Computers & Fluids 47 (2011) 189–204

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Hybrid-Game Strategies for multi-objective design optimization in engineering

DongSeop Lee [a,b,*], Luis Felipe Gonzalez [c,1], Jacques Periaux [a,d,2], Karkenahalli Srinivas [b,3], Eugenio Onate [a,d,4]

[a] International Center for Numerical Methods in Engineering, (CIMNE), Edificio C1, 08034 Barcelona, Spain
[b] School of Aerospace Mechanical & Mechatronic Engineering (AMME),University of Sydney, NSW 2006, Australia
[c] School of Engineering System, Queensland University of Technology, Brisbane, Australia
[d] Universitat Politecnica de Catalunya (UPC), 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 22 December 2009
Received in revised form 17 August 2010
Accepted 10 March 2011
Available online 21 March 2011

Keywords:
Multi-Objective Evolutionary Algorithm (MOEA)
Shape optimization
Game Strategies
Hybrid-Game
Nash-equilibrium
Pareto front

ABSTRACT

A number of Game Strategies (GS) have been developed in past decades. They have been used in the fields of economics, engineering, computer science and biology due to their efficiency in solving design optimization problems. In addition, research in multi-objective (MO) and multidisciplinary design optimization (MDO) has focused on developing robust and efficient optimization methods to produce a set of high quality solutions with low computational cost. In this paper, two optimization techniques are considered; the first optimization method uses multi-fidelity hierarchical Pareto optimality. The second optimization method uses the combination of two Game Strategies; Nash-equilibrium and Pareto optimality. The paper shows how Game Strategies can be hybridised and coupled to Multi-Objective Evolutionary Algorithms (MOEA) to accelerate convergence speed and 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-Game Strategies are clearly demonstrated.

1. Introduction

With an ever-increasing complexity in design engineering problems, research of multi-objective (MO) and multidisciplinary design optimization (MDO) focuses on developing robust and efficient optimization methods to produce high quality designs with low computational cost [1–4]. In such situations, Game Strategies (GSs) has proposed as one of the key technologies to save CPU usage and produce high model quality due to their efficiency in design optimization [4,5,7]. In this paper, two GSs are considered and applied to two optimization methods; the first optimization method employs the concept of multi-fidelity hierarchical Pareto optimality evolutionary algorithm (HAPMOEA) [6]. The second method uses a combination of the concepts of Nash-equilibrium [7] and Pareto optimality [8] (herein named Hybrid-Game) coupled to Multi-Objective Evolutionary Algorithm (MOEA). HAPMOEA uses three hierarchical layers with seven populations (Pareto-games) which are divided by model fidelity conditions. Hybrid-Game consists of one Pareto-Player and several Nash-Players and can produce a Nash-equilibrium and Pareto non-dominated solutions simultaneously [9]. The reason of using Nash-game is to use it as pre-conditioner. It speeds up the search for a local solution which will then be seeded to the Pareto-Player to produce global solutions. The hybrid game is especially developed to solve complex design problems such as robust MO/MDO which requires high computational cost.

The evolutionary optimization methods HAPMOEA and Hybrid-Game are coupled to a Multi-Objective Evolutionary Algorithms (MOEA) in an asynchronous parallel computation and are implemented to solve a single-disciplinary multi-objective design and uncertainty based multidisciplinary design problems.

The rest of paper is organized as followed; Section 2 describes the methodologies and presents algorithms for HAPMOEA and Hybrid-Game. Section 3 presents the aerodynamic analysis tools used in this work. Real world MO design problem is conducted in Section 4. Conclusions are presented in Section 5.

2. Methodology

Both methods HAPMOEA and Hybrid-Game have same feature of Multi-Objective Evolutionary Algorithms (MOEAs). HAPMOEA uses hierarchical multi-population Pareto optimality while the

[*] Corresponding author at: Parque Mediterraneo de la Tecnologia (PMT), Edificio C3, Room 312, CIMNE/UPC, C/Esteve Terradas n5, 08860 Castelldefels, Spain. Tel.: +34 934 015 696; fax: +34 934 016 517.

E-mail addresses: ds.chris.lee@gmail.com (D. Lee), felipe.gonzalez@qut.edu.au (L.F. Gonzalez), jperiaux@gmail.com (J. Periaux), k.srinivas@usyd.edu.au (K. Srinivas), onate@cimne.upc.edu (E. Onate).

[1] Tel.: +61 7 3138 1363; fax: +61 7 3138 1469.

[2] Tel.: +34 932 057 016; fax: +34 934 016 517.

[3] Tel.: +61 2 9351 4289; fax: +61 2 9351 4841.

[4] Tel.: +34 932 057 016; fax: +34 934 016 517.

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.03.007

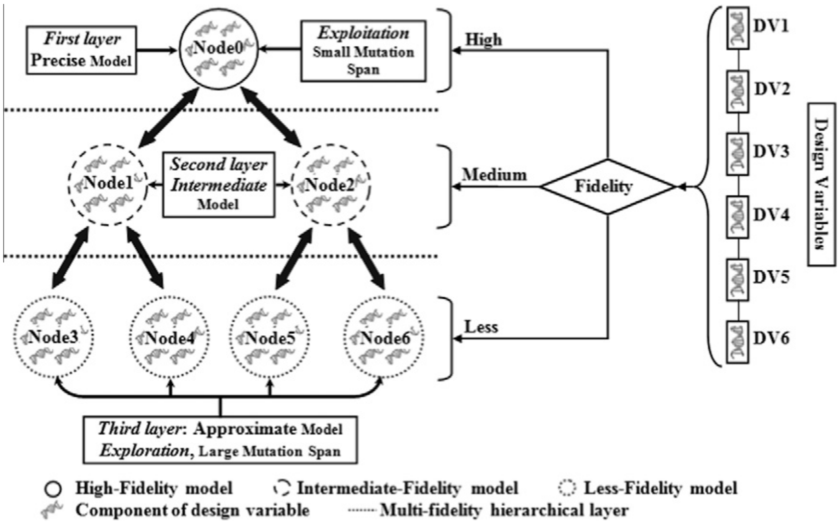
First layer Precise Model
Node0
Exploitation Small Mutation Span
High
Second layer Intermediate Model
Node1
Node2
Medium
Fidelity
Node3
Node4
Node5
Node6
Less
Third layer: Approximate Model Exploration, Large Mutation Span
DV1
DV2
DV3
DV4
DV5
DV6
Design Variables
High-Fidelity model
Intermediate-Fidelity model
Less-Fidelity model
Component of design variable
Multi-fidelity hierarchical layer

Fig. 1. Topology of HAPMOEA.

concepts of Nash-equilibrium and Pareto optimality are implemented for Hybrid-Game.

The evolutionary algorithm used in this paper is based on Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES) [6,10,11]. The optimization methods; HAPMOEA and Hybrid-Game use MOEA coupled to several analysis tools. The methods incorporate the concepts of Covariance Matrix Adaptation (CMA) [12,13], Distance Dependent Mutation (DDM) [11], and implementation of the asynchronous parallel computation [14,15]. The methods couple an MOEA, analysis tools and a statistical design tool that takes uncertainty into account.

2.1. Multi-fidelity (Population) hierarchical Pareto optimality

The first method multi-fidelity hierarchical Pareto optimality [16] uses three layers of hierarchical Pareto optimal game and is shown in Fig. 1. The optimiser has capabilities to handle multi-fidelity/physics models for the solution. There are three layers and seven different populations in HAPMOEA; the first layer (Node0: fine-grid population) concentrates on the refinement of solutions. The third layer (Node3–Node6: four coarse-grid populations) uses approximate model therefore these populations are entirely devoted to exploration. The second layer (Node1 and Node2: two intermediate-grid populations) is to compromise solutions between exploration (third layer) and exploitation (first layer). There is also a migration operation after predefined number of function evaluations; individuals migrate up and down from third to first layer and from first to third layer during the optimization. As an example, if the problem considers six design variables (DV1–DV6); each Pareto-game population at each layer evaluates the same and all fitness/objective function, and considers the whole design variable span (DV1–DV6). The topology of HAPMOEA is normally fixed in the authors' test cases for multi-objective and multidisciplinary designs. Details of HAPMOEA can be found in Ref. [6].

2.2. Hybrid-Game (Hybrid-Nash)

The Hybrid-Game uses the concepts of Nash-Game and Pareto optimality and hence it can simultaneously produce Nash-equilibrium and a set of Pareto non-dominated solutions [9]. The reason for hybridisation of the concepts of Nash-Game and Pareto optimality is to cover each other's major drawbacks; Nash-Game has a fast search but only one global solution (Nash-equilibrium) while standard EAs based on Pareto optimality generally require a large number of function evaluations to find optimal non-dominated solutions. Nash-Game herein speeds up to search one of the global solutions which will be seeded to the Pareto-Game population at every generation or after a predefined sequential function evaluation. Each Nash-Player has its own design criteria and uses its own optimization strategy. Therefore the Hybrid-Game can accelerate MOEA to find Pareto optimal solutions based on the elite design obtained by the Nash-Game.

Fig. 2a shows the hybridisation between HAPMOEA and Nash-Game, and one example of the Hybrid-Game topology. The Hybrid-Game consists of four different populations; three Nash-Players (NP1, NP2 and NP3) and one Pareto-Player/Global-Player (Node0: high fidelity/resolution – precise model from HAPMOEA). The Hybrid-Game locates the Pareto-Player/Global-Player at the core of Nash-Game and the elite designs obtained by Nash-Players will be seeded to the Pareto-Player/Global-Player at every generation or after a predefined number of function evaluations. This optimization mechanism allows the Pareto-Player/Global-Player to accelerate to find Pareto non-dominated solutions or global solutions. The topology of Hybrid-Game represents a top view of a trigonal pyramid instead of a two dimensional hierarchical pyramid shape. The topology of Hybrid-Game is flexible; if there are four Nash-Players then the topology will be a quadrangular pyramid.

