MISSION OPTIMISATION AND MULTI-DISCIPLINARY DESIGN OF HYBRID UNMANNED AERIAL SYSTEMS (UAS) USING ADVANCED NUMERICAL TECHNIQUES

Jane Y. Hung, Luis F. González, Rodney A. Walker
Australian Research Centre for Aerospace Automation (ARCAA)
Queensland University of Technology
Brisbane, Queensland, Australia

Keywords: *Hybrid Unmanned Aerial Systems (UAS), Multi-Disciplinary Design and Simulation, Evolutionary Algorithms, Mission Optimisation*

Abstract

This paper describes the theory and practical application of Hierarchical Synchronous Parallel Multi-objective Evolutionary Algorithms (HAPMOEA) for mission optimisation of Hybrid Powered Unmanned Aerial Systems (HPUAS). A real design or simulation will have more than one objective such as minimising fuel consumption, drag and/or time to complete the mission. It is usually the case that the problem is highly non-linear and non-differentiable. New techniques are required, and one of such techniques, even though computationally more intensive than gradient-based methods, are Evolutionary Algorithms (EAs). This paper describes the development and implementation of the hybrid-powered UAV and the coupling of an advanced EA methodology with simulation analysis tools. Result will indicate the practicality and robustness of the method in finding optimal solutions and Pareto trade-offs between fuel consumption and time to complete the mission of a hybrid UAS by producing a set of non-dominated trajectories and mission from which the designer can choose.

1 Introduction

Unmanned Aerial Systems (UAS) are becoming important military and commercial assets for diverse applications, ranging from reconnaissance and surveillance, to aid relief and monitoring tasks [1]. These vehicles are now available in a broad size and capability range and are intended to fly in regions where the presence of onboard human pilots is either too risky or unnecessary.

Civilian applications for UAS technology are quickly emerging as a large and lucrative new aerospace market. Examples of civilian applications include coastal surveillance, power-line inspection, traffic monitoring, bush-fire monitoring, precision farming and remote-sensing.

A large portion of current UAS propulsion systems use either Internal Combustion Engines (ICE) or Electric Motor (EM) powerplants, often sourced commercially off the shelf (COTS) and then modified. Some developers have obtained excellent endurance from these modified COTS aeromodel ICE powerplants, but these units remain inflexible for a range of operational requirements such as manual starting, no in-flight restart and fixed pitch propeller design.

[2] shows that at the cost of an increased weight penalty, a suitable hybridisation of ICE and EM powerplants can lead to overall improvements in range, endurance, and payload capacity whilst simultaneously allowing greatly enhaced UAS operational flexibility.

The multi-physics aspects of these vehicles can benefit from alternative approaches for design and optimisation [3, 4].

2 UAS Mission Optimisation

UAS are being developed for environmental and agricultural purposes such as weather forecast, storm detection, bushfire detection, farm field seeding and aerobiological sampling. All these scenarios involve a common task currently carried out by a human operator: Mission Planning.

Traditionally optimal path plans are found using deterministic optimisers, but a common problems is when the procedure becomes trapped in local minima [5, 6]. Other techniques such as EAs are robust to find global solutions but suffer from large computational expenses. Therefore, one of the main objectives in optimal mission planning is to develop effective and efficient optimisation techniques in terms of computational cost and solution quality [5, 6].

A good summary of path planning algorithms was presented by LaValle in [7], while Francois *et al.* [8] elaborated on the importance of path planning, described the types of path planning techniques and explaned why Evolutionary and Genetic Algorithms are preferred as the most viable search algorithm for a real-time UAS path planner.

The following sections describe an advanced EA used in the context of mission path planning.

3 Robust Framework

The complext task of aircraft design is now assisted by highly sophisticated analysis tools such as computational fluid dynamics (CFD), finite element analysis (FEA) and mission simulation. The logical extension to this progress is undoubtedly optimisation.

Current optimisation efforts focus on the use of gradient-based techniques, which are mostly suitable for single-objective optimisation or when the objectives are differentiable. The recent emergence of new techniques such as Evolutionary Algorithms (EAs) [4, 8, 9], although can be computationally more intensive than gradient-based methods, have proven to be beneficial in the area of multi-disciplinary UAS design.

