47b.html
[30] D. S. Lee, L. F. Gonzalez, K. Srinivas, and J. Périaux, "Uncertainty based multidisciplinary evolutionary optimization of unmanned aerial system (UAS) using HAPMOEA," in *Proc. 5th Int. Conf. Comput. Fluid Dynamics*, Seoul, Korea, 2008.



D. S. Lee was born in Korea, on January 26, 1976. He received the B.E. degree in aerospace mechanical mechatronic engineering and the Ph.D. degree from the School of Aerospace Mechanical Mechatronic Engineering (AMME)-Honor, University of Sydney, Sydney, Australia, in 2004 and 2008, respectively.

He is currently a Researcher with the International Center for Numerical Methods in Engineering (CIMNE)/UPC, Barcelona, Spain for European Projects. His areas of expertise include computational fluid dynamics, multiobjective and multidisciplinary design optimization in aerospace engineering, evolutionary algorithms, game strategies, robust/uncertainty design. He has published two book chapters, and five journal papers. His current research interests are efficient optimization methods, unmanned aerial system (UAS), and mission trajectory planning systems (MTPS).

L. F. Gonzalez (M'06) was born in Columbia, on January 2, 1974. He received the B.E. degree, in 2002, and the Ph.D. degree in CP engineering from the School of Aerospace Mechanical Mechatronic Engineering (AMME), University of Sydney, Sydney, Australia, in 2006.

He is currently a Lecturer with the School of Engineering Systems, Queensland University of Technology, Brisbane, Australia. His areas of expertise include mechanical and project engineer in project planning, engineering design, project management, and manufacturing and maintenance service for different manufacturing, metallurgic, aeronautical, and heavy industry companies. He has published five journal papers and 20 refereed publications. His research interests include multidisciplinary design optimization, evolutionary computing, and unmanned aerials systems.

J. Périaux was born in France, on April 15, 1942. He received the Ph.D. degree from the University of Paris, Paris, France, in 1979.

He is currently an United Nations Educational, Scientific and Cultural Organization Chair with the International Center for Numerical Methods in Engineering (CIMNE)/UPC and also the Finland Distinguished Professor Programme at University Jyväskylä, Jyväskylä, Finland. His areas of expertise include numerical solution of nonlinear partial differential equations in computational fluid dynamics and electromanetics, aerodynamic design of manned/unmanned aircraft vehicles, multidisciplinary design optimization, evolutionary algorithms, and game theory.

K. Srinivas was born in India, on November 18, 1946. He received the B.E. degree, in 1968, the M.E. degree and the Ph.D. degree from the Indian Institute of Science (IISC), Bangalore, India, in 1970 and 1977, respectively.

He has been a Senior Lecturer with the School of Aerospace Mechanical Mechatronic Engineering (AMME), University of Sydney, Sydney, Australia, since 1989. His areas of expertise include computational fluid dynamics, optimization in aerospace, and biomedical engineering.