Fig. 2b shows an example of design variables (DV1–DV6) distribution in a Hybrid-Game. The distribution of design variables is as follows; Nash-Player1 (black circle) only considers black square design components (DV1, DV4), DV2 and DV5 are considered by Nash-Player2 (hidden line circle). Nash-Player3 (dot line circle) considers DV3 and DV6. The Pareto-Player (center line circle) considers the complete design variable span (DV1–DV6). The elite designs (DV1–DV6) obtained by Nash-Players (N-Player1, N-Player2 and N-Player3) will be seeded to the population of Pareto-Player (P-Player) after each generation or after a predefined number of function evaluations has been reached during the optimization. The Hybrid-Game decomposes one design problem into three simpler design problems for Nash-Game which consists of three Nash-Players (N-Player1, N-Player2 and N-Player3). The Pareto-Player (P-Player) will use elite designs (DV1–DV6) obtained by Nash-Game to find Pareto non-dominated solutions. Therefore,

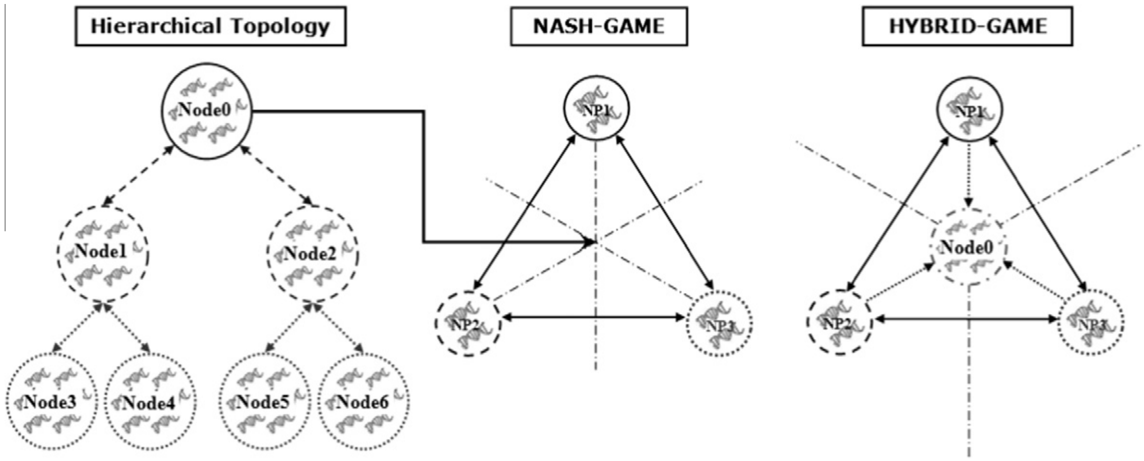

Fig. 2a. Hybridization of MOEA and Hybrid-Game topology.

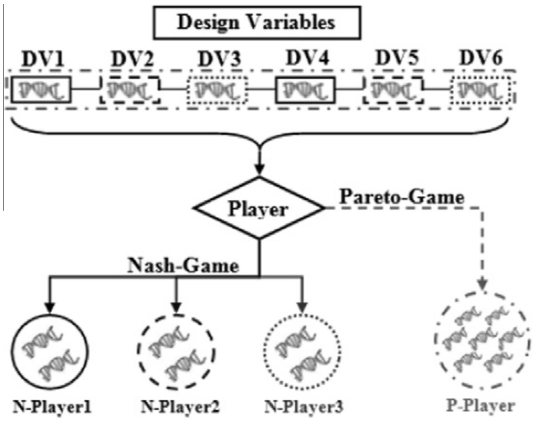

Fig. 2b. Design variables for Hybrid-Game.

Pareto-Player can accelerate its searching diversity to true Pareto front or global solutions. The Hybrid-Game can be applied to non-generational EAs such as HAPMOEA as well as to generation based approaches.

Lee et al. [17] studied the concept of Hybrid-Game (Pareto and Nash) coupled to a well-known MOEA; Non-dominating Sort Genetic Algorithm II (NSGA-II) [18]. Their research shows that the Hybrid-Game improves the NSGA-II performance by 80% when compared to the original NSGA-II. In addition, Lee et al. [19] hybridised NSGA-II with Nash-Game strategy to study a role of Nash-Players in Hybrid-Game by solving multi-objective mathematical test cases; non-uniformly distributed non-convex, discontinuous and mechanical design problem. Their research also shows that HAPMOEA can be also hybridised with Nash-Game to solve a real-world robust multidisciplinary design problem. Numerical results show that the Hybrid-Game improves 70% of HAPMOEA performance while producing better Pareto optimal solutions.

It should be noted that the Nash-equilibrium solution can be one of Pareto front members since the elite designs obtained by Nash-Players are seeded to the Pareto-Player population however the Nash-equilibrium will not be part of the Pareto-front if the Nash-equilibrium is dominated by Pareto-Player solutions. The validation of Hybrid-Game and HAPMOEA can be found in Refs. [9,17,19–21].

2.3. Mathematical benchmarks

This section illustrates the use of Hybrid-Game for two multi-objective mathematical design problems including non-uniformly distributed non-convex design, and discontinuous multi-objective designs. Both NSGA-II and Hybrid-Game applied to NSGA-II are considered.

2.3.1. Non-uniformly distributed non-convex design

This problem defined in Ref. [22] considers a non-uniformly distributed non-convex problem. It is an extended version of a non-linear problem where the objective is to minimise the following equations:

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

$$f_2(x_1, x_2) = g(x_2) \cdot h(f_1(x_1), g(x_2)) \tag{2}$$

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

$$g(x_2) = \begin{cases} 4 - 3\exp\left(-\left(\frac{x_2-0.2}{0.02}\right)^2\right) & \text{if } 0 \leqslant x_2 \leqslant 0.4 \\ 4 - 3\exp\left(-\left(\frac{x_2-0.7}{0.2}\right)^2\right) & \text{if } 0.4 \leqslant x_2 \leqslant 1 \end{cases} \quad h(f_1, g) = \begin{cases} 1 - \left(\frac{f_1}{g}\right)^\alpha & \text{if } f_1 \leqslant g \\ 0 & \text{otherwise} \end{cases} \quad \alpha = 4$$

Random solutions are shown in Fig. 3a. Fig. 3b compares the convergence obtained by NSGA-II and Hybrid-Game coupled to NSGA-II (Hybridised NSGA-II). The optimization is stopped after 50 generations with a population size of 100. It can be seen that the NSGA-II requires more function evaluations (marked with red circle) while the Hybridised NSGA-II has already capture the true Pareto front.

The initial populations obtained by NSGA-II and Hybridised NSGA-II are shown in Figs. 4a and 4b respectively where the NSGA-II has a better non-dominated solutions (nine members) distribution than the Hybridised NSGA-II with seven non-dominated solutions. However, it can be seen that the Nash-Players of Hybridised NSGA-II found the elite solution (mark with circle) which dominates Pareto members 1–5 of Pareto-Game. This Nash elite design will be seeded to the population of Pareto-Game and will become a Pareto member 1 then the Pareto-Game will generate the candidates based on the Nash elite design. This is the reason why the Hybridised NSGA-II had faster convergence than NSGA-II.

2.3.2. Discontinuous MO (TNK) design

The problem TNK proposed in Ref. [23] considers minimisation of the following equation:

$$f_1(x_1) = x_1 \quad \text{and} \quad f_2(x_2) = x_2 \tag{3}$$

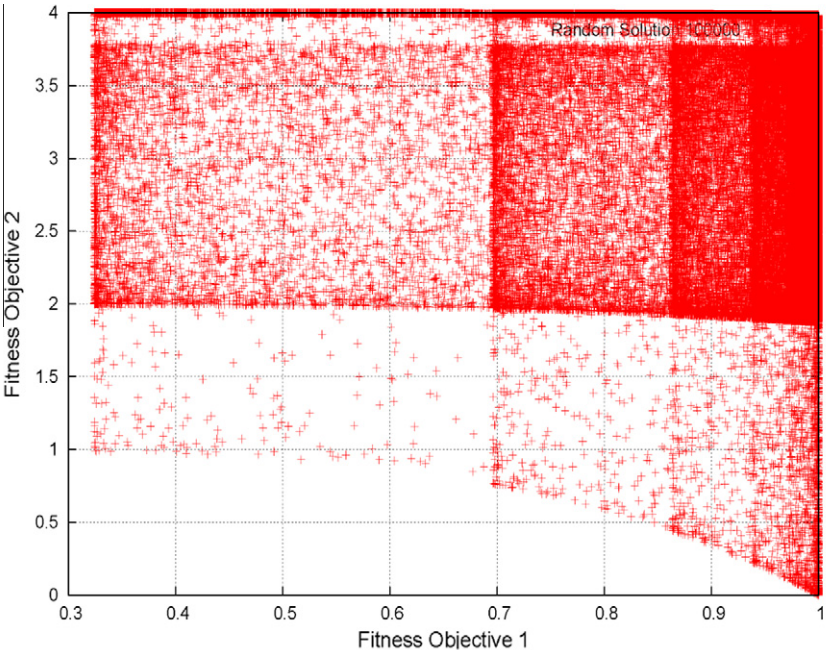

Fig. 3a. Random solutions (100,000 pints).

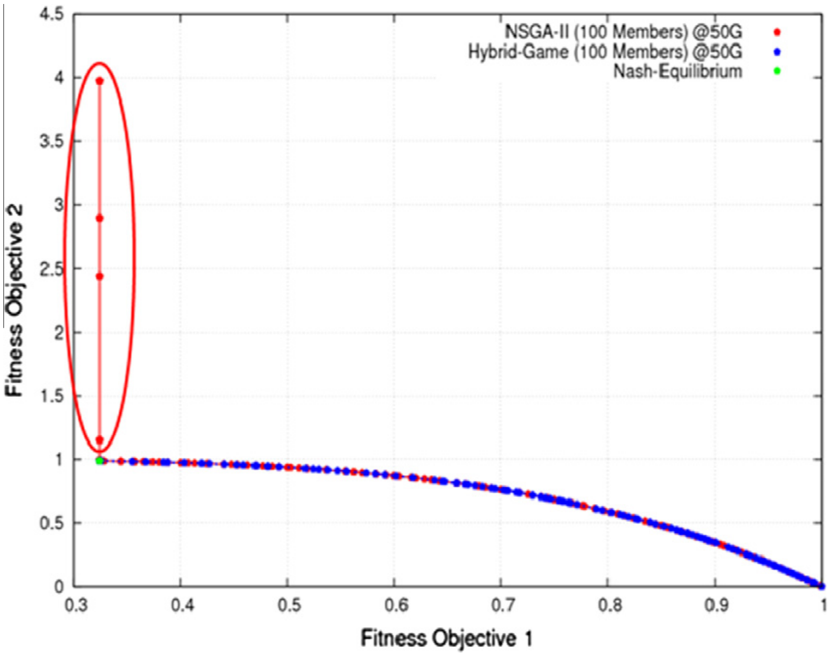

Fig. 3b. Comparison of Pareto fronts.

Subject to

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

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

where $0 \leqslant x_1, x_2 \leqslant \pi$

Random solutions are shown in Fig. 5a. Fig. 5b compares the convergence obtained by NSGA-II and the Hybridised 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 while the Hybrid-Game converged to the true Pareto front.