A real design of an UAS will have more than one objective, such as minimising fuel consumption, drag, as well as maximising range and endurance. In this research, the Hierarchical Asynchronous Parallel Multi-objective Evolutionary Algorithm (HAPMOEA) tool was used. This tool uses advanced concepts for the solution: a modified canonical evolution strategy [10, 11], capabilities for multi-objective optimisation using a Pareto tournament selection [12], capabilities for parallel computation using a modified asynchronous approach [4, 13] and a hierarchical/multi-fidelity approach for the solution [4, 8]. HAPMOEA has been compared and has shown some computational benefits to other EA methods [4, 14].

3.1 Hierarchical/Multi-Fidelity Population Topology

A hierarchical/multi-fidelity population topology, when integrated into an evolution algorithm, means that a number of separate populations are established in a hierarchical layout to solve the given problem, rather than a single 'cure-all' type single population layout. This method was proposed by Sefrioui [8] and is shown in Fig. 1.

The bottom layer uses a simple model and can be entirely devoted to exploration, the intermediate layer is a compromise between exploitation and exploration, while the top layer uses a refined model and concentrates on promising solutions.

In simulation tools, the accuracy of the solution is often related to the sampling time, or how often each instance of the simulation cycle is calculated. The smaller the sampling time, the more accurate the simulation will be to real-time execution. However, a smaller sampling time implies a long simulation time, due to limitations on the computational power available.

Applying the hierarchical/multi-fidelity topology to simulations, the top layer uses a precise time-consuming sampling time, whereas coarser sampling times are used in the intermediate and bottom layers, resulting in a more

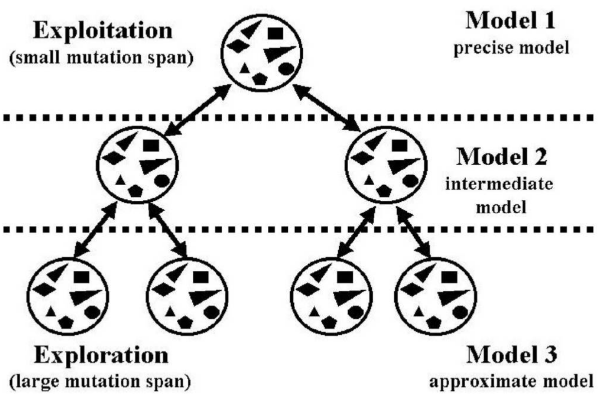

Fig. 1 Hierarchical topology.

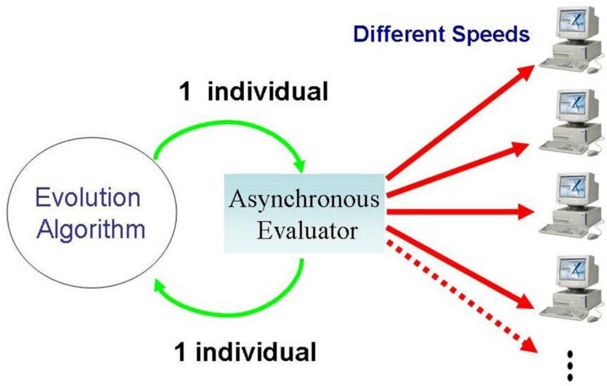

Fig. 2 Asynchronous Evaluation.

approximate model.

3.2 Parallel Computing

The algorithm used in this approach is a master-slave pMOEA but incorporates the concept of isolation and migration through hierarchical topology binary tree structure, where each level executes different MOEAs/parameters (heterogeneous). The parallel environment used is a cluster of PCs, wherein the master carries on the optimisation process while remote nodes compute the analysis solver environment. The message-passing model used is the Parallel Virtual Machine (PVM) [15].

A schematic of the parallelisation approach with asynchronous evaluation is shown in Fig. 2. This algorithm has been tested in a cluster of heterogeneous CPUs, RAMs, caches, memory access times, storage capabilities and communication attributes.

In this work, a cluster than can be configured with up to 18 machines with performances varying between 2.0 and 2.4GHz was used. Studies showing the performance of the algorithms are presented in [14].

3.3 Asynchronous Solution

When considering the solution to a Multi-objective/disciplinary Optimisation problem, several issues arise, as many methods of solution used in engineering today may take different times to complete their operation [11]. The classic example of this is a CFD, an FEA solver or a MATLAB® Simulink® simulation.

With a typical industrial code used for simulation and analysis of aircraft, the time for the solution to converge to a specified level (either machine zero or an arbitrarily selected higher value) can vary over a significant range. The time taken for an iterative solution of non-linear partial differential equations is strongly dependent upon geometry or trajectory flown.

The previous generation of EAs have mostly used a generation-based approach and so are the traditional genetic algorithm and evolution strategy. A difficulty with generational models is that they create an unnecessary bottleneck when used on parallel computers. If the population size is approximately equal to the number of processors, and most of the candidate offspring that are sent for solution can be successfully evaluated, then some processors will complete their task quickly with the remainder taking more time. With a generational approach, those processors that have already completed their solutions will remain idle until all processors have completed their work [10].

The approach used here is to ignore any concept of a generation-based solution. This approach is similar to the work by Wakunda and Zell [13] and other non-generational approaches. However, the selection operator is quite different, as it couples one-by-one (steady-state) function evaluation with a direct multi-objective fitness criterion.

Whilst a parent population exists, offspring are not sent as a complete 'block' space to the parallel slaves for solution. Instead, one candidate is generated at a time, and is sent to any idle processor where it is evaluated at its own speed.

When candidates have been evaluated, they are returned to the optimiser and either accepted by insertion into the main population or rejected. This requires a new selection operator because one offspring cannot now be compared against another, or even against the main population due to the variable-time evaluation. To overcome this, the recently evaluated offspring was compared against a previously established rolling benchmark and, if successful, it replaces, according to some rule, a pre-existing individual in the population.

This benchmarking is implemented via a separate evaluation buffer, B, which provides a statistical 'background check' on the comparative fitness of the solution. The length of the buffer should represent a reasonable statistical sample size, but need not be too large; approximately twice the population size is more than adequate.

When an individual has had a fitness assigned, it is then compared to past individuals (both accepted and rejected) to determine whether or not it should be inserted into the main population. If it is to be accepted, then some replacement strategy is invoked and it replaces a member of the main population. The replace-worst-always method is used exclusively in this work.

3.4 Multi-Objective Optimisation

Most EAs configured for multi-objective optimisation currently use the non-dominated sorting approach. This is a straightforward way to adapt an algorithm that is designed as a single-objective optimiser into a multi-objective optimiser, and is used by many researchers [12].

The problem with sorting approaches is that the method is not a fully integrated one. Briefly, a sorting method works by computing the set of non-dominated solutions amongst a large statistical sampling (either a large population or previous data), and assigning these solutions a rank one. Then, ignoring these points, the process is repeated until a 'second' Pareto front is found, and this is assigned a rank two. This process continues until all points are ranked, and then the value of the rank is assigned to the individual as a new single-objective fitness.

A problem arises now on whether it is fair to assign individuals in the second rank numerically half the fitness of the first, and whether the third rank deserves a third of the fitness of the first. This poses a dilemma regarding the level of equality present amongst the solutions, as often solutions with excellent information may lie adjacent to, but not in, rank one.

To solve this 'artificial scaling' problem, it is possible to introduce scaling, sharing and niching schemes. However, all of these require problem-specific parameters or knowledge, even in adaptive approaches. It is of course always desirable to compose an algorithm that does not introduce such unnecessary parameters.

The on-the-fly selection operator was implemented by means of a Pareto tournament selection operator. To implement an optimisation algorithm that is equally applicable to both single- and multi-objective problems, a suitable selection operator capable of handling either situation must be developed. An extension of the standard tournament operator popular in many approaches was proposed in [12].

The current operator is a novel approach in that is requires no additional 'tuning' parameters, works seamlessly with the asynchronous selection buffer, B, and is very easy to encode. To determine whether a new individual, x, is to be accepted into the main population, it is compared with the selection buffer by assembling a small subset of the buffer called the *t tournament functions*, which is as follows:

$$Q = [q_1 \quad q_2 \quad \cdots \quad q_n] \tag{1}$$

Q is assembled by selecting individuals from the buffer, exclusively at random, until it is full. Then it ensures that the new individual is not dominated by any in the tournament. If this is the case, then it is immediately accepted, and is inserted according to the replacement rules.

The only parameter that needs to be determined in advance is the tournament size, which is already present in a single-objective optimisation. Selection of this parameter requires a small amount of problem specific knowledge, and should vary between $Q = \frac{1}{2}B$ (strong selective pressure) and $Q = \frac{1}{6}B$ (weak selective pressure). The optimiser is not overly sensitive to this value, provided the user errs on the side of weak selective pressure (smaller tournaments) in the absence of better information.