2.4. Algorithms for HAPMOEA and Hybrid-Game

Algorithms for HAPMOEA and Hybrid-Game are shown in Figs. 6a and 6b to further illustrate the HAPMOEA and the Hybrid-Game concepts. It is assumed that the problem considers the fitness function $f = \min(x_1, x_2, x_3)$ as an example.

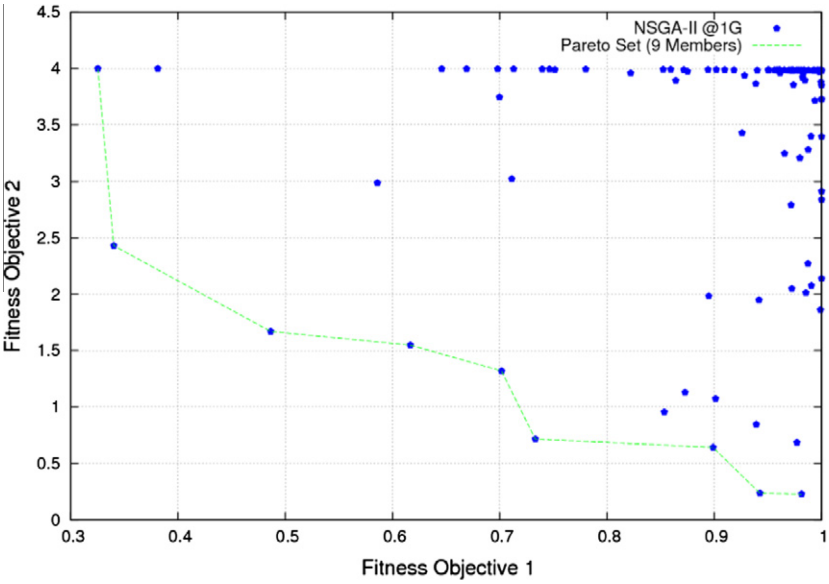

Fig. 4a. Non-dominated solutions (NSGA-II).

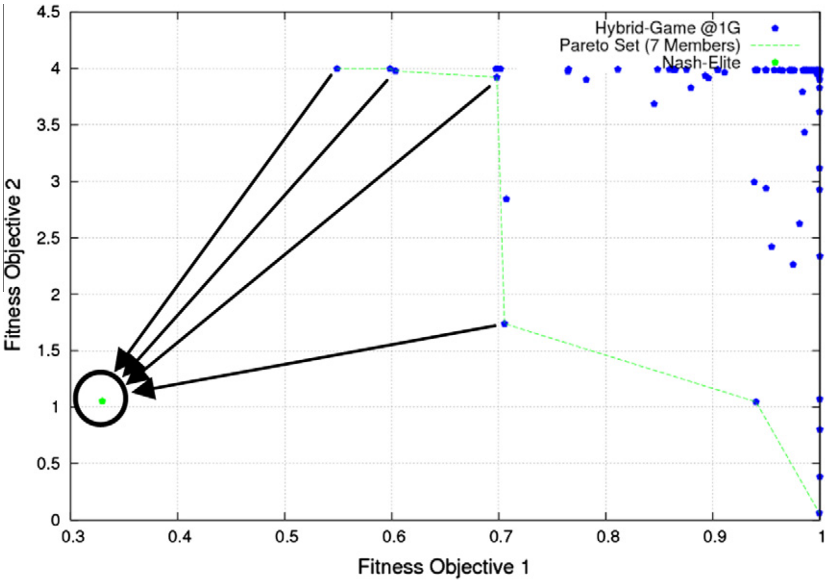

Fig. 4b. Non-dominated solutions (Hybridised NSGA-II).

2.4.1. HAPMOEA-L3 (Fig. 6a)

The method has eight main steps as follows;

- Step 1: Define population size and number of generation for hierarchical topology (Node0–Node6), number of design variables (x_1, x_2, x_3) and their design bounds, model quality for each layer (Layer1 (Node0): precise, Layer2 (Node1, Node2): intermediate, Layer3 (Node3–Node6): least precise).
- Step 2: Initialize seven random populations for Node0–Node6.

while termination condition (generation or elapsed time or predefined fitness value).

- Step 3: Generate offspring using mutation or recombination operations.
- Step 4: Evaluate each offspring and compute fitness functions.
- Step 4.1: Evaluate offspring for each node in terms of precise, compromise, least precise.
- Step 5: Sort each population for each node based on its fitness.
- Step 6: Replace best individual into non-dominated population of each node.

end

- Step 7: Designate results for the optimization; Pareto optimal front obtained by Node0 at first layer (precise model) for multi-objective design problem otherwise plot convergence of optimization based on best-so-far individual.
- Step 8: Do post-optimization process; if problem considers aerodynamic wing design Mach sweep will be plotted corresponding to objective (CD, CL, L/D).

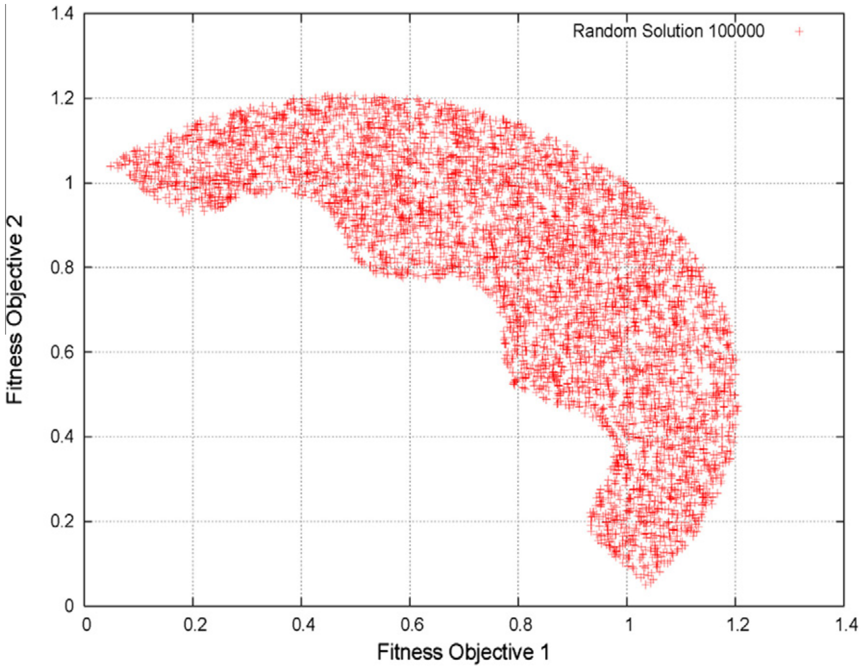

Fig. 5a. Random solutions (100,000 pints).

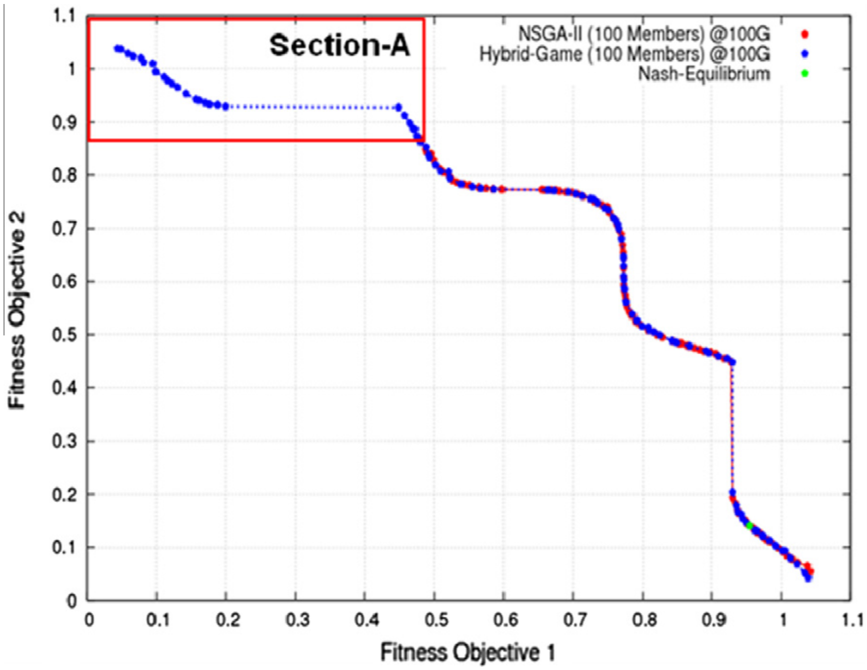

Fig. 5b. Comparison of Pareto fronts for TNK.

2.4.2. Hybrid-Game (Fig. 6b)

The method has eight main steps as follows;

- Step 1: Define population size, number of generation or function evaluations for each Nash-Players (N-Player1, N-Player2, N-Player3) and Pareto Player (P-Player), dimension of decision variables (x_1, x_2, x_3) and their design bounds. Split decision variables for each player (N-Player1: x_1, N-Player2: x_2, N-Player3: x_3, P-Player: x_1, x_2, x_3).
- Step 2: Initialize random population for each player.
- while termination condition (generation or elapsed time or predefined fitness value).
- Step 3: Generate offspring using mutation or recombination operations.
- Step 4: Evaluate each offspring and compute its fitness functions.
- Step 4.1: Evaluate offspring in Nash-Game.
  - N-Player1: Use x_1 with design variables x_2, x_3 fixed by N-Player2 and N-Player3.