The egalitarian approach to the tournament, by selecting individuals at random, ensures good diversity amongst the selected individuals; no niching or forced separation of individuals has been found to be necessary. It can also be seen that in the event that the fitness vectors have only one element (a single-objective optimisation), this operator simplifies to the standard tournament selection operator [12].

4 Multi-Disciplinary Design using Advanced Numerical Techniques

To demonstrate the application of HAPMOEA in the area of Hybrid UAS mission planning, computer simulations have been designed and implemented. Computer simulations are used in lieu of actual flight tests to reduce time and costs.

The design of the system was developed using HAPMOEA optimiser coupled with an Aircraft Simulation Model (ASM), which is described below.

4.1 Aircraft Simulation Model (ASM)

The computer simulations for demonstrating HAPMOEA in Hybrid UAS mission planning were conducted in the MATLAB® Simulink® simulation environment combined with the AeroSim Blockset [16], which offers a comprehensive aircraft simulation and analysis package [17]. The AeroSim Blockset also offers a detailed model of an Aerosonde™ UAV, a real-world UAS, with a complete set of parameter to simulate the Aerosonde™ in flight. A modified version of this model, to incorporate the hybrid propulsion system, was utilised in the construction of the ASM. The entire ASM is shown in Fig. 3 [18], which includes the *Aircraft Control* and *Flight Planner* blocks to the original Aerosonde™ model for unmanned operations.

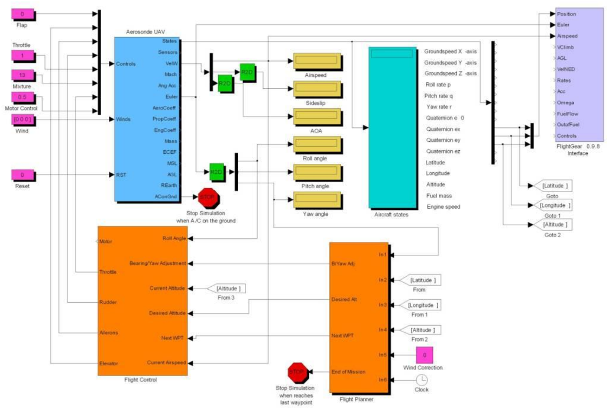

Fig. 3 Aircraft Simulation Model.

The current ASM has not yet taken into consideration the effects of weather elements such as wind on the performance of the aircraft during flight. This will be included in future revisions of the aircraft simulation model.

4.1.1 Modified Aerosonde UAV Block

The *Aerosonde UAV* block contains a detailed model of an Aerosonde™ UAV. Modifications have been made in the Propulsion model inside this block via the addition of an *Electric Motor & Generator (EM&G)* block to incorporate the Hybrid-Electric Powerplant (HEP), shown in Fig. 4 [19], that is a main aspect in this research. Each component in the HEP, excluding the Internal Combustion Engine (ICE) and Fuel components which are already represented inside the *Aerosonde UAV* block, is modelled by a module inside the *EM&G* block, and these modules are described below.

The *Electric Motor (EM)* module is based on a Plettenberg HP220/25 motor with constant 18V input [20]. The data values for HP220/25 is entered and used in the module as a Look-Up Ta-

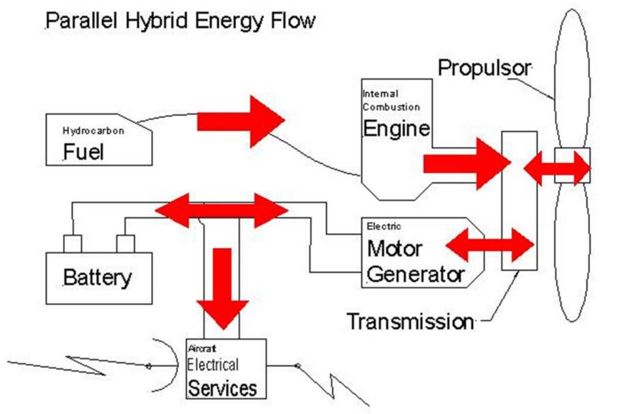

Fig. 4 Parallel Aircraft Hybrid-Electric Powerplant (HEP) Schematic.

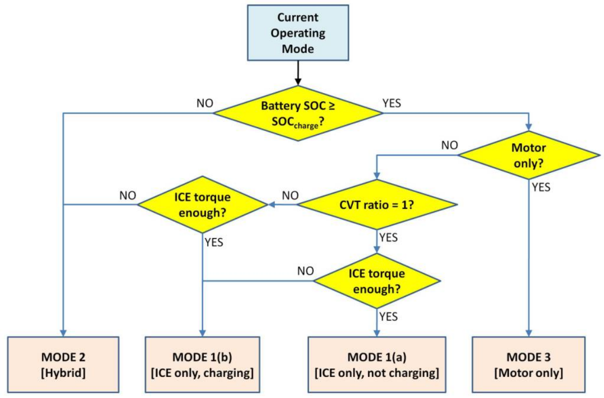

Fig. 5 The Strategy for the Switching of Operating Modes.

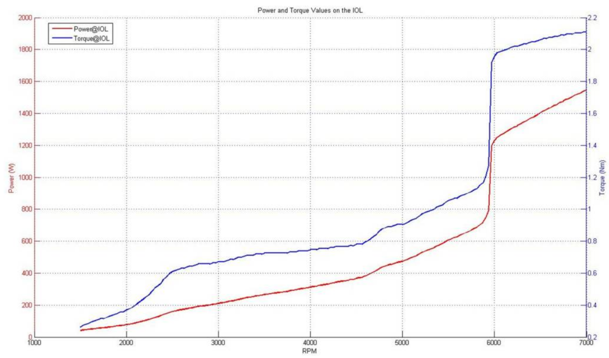

Fig. 6 Power (red) and torque (blue) values for the IOL.

ble (LUT), with linear interpolation used between data values. The *EM* module acts as a supplementary powerplant to the ICE when required, or as the sole powerplant when the aircraft is flying in Motor-only mode.

The *Generator (GEN)* module is also based on a Plettenbery HP220/25 motor, with *GEN* being functionally the "reverse" of *EM*. Its only function is to provide the current to charge the battery when extra torque is available in the propulsion system (i.e. from the ICE).

The *Battery (BAT)* module uses the data of two Air Thunder 5000mAh 6-cell Lithium Polymer (Li-Po) Battery Pack [21] in series to obtain the required output voltage for the *EM*. When the *BAT* module is in use, its *State of Charge (SOC)* parameter is monitored as indication whether it can still provide voltage to the *EM*, or if charging is required.

The *Aircraft Electrical Services (AES)* include all electronics which draw current from the *BAT* module, i.e. onboard avionics, camera, etc. Currently, only avionics have been taken into consideration in the implementation of this module.

The *Transmission (Tx)* module is assumed to be a Continuous Variable Transmission (CVT) which has a gear ratio range from 0.455 to 2.47 [22].

Additionally, a *Hybrid Propulsion Controller (HPC)* is also included in the *EM&G* block. This controller enables the switching of HEP operation modes according to the strategy represented in Fig. 5.

The controller utilises the Ideal Operating Line (IOL) method of operating the ICE in order to achieve the best performance while consuming the least amount of fuel possible [23]. Using the data values provided as part of the Aerosonde™ model in the AeroSimn Blockset, an analysis of the Aerosonde™ was carried out to obtain the IOL for the ICE, which is shown in Fig. 6.

4.1.2 Flight Planner Block

The *Flight Planner* block, developed at QUT, calls a MATLAB® function which calculates the necessary bearing/yaw adjustment from the current position of the aircraft to reach a desired waypoint according to a pre-specified list of waypoints. The waypoints are listed in the latitude, longitude and altitude coordinates. This bear-

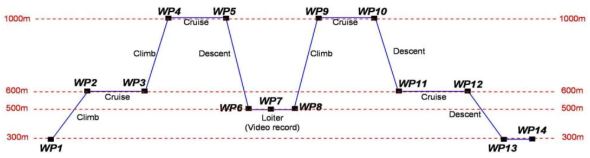

Fig. 7 Mission Profile (not to scale).

ing/yaw adjustment, along with other parameters, are passed to the *Flight Control* block.

4.1.3 Flight Control Block

The *Flight Control* block obtains output from the *Flight Planner* block and uses PID and PI controllers to compute the values for aircraft controls. These values for the control inputs to the *Aerosonde UAV* block to complete the loop for unmanned aircraft operations.

5 Practical Test Cases

Practical test cases were set up to demonstrate the capabilities of the HAPMOEA optimiser in the area of UAS mission planning. The following sections describe the process.

5.1 Mission Scenario

In order to demonstrate the capabilities of HAPMOEA in HPUAS mission planning, a baseline mission scenario was constructed. This mission scenario includes basic UAS operations such as Climb, Cruise, Descent and Loiter, and follows the mission profile in Fig. 7.