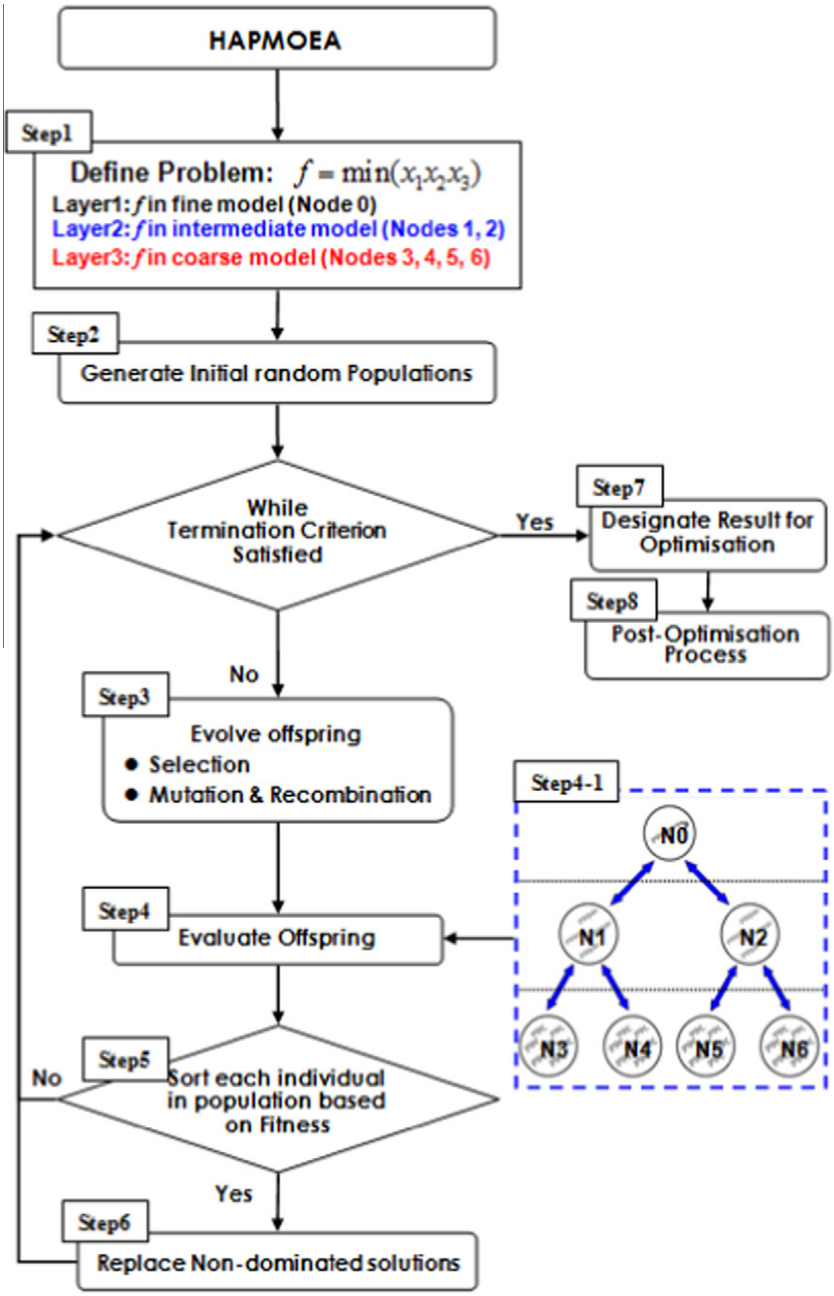
HAPMOEA
Step1
Define Problem: $f = \min(x_1x_2x_3)$
Layer1: f in fine model (Node 0)
Layer2: f in intermediate model (Nodes 1, 2)
Layer3: f in coarse model (Nodes 3, 4, 5, 6)
Step2
Generate Initial random Populations
While Termination Criterion Satisfied
Yes
Step7
Designate Result for Optimisation
Step8
Post-Optimisation Process
No
Step3
Evolve offspring
● Selection
● Mutation & Recombination
Step4-1
N0
N1
N2
N3
N4
N5
N6
Step4
Evaluate Offspring
Step5
Sort each individual in population based on Fitness
No
Yes
Step6
Replace Non-dominated solutions

Fig. 6a. Algorithm of HAPMOEA.

  - N-Player2: Use x_2 with design variables x_1, x_3 fixed by N-Player1 and N-Player3.
  - N-Player3: Use x_3 with design variables x_1, x_2 fixed by N-Player1 and N-Player2.
- Step 4.2: Evaluate offspring for P-Player.
  - if (the first offspring at each generation is considered)
  - P-Player: use elite design (x_1^*, x_2^*, x_3^*) obtained by Nash-Game at Step 4.1.
  - else
  - P-Player: use x_1, x_2, x_3 obtained by mutation or recombination operation as default.
- Step 5: Sort each population for each player based on its fitness.
- Step 6: Replace the non-dominated individual into best population for P-Player.
- end
- Step 7: Designate results.

P-Player: Plot Pareto optimal front for multi-objective design problem otherwise plot convergence of optimization based on best-so-far individual.

Nash-Game: plot Nash-equilibrium obtained by N-Player1, N-Player2, N-Player3.

- Step 8: Do post-optimization process; if problem considers aerodynamic wing design Mach sweep will be plotted corresponding to objective (CD, CL, L/D).

3. Aerodynamic analysis tools

The first aerodynamic analysis tool used in this paper is a potential flow solver. The solver is used for analysing inviscid, isentropic, transonic shocked flow past 3D swept wing configurations [24].

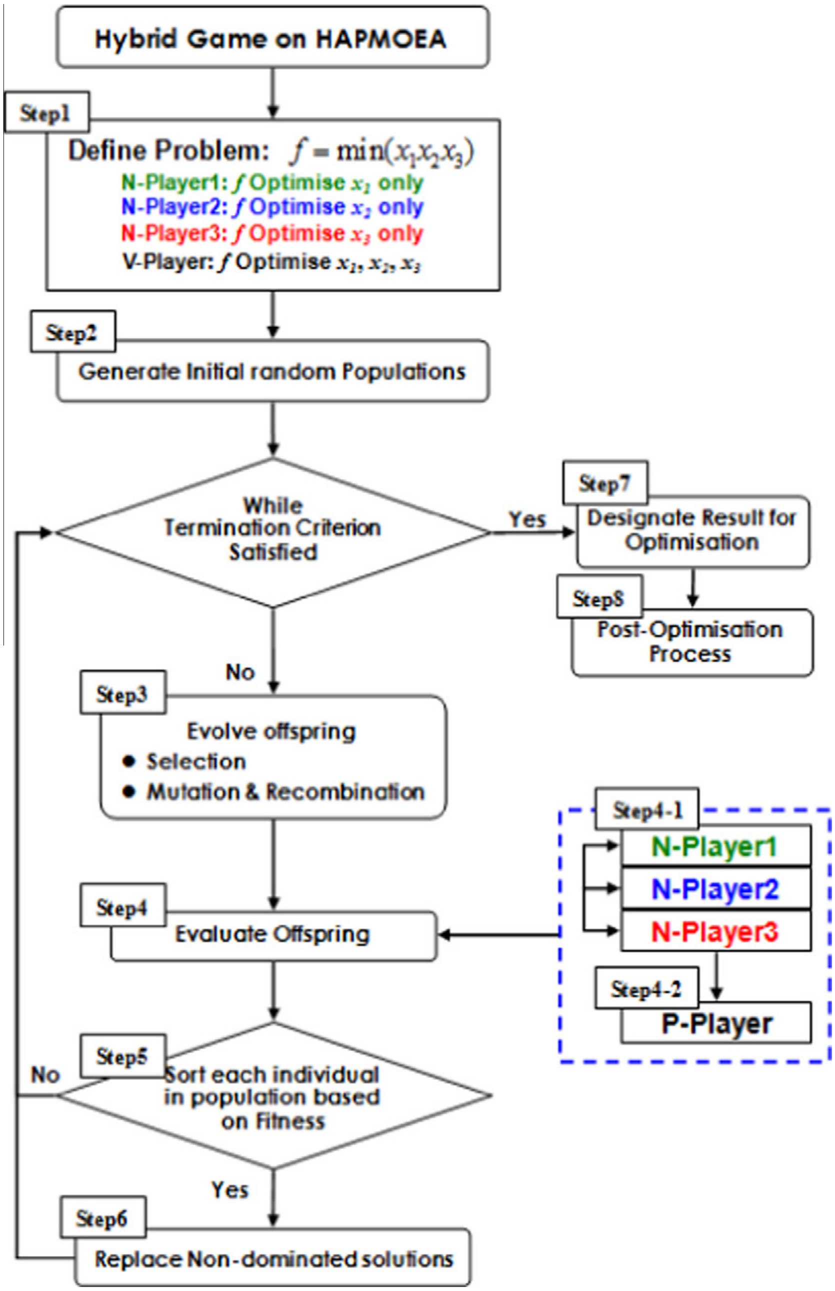
Hybrid Game on HAPMOEA
Step1
Define Problem: $f = \min(x_1 x_2 x_3)$
N-Player1: f Optimise x_1 only
N-Player2: f Optimise x_2 only
N-Player3: f Optimise x_3 only
V-Player: f Optimise x_1, x_2, x_3
Step2
Generate Initial random Populations
While Termination Criterion Satisfied
Yes
Step7
Designate Result for Optimisation
Step8
Post-Optimisation Process
No
Step3
Evolve offspring
Selection
Mutation & Recombination
Step4-1
N-Player1
N-Player2
N-Player3
Step4-2
P-Player
Step4
Evaluate Offspring
Step5
No
Sort each individual in population based on Fitness
Yes
Step6
Replace Non-dominated solutions

Fig. 6b. Algorithm of Hybrid-Game.

The second analysis tool is for friction drag which is externally computed by utilizing the program FRICTION code [25]. The code provides an estimate of the laminar and turbulent the skin friction suitable for use in aircraft preliminary design. Details on the validation of the potential flow solver can be found in Ref. [21] where it is shown that the results obtained by the potential flow solver are in good agreement with experimental data.

4. Real world design problem: MO design optimization of UCAS

In this section, the Hybrid-Game is used to show the benefit of using Nash-game and Pareto-game simultaneously. To do so, the results obtained by Hybrid-Game will be compared to the results obtained by HAPMOEA. It is shown that the Hybrid-Game has the potential to produce high quality solutions and to reduce computational expense.

4.1. Formulation of design problem

Unmanned (Combat) Aerial Systems (UCASs) have broad applicability for a number of civilian and military missions [26,27]. The type of vehicle considered in this section is a Joint Unmanned Combat Air Vehicle (J-UCAV) that is similar in shape to the Northrop Grumman X-47B [28]. This test case considers the design optimization of the UCAV wing aerofoil sections and planform geometry. The objectives are to maximize both mean values of lift coefficient ($\overline{C_L}$) and lift to drag ratio ($\overline{L/D}$) to maximize a manoeuvrability and range of UCAV. The baseline UCAV is shown in Figs. 7a and 7b.

The wing planform shape is assumed as an arrow shape with jagged trailing edge. The aircraft maximum gross weight is approximately 46,396 lb (21,045 kg) and the empty weight is 37,379 lb (16,955 kg). The design parameters for the baseline wing configuration are illustrated in Fig. 7b and Table 1. In this test case, the fuselage is assumed from 0% to 25% of the half span. The crank

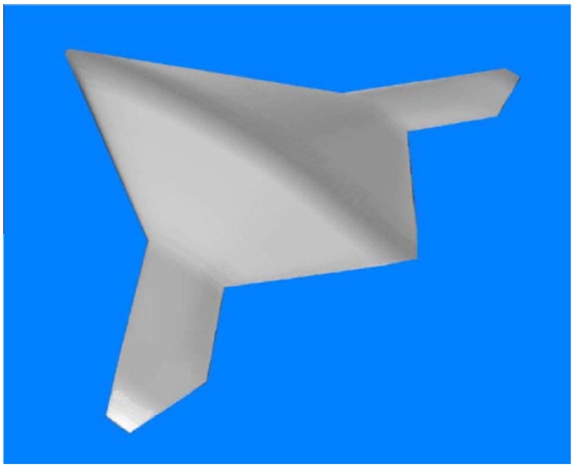

Fig. 7a. Baseline design (3D-view).

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 wing design contains three types of airfoils; the NACA 66-021 and NACA 67-1015 are located at inboard (root and crank1) and the NACA 67-008 are located at the outboard sections (crank2 and tip). These airfoils are shown in Fig. 8. The maximum thickness at root section is 21% of the root chord that is about 3% thicker than X-47B to increase avionics, fuel capacity and missile payloads.

The mission profile of UCAV considered Reconnaissance, Intelligence, Surveillance and Target Acquisition (RISTA) as illustrated in Fig. 9. The mission profile is divided into eight Sectors: where

*Sector*1: Take-off and Climb	*Sector*2: Cruise	*Sector*3: Transition dash	*Sector*4: Ingress
*Sector*5: Target acquisition and strike	*Sector*6: Returning cruise	*Sector*7: End returning cruise	*Sector*8: Descent and Landing

Table 1
Baseline UCAV wing configurations.

AR	b	Λ_{R-C1}	Λ_{C1-C2}	Λ_{C2-T}	λ_{C1}	λ_{C2}	λ_T	$\Gamma_{Overall}$
4.377	18.9 m	55°	29°	29°	20	20	2	0°

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

Fig. 10 shows the weight distribution along the mission profile (*Sector*1–*Sector*8). The weight between *Sector*4 and *Sector*5 is significantly reduced after 80% of ammunition weight is used for target strike.

In this paper, flight conditions for *Sector*2–*Sector*4 are considered for optimization. The minimum lift coefficients ($C_{L_{Minimum}}$) are 0.296 and 0.04 for *Sector*2 and *Sector*4 respectively as shown in Fig. 11. The baseline design produces 30% higher lift coefficient at *Sector*2 when compared to $C_{L_{Minimum}}$ while only 7% higher at *Sector*4. The aim of this optimization is the improvement of aerodynamic performance (L/D) at *Sector*4 while maintaining aerodynamic performance at *Sector*2.

4.2. Representation of design variables

The problem considers design variables for wing and aerofoil design parameters. 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 as illustrated in Figs. 12a–12d.

The wing planform shape is parameterised by considering the variables described in Fig. 13 and their design bounds are shown in Table 2. Three wing section areas, three sweep angles and two taper ratios are considered. These wing design parameters result in different span length (b) and Aspect Ratio (AR) for each candidate. One constraint is that the taper ratio at crank2 should not be higher than the taper ratio at crank1, i.e. ($\lambda_{C2} \leqslant \lambda_{C1}$).

4.3. Hybrid-Game (Pareto + Nash) setup

In this paper, Hybrid-Game employs five Nash-Players and one Pareto-Player as shown in Table 3. The Pareto-Player of Hybrid-Game uses all 76 design variables for the aerofoil sections and wing planform. Aerofoil sections at root, crank1, crank2 and tip are opti-

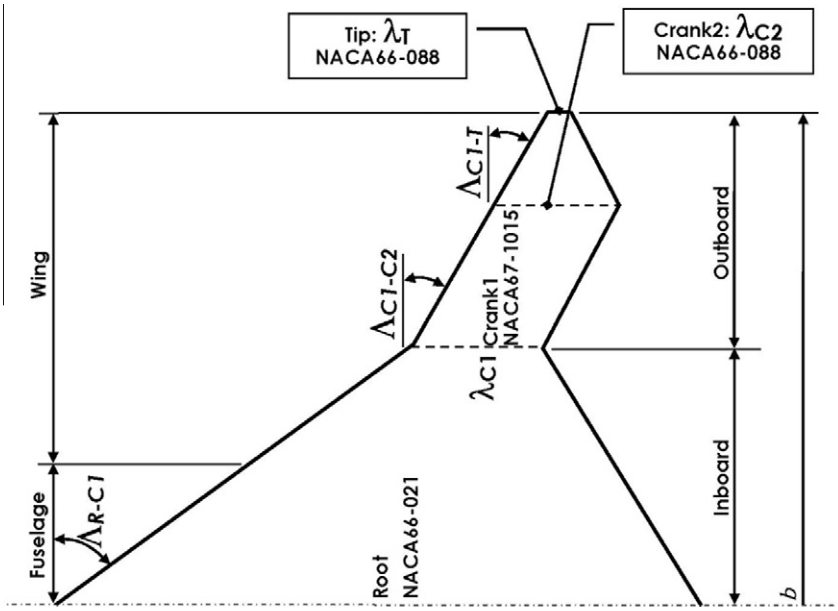

Fig. 7b. Baseline UCAV configuration.

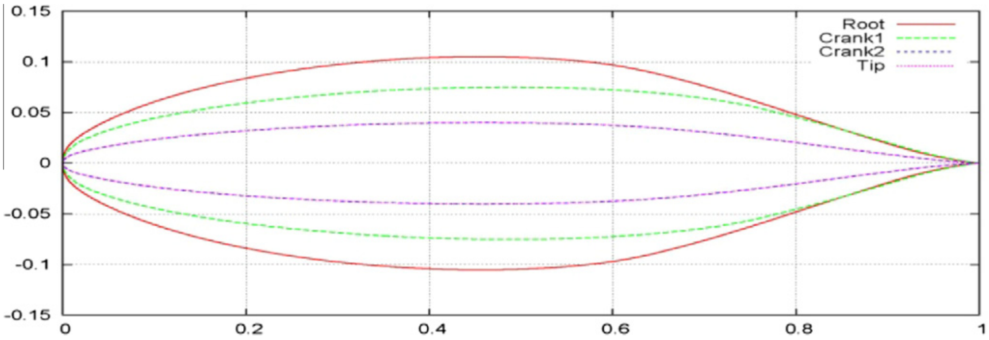

Fig. 8. Baseline UCAV wing airfoil sections.

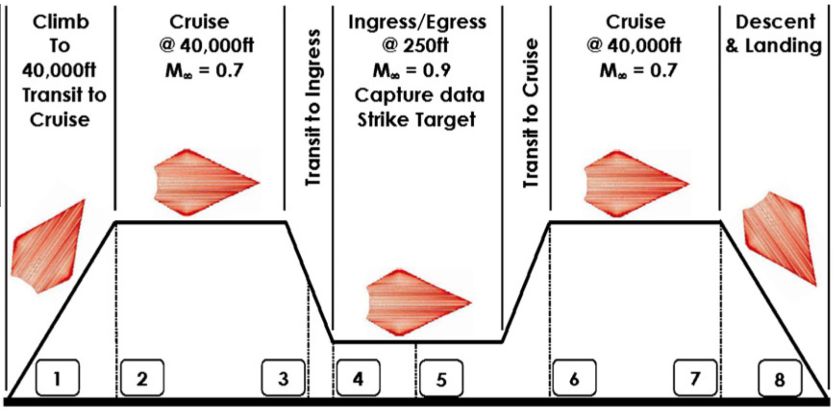
Climb To 40,000ft Transit to Cruise
Cruise @ 40,000ft M∞ = 0.7
Transit to Ingress
Ingress/Egress @ 250ft M∞ = 0.9 Capture data Strike Target
Transit to Cruise
Cruise @ 40,000ft M∞ = 0.7
Descent & Landing
1
2
3
4
5
6
7
8

Fig. 9. Mission profile of baseline UCAV.

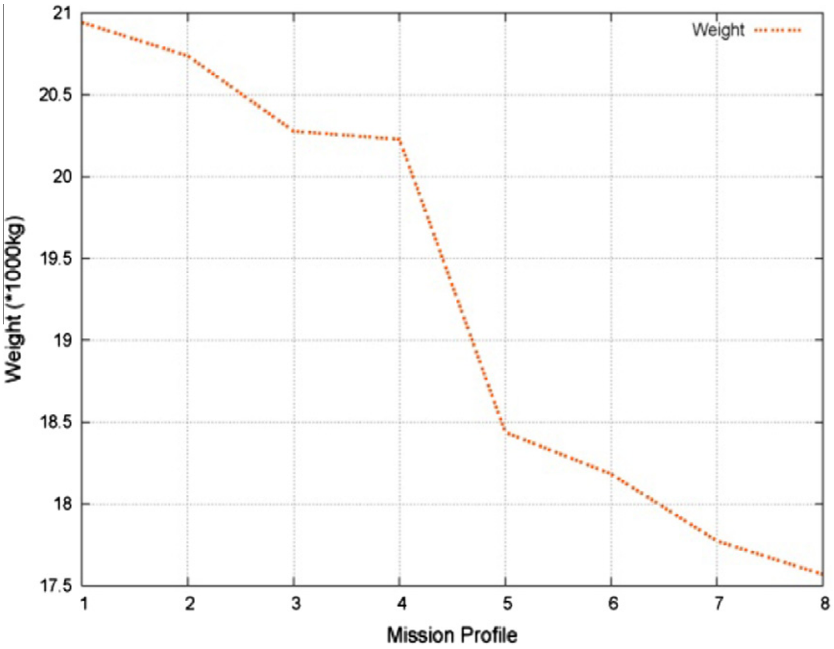

Fig. 10. Weight distribution along the mission.

mized by Nash-Players 1–4 (4 × 17 design variables) while Nash-Player5 optimizes wing planform only (eight design variables). In contrast, each node (Node0–6) of HAPMOEA considers all 76 design variables for the aerofoil sections and wing planform.

4.4. Fitness functions

Table 4 shows the fitness functions considered by the Pareto-Player and Nash-Players. The Pareto-Player considers two fitness

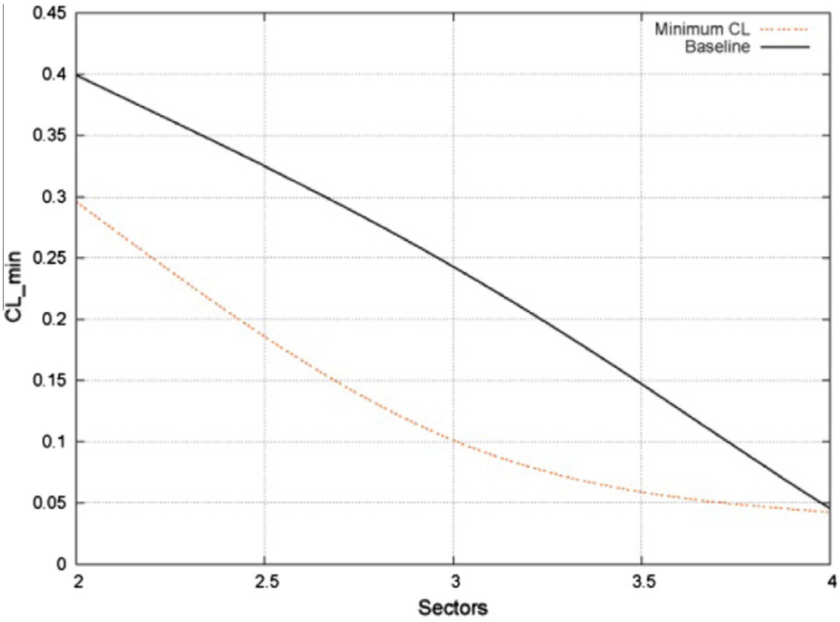

Fig. 11. $C_{L_{Min}}$ for *Sector*2–*Sector*4.

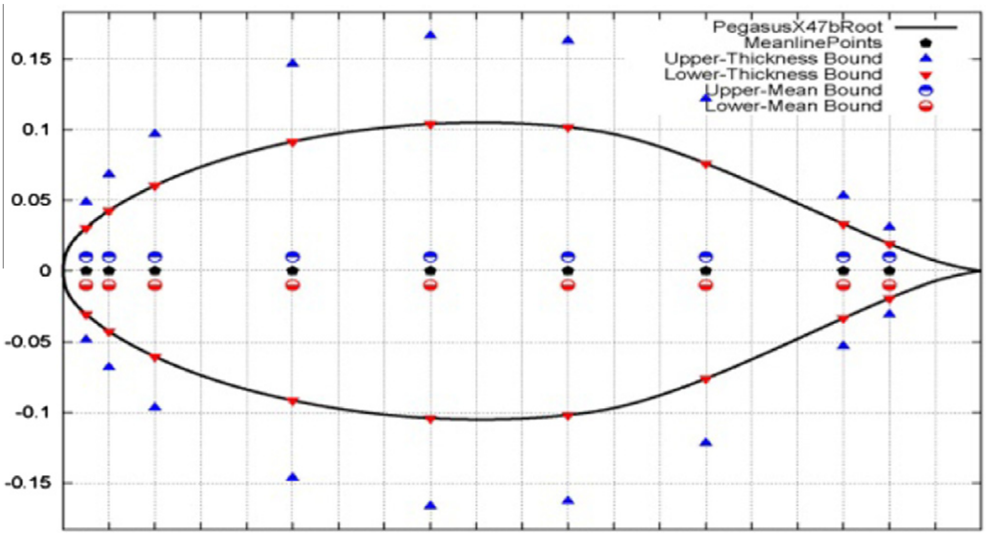

Fig. 12a. Control points at root section.

functions to maximize mean C_L and mean L/D. The five Nash-Players consider one fitness function to optimize aerofoil sections and wing planform shape. In practice, the Nash-Players look for their elite/best aerodynamic shape for aerofoil sections and wing planform design parameters and these elite design parameters will be seeded to the Pareto-Player to accelerate its convergence. Each node (Node0–Node6) of HAPMOEA uses two fitness functions of Pareto-Player.

The flight conditions are;

- *Sector*2: $M_\infty = 0.7$, $\alpha = 6.05^\circ$ and altitude 40,000 ft. *Sector*4: $M_\infty = 0.9$, $\alpha = 0.5^\circ$ and altitude 250 ft.

4.5. Interpretation of numerical results

Both HAPMOEA and Hybrid-Game were run using two 2.4 GHz processors. The HAPMOEA algorithm was allowed to run approximately for 6667 function evaluations and took 200 h. The Hybrid-Game (Hybrid MOEA) algorithm was run approximately for 1300 function evaluations and took 50 h. The Pareto fronts obtained by HAPMOEA and Hybrid-Game are compared to the baseline design in Fig. 14. It can be seen that Hybrid-Game produces better non-dominated solutions in terms of fitness function 1 and 2 when compared to HAPMOEA.

Table 5 compares the Pareto optimal solutions obtained by HAPMOEA and Hybrid-Game. It can be seen that Hybrid-Game produces twice the value of the lift coefficient (C_L) while slightly better results for inverse mean lift to drag ratio (L/D) when compared to the Pareto members obtained by HAPMOEA.

Table 6 compares the quality of drag coefficient obtained by HAPMOEA and Hybrid-Game using the uncertainty mean and variance statistical formulas. It can be seen that Pareto members of Hybrid-Game produces lower drag at [*Sector*2:*Sector*4] while Pareto members from HAPMOEA produce stable drag.

The Sector sweep is plotted with the lift coefficient and lift to drag ratio as shown in Figs. 15a and 15b. The range of Sector sweep

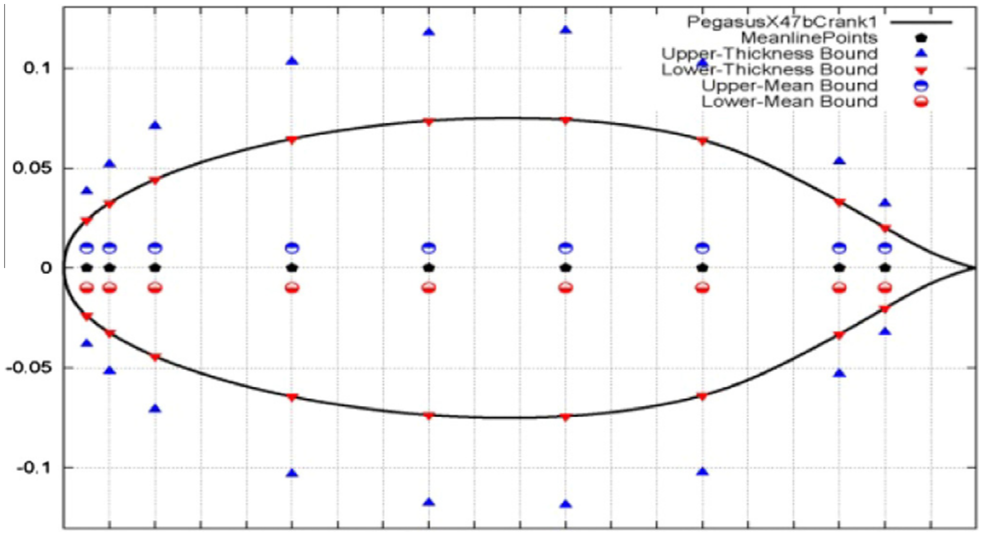

Fig. 12b. Control points at crank1 section.

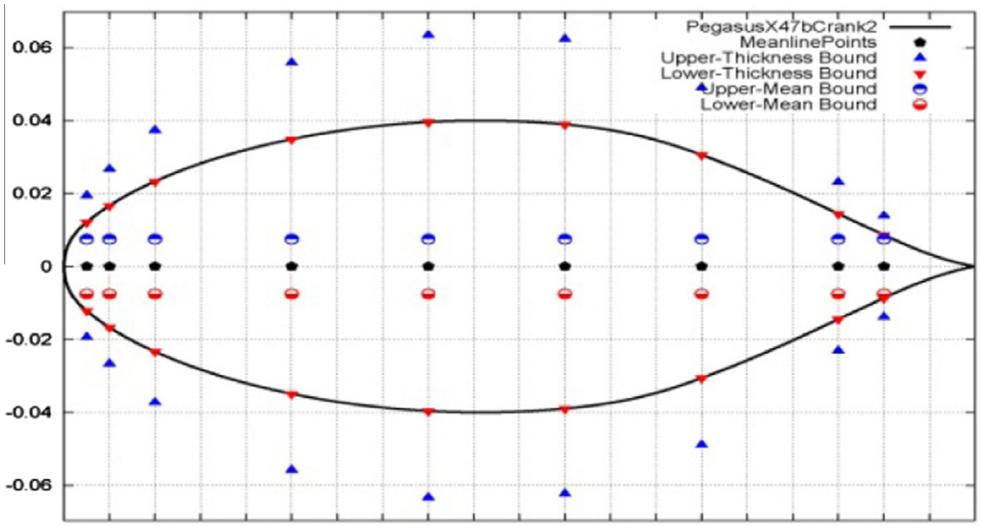

Fig. 12c. Control points at crank2 section.

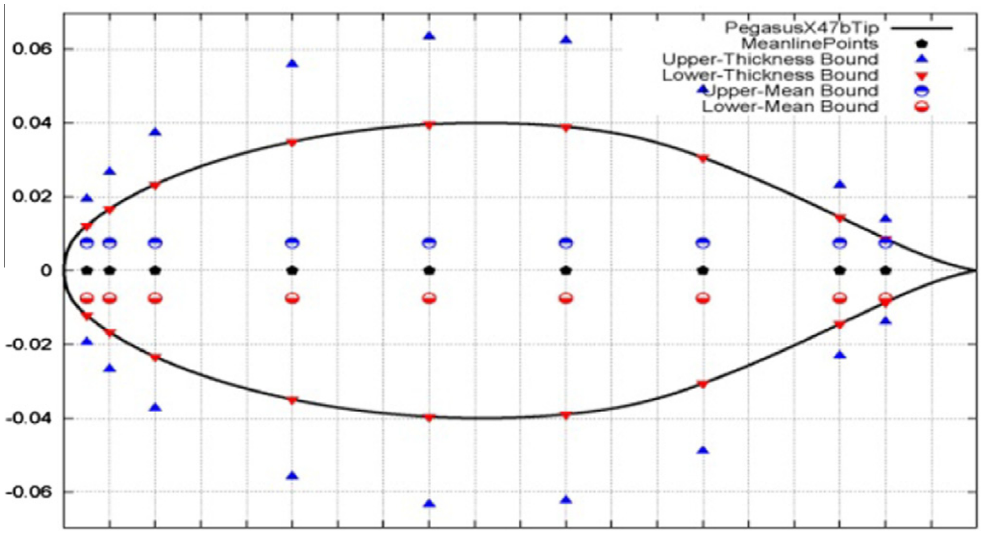

Fig. 12d. Control points at tip section.

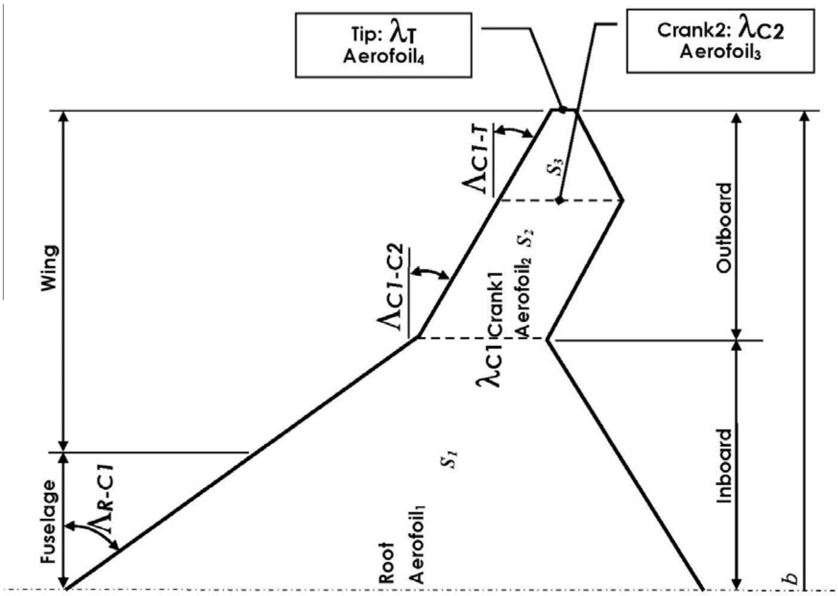

Fig. 13. Wing planform design variables.

Table 2
Wing planform design bounds.

Variables bounds	S_1	S_2	S_3	Λ_{R-C1} (°)	Λ_{C1-C2} (°)	Λ_{C2-T} (°)	λ_{C1}	λ_{C2}
Lower	50.46	10.09	5.05	49.5	25	25	0.15	0.15
Upper	63.92	16.82	10.09	60.5	35	35	0.45	0.45

Note: Area (S) is in m^2 and one geometrical constraint is applied $\lambda_{C2} \leqslant \lambda_{C1}$.

Table 3
Distribution of design variables for Hybrid-Game and HAPMOEA.

Design variables	Hybrid-Game on HAPMOEA						HAPMOEA
	N-P1	N-P2	N-P3	N-P4	N-P5	P-Player	
Aerofoil$_{\text{Root}}$ (17 DVs)	√					√	√
Aerofoil$_{\text{Crank1}}$ (17 DVs)		√				√	√
Aerofoil$_{\text{Crank2}}$ (17 DVs)			√			√	√
Aerofoil$_{\text{Tip}}$ (17 DVs)				√		√	√
Wing Planform (8 DVs)					√	√	√

Note: N-P$_i$ represents ith Nash-Player and P-Player indicates the Pareto-Player.

Table 4
Fitness functions for players of Hybrid-Game.

Player	Fitness function	Optimization criteria
Pareto-Player	$fitness(f_{Pareto}) = \min(1/(\overline{C_L}))$ $fitness(f_{Pareto}) = \min(1/(\overline{L/D}))$	Optimize wing planform and aerofoil sections at root, crank1, crank2 and tip to maximize $\overline{C_L}$ and$\overline{L/D}$
Nash-Player1	$fitness(f_{NP1}) = \min(1/(\overline{C_L}))$	Maximize total wing $\overline{C_L}$ using design variables for Aerofoil$_{\text{Root}}$ only, all other design variables are fixed
Nash-Player2	$fitness(f_{NP2}) = \min(1/(\overline{C_L}))$	Maximize total wing $\overline{C_L}$ using design variables for Aerofoil$_{\text{Crank1}}$ only, all other design variables are fixed
Nash-Player3	$fitness(f_{NP3}) = \min(1/(\overline{C_L}))$	Maximize total wing $\overline{C_L}$ using design variables for Aerofoil$_{\text{Crank2}}$ only, all other design variables are fixed
Nash-Player4	$fitness(f_{NP4}) = \min(1/(\overline{C_L}))$	Maximize $\overline{C_L}$ using design variables for Aerofoil$_{\text{Tip}}$ only, other design variables are fixed
Nash-Player5	$fitness(f_{NP5}) = \min(1/(\overline{L/D}))$	Maximize total wing $\overline{L/D}$ using design variables for wing planform only, all other design variables are fixed

Note: $\overline{C_L} = \frac{1}{2}(C_{L_{Sector2}} + C_{L_{Sector4}})$ and $\overline{L/D} = \frac{1}{2}(L/D_{Sector2} + L/D_{Sector4})$.

is $M_\infty \in [0.7{:}0.9]$, $\alpha \in [6.05°{:}0.5°]$ and altitude $(ft) \in [40{,}000{:}250]$. Pareto fronts obtained by HAPMOEA and Hybrid-Game produce higher C_L and L/D when compared the baseline design. Pareto members from both optimization techniques produce similar

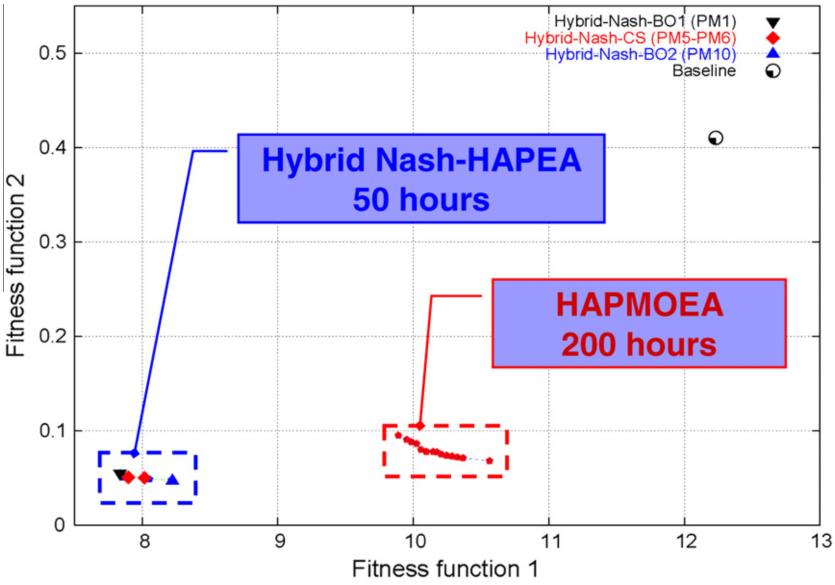

Fig. 14. Comparison of Pareto optimal front obtained by Hybrid-Game and HAPMOEA.

Table 5
Comparison of fitness values obtained by HAPMOEA and Hybrid-Game.

Objective	Baseline design	HAPMOEA (200 h)			*Hybrid-Game (50 h)*		
		PM1 (BO1)	PM6 (CS)	PM15 (BO2)	*PM1 (BO1)*	*PM6 (CS)*	*PM10 (BO2)*
$1/(\overline{C_L})$	12.232	9.890 (−19%)	10.096 (−17%)	10.562 (−14%)	*7.836 (−36%)*	*8.017 (−34%)*	*8.223 (−32%)*
$1/(\overline{L/D})$	0.410	0.095 (−77%)	0.078 (−81%)	0.068 (−83%)	*0.054 (−87%)*	*0.050 (−88%)*	*0.046 (−89%)*

Note: BO represents the best objective solution and CS stands for the compromised solution.

Table 6
Comparison of $C_{D_{Quality}}$ obtained by HAPMOEA and Hybrid-Game.

Objective	Baseline design	HAPMOEA (200 h)			*Hybrid-Game (50 h)*		
		PM1 (BO1)	PM6 (CS)	PM15 (BO2)	*PM1 (BO1)*	*PM6 (CS)*	*PM10 (BO2)*
$\overline{C_D}$	12.232	0.025	0.011 (−56%)	0.009 (−64%)	*0.009 (−64%)*	*0.009 (−64%)*	*0.0089 (−64%)*
δC_D	0.410	5.49 × 0–5	1.49×10^{-5}	1.56×10^{-5}	2.11×10^{-5}	2.29×10^{-5}	2.24×10^{-5}

Note: Quality is represented by mean (performance) and variance (sensitivity/stability).

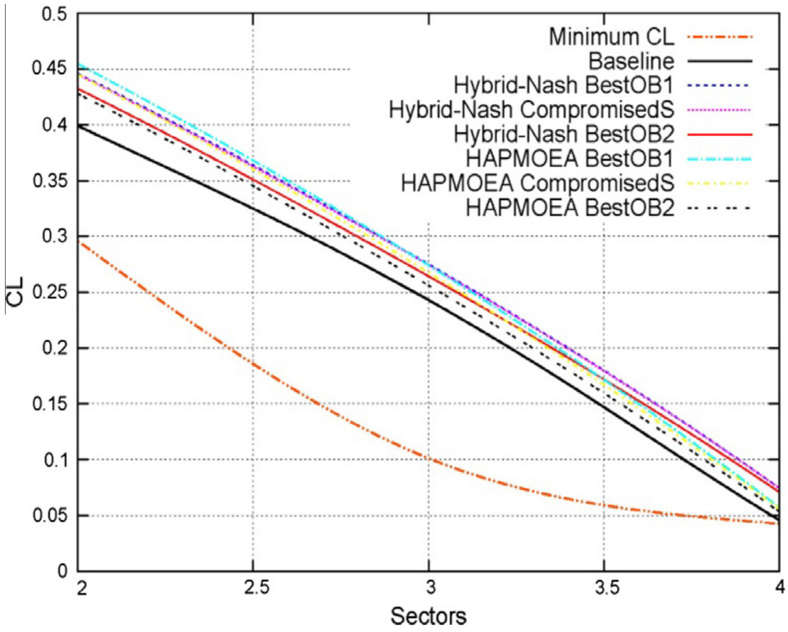

Fig. 15a. C_L vs. Sectors: Hybrid-Nash represents Hybrid-Game.

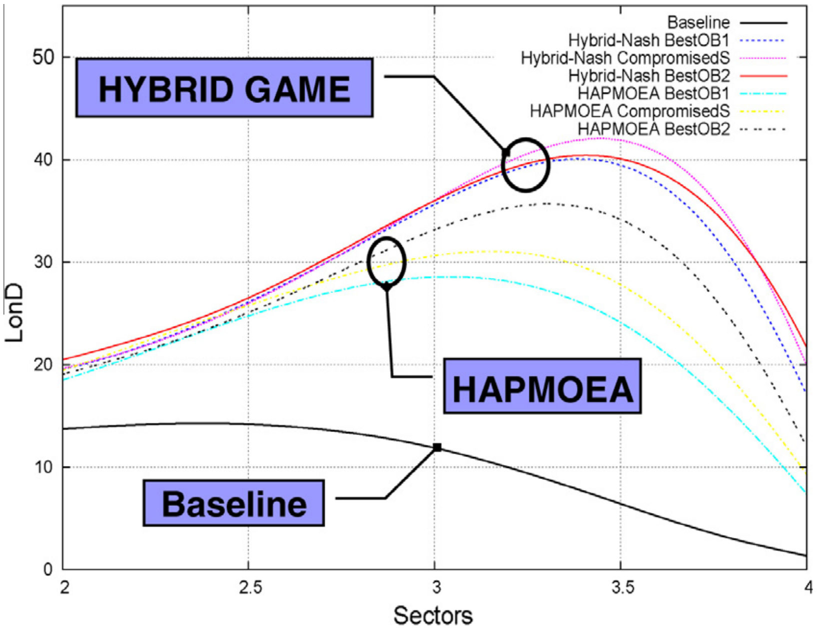

Fig. 15b. *L*/*D* vs. Sectors: Hybrid-Nash represents Hybrid-Game.

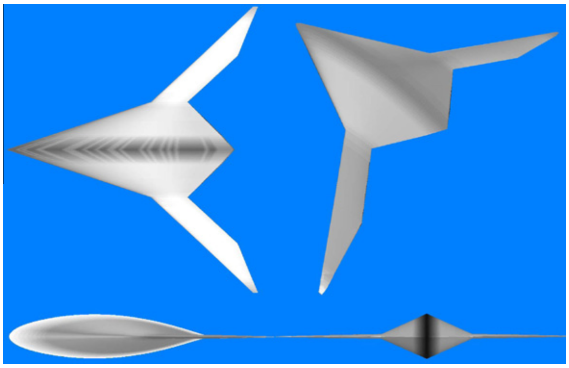

Fig. 16a. Pareto member 5 obtained by (HAPMOEA).

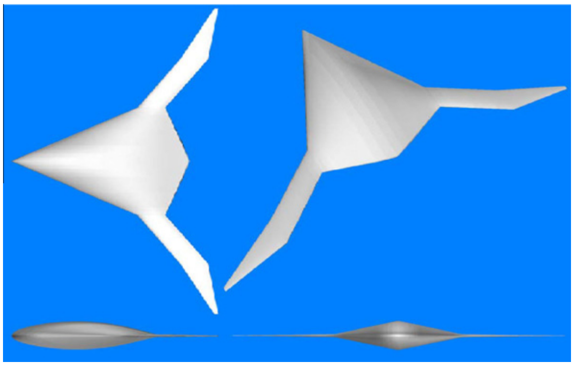

Fig. 16b. Pareto member 6 obtained by (Hybrid-Game).

results for C_L and L/D at *Sector*2 however the Pareto non-dominated solutions from Hybrid-Game produce better C_L and L/D at *Sector*4.

The top, side, front and 3D view of compromised model from HAPMOEA (Pareto member 5) and Hybrid-Game (Pareto member 6) are shown in Figs. 16a and 16b. Even though the Hybrid-Game spent less computational time when compared to HAPMOEA, both compromised solutions are geometrically similar.

4.6. Summary

This paper explored the optimization methods: HAPMOEA and Hybrid-Game (Hybridised MOEA) for solving multi-objective design optimization UCAV. HAPMOEA has been hybridised by using Nash-Game without hierarchical multi-population/fidelity topology: Hybrid-Game. Numerical results show that by introducing Nash-Game as a companion optimizer to help or guide the multi-objective evolutionary optimizer to capture a fast Pareto non-dominated front. It is also shown that the Hybrid-Game (Hybridised MOEA) reduces the computational cost while generating better Pareto front when compared to HAPMOEA. The reason why Hybrid-Game has superiority in terms of optimization efficiency is that the Nash-Game decomposes one complex multi-objective design problem into five simpler single-objective design problems corresponds to five Nash-Players.

5. Conclusions

The optimization methods HAPMOEA and Hybrid-Game were demonstrated and they were implemented to solve multi-objective design problem. It is shown that both methods produce a set of useful Pareto non-dominated solutions. It was also shown that the coupling of Pareto optimality and Nash-Game; Hybrid-Game has superiority on both computational efficiency and solution quality when compared to HAPMOEA. Both methodologies couple a robust MOEA, with aerodynamic analysis tools. A family of Pareto optimal designs obtained by both HAPMOEA and Hybrid-Game provides a selection to the design engineers to proceed into more detail phases of the design process. Even though the numerical results of the methods show the simultaneous improvement in UCAV aerodynamic performance on C_L and L/D there is a fluctuation between *Sector*2–*Sector*3 and *Sector*3–*Sector*4 which can cause flight control or structural failures. This fluctuation can be avoided by using uncertainty design technique during optimization. Future work will focus on coupling Hybrid-Game and high fidelity aerodynamics and electromagnetic analysis tools under uncertainty in flight conditions and electromagnetic frequencies.

Acknowledgements

The authors gratefully acknowledge E.J. Whitney and M. Sefrioui Dessault Aviation for discussions on Hierarchical EAs. We would like to acknowledge A. Jameson and David A. Caughy for granting the potential flow software, and W.H. Mason for the use of FRICTION.

References

[1] Tang Z, Périaux J, Désidéri J-A. Multi criteria robust design using adjoint methods and game strategies for solving drag optimization problems with uncertainties. In: East west high speed flow fields conference 2005, Beijing, China, 19–22 October, 2005. p. 487–93.
[2] Taguchi G, Chowdhury S. Robust engineering. New York: McGraw-Hill; 2000.
[3] Lee DS, Gonzalez LF, Periaux J, Srinivas K. Evolutionary optimisation methods with uncertainty for modern multidisciplinary design in aeronautical engineering, Notes on numerical fluid mechanics and multidisciplinary design (NNFM 100), 100 volumes NNFM and 40 years numerical fluid mechanics. Heidelberg: Springer-Berlin; 2009. p. 271–84. ISBN 978-3-540-70804-9 [chapter 3.].
[4] Periaux J, Mantel B, Chen HQ. Genetic algorithms applied to domain decomposed flow computations. In: D'esid'eri R-A, Hirsch C, Tallec P, Pandolfi M, Periaux J, editors. Computational fluid dynamics '96 proc third ECCOMAS computational fluid dynamics conference, Paris France, 1996. p. 866–70.
[5] Periaux J, Chen HQ, Mantel B, Sefrioui M, Sui HT. Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. Finite Elem Anal Des 2001;5(5):417–29. ISSN:0168-874X.
[6] Lee DS, Gonzalez LF, Whitney EJ. Multi-objective, multidisciplinary multi-fidelity design tool: HAPMOEA – user guide; 2007.
[7] Sefrioui M, Periaux J. Nash genetic algorithms: examples and applications. In: Proceedings of the 2000 congress on evolutionary computation CEC00. IEEE Press, La Jolla Marriott Hotel La Jolla, California, USA, 2000. p. 509–16. ISBN:0-7803-6375-2.
[8] Deb K. Multi-objective evolutionary algorithms: introducing bias among pareto-optimal solutions. KanGAL report no. 99002. Kanpur Genetic Algorithms Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India, 1999.
[9] Lee DS. Uncertainty based multiobjective and multidisciplinary design optimization in aerospace engineering, The Univ. of Sydney, Sydney, NSW, Australia, section 10.7, 2008. p. 348–70.
[10] Koza J. Genetic programming II. Massachusetts Institute of Technology, 1994.
[11] Michalewicz Z. Genetic algorithms + data structures = evolution programs. Artificial intelligence. Springer-Verlag; 1992.
[12] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies. Evolut Comput 2001;9(2):159–95.
[13] Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolut Comput 2003;11(1):1–18.
[14] Wakunda J, Zell A. Median-selection for parallel steady-state evolution strategies. In: Schoenauer Marc, Deb Kalyanmoy, Rudolph Günter, Yao Xin, Lutton Evelyne, Julian Juan Merelo, editors. Parallel problem solving from nature – PPSN VI. Berlin; Springer; 2000. p. 405–14.
[15] Van Veldhuizen DA, Zydallis JB, Lamont GB. Considerations in engineering parallel multiobjective evolutionary algorithms. IEEE Trans Evolut Comput 2003;7(2):144–73.
[16] Sefrioui M, Périaux J. A hierarchical genetic algorithm using multiple models for optimization. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, et al., editors. Parallel problem solving from nature, PPSN VI. Springer; 2000. p. 879–88.
[17] Lee DS, Gonzalez LF, Periaux J. UAS Mission Path Planning System (MPPS) Using hybrid-game coupled to multi-objective design optimizer. J Dyn Syst- T-ASME 2010; 132(4):041005-1–11. ISSN 00220434.
[18] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 2002;6(2):182–97.
[19] Lee DS, Gonzalez LF, Periaux J, Srinivas K. Hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering. IEEE Trans Evolut Comput, in press, TEVC-00213-2009.
[20] Lee DS, Gonzalez LF, Srinivas K, Periaux J. Robust evolutionary algorithms for UAV/UCAV aerodynamic and RCS design optimisation. Int J Comput Fluids 2008;37(5):547–64. ISSN: 0045-7930.
[21] Lee DS, Gonzalez LF, Srinivas K, Periaux J. Robust design optimisation using multi-objective evolutionary algorithms. Int J Comput Fluids 2008;37(5):565–83. ISSN:0045-7930.
[22] Deb K. Multi-objective optimization using evolutionary algorithms. Chichester (UK): Wiley; 2001.
[23] Deb K. Nonlinear goal programming using multi-objective genetic algorithms. J Operat Res Soc 2001;52(3):291–302.
[24] Jameson A. Caughey DA, Newman PA, Davis RM. NYU transonic swept-wing computer program – FLO22, Langley Research Center, 1975.
[25] Mason W. Applied computational aerodynamics. Appendix D: programs, Tuesday, January 21, 1997.
[26] Geer H, Bolkcom C. Unmanned aerial vehicles: background and issues for congress. CRC report for congress, 2005.
[27] Francis MS. AUVSI's unmanned systems program review. Association for Unmanned Vehicle Systems International, Washington, DC, 2005.
[28] Northrop Grumman. <http://www.is.northropgrumman.com/systems/nucasx47b.html>.