A long-distance mission scenario with a total distance of approximately 400km has been constructed. This realistic mission scenario utilises GPS waypoints located in central Queensland, Australia, indicated in Fig. 8.

However, the waypoints which the HPUAS has to pass through are not definite and may be adapted to achieve the optimisation objectives defined above. An example of such adaptation of mission waypoints is demonstrated in Fig. 9.

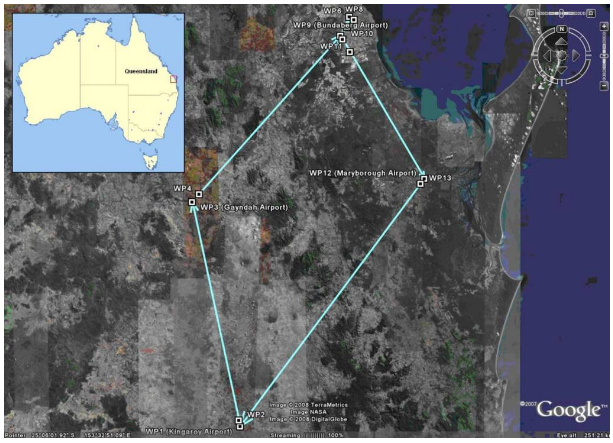

Fig. 8 Baseline Mission Scenario with GPS Waypoints in Central Queensland, Australia (image generated using Google^{TM} Earth.

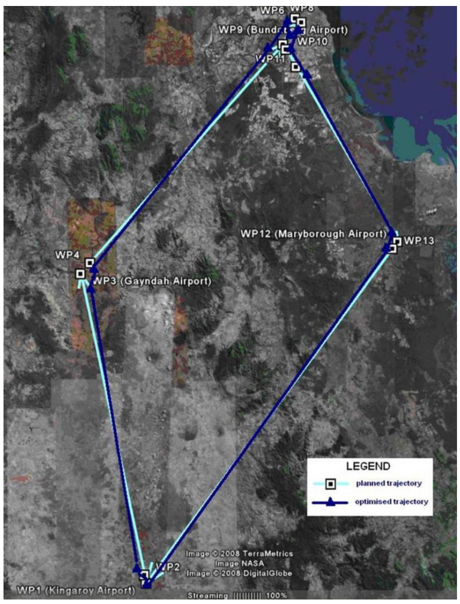

Fig. 9 Example Comparison of a Set of Optimised Waypoints (navy) for Minimum Fuel and Flight Time to the Baseline Flight Mission (light blue) (not to scale).

5.2 Problem Definition

This multi-objective problem considers the waypoint optimisation for a HPUAS mission. The objectives considered are the minimisation of fuel consumption and the time required to execute the mission. The HPUAS baseline flight mission is defined using the GPS waypoints shown in Fig. 8.

It needs to be noted that the leg from Waypoints 6 to 8 in the mission is the 'loiter' phase, as can be seen from Fig. 7, during which some special mission requirements are carried out. Therefore, it is desired that these waypoints remain as specified and not be involved in the optimisation process.

5.3 Definition of Design Variables

The design variables used for HPUAS mission waypoint optimisation are the locations of the waypoints. The flight mission is made up of a series of waypoints and each waypoint is defined by its coordinates: latitude, longitude and altitude. The entire mission will be passed through the optimisation process in order to determine a set of waypoints which whill achieve the optimisation objectives.

In this paper, South latitudes and East longitudes are taken as positive values, while North and West as negative. This convention was chosen for convenience as the waypoints considered for the simulation are located in Australia. Also, the latitudes and longitudes are measured in radians, and the altitudes in metres.

5.4 Definition of Fitness Objective Functions

The fitness functions are defined as the minimisation of fuel consumption, FC, and the time required to execute the mission, T_{req}:

$$min\,(f_1) : f_1 = FC \tag{2}$$

$$min\,(f_2) : f_2 = T_{req} \tag{3}$$

5.5 Definition of Constraints

The process of optimising HPUAS mission waypoints takes into consideration of a number of constraints, namely the upper and lower bounds of waypoint coordinates and physical constraints.

5.5.1 Upper and Lower Bounds of a Waypoint

The upper and lower bounds of a waypoint defines a set of values which a candidate waypoint, the possible coordinates to be passed through during the mission, is selected. The generation of a set of candidate waypoints forms a key component of the optimisation process. These upper and lower bounds of each coordinate - namely latitude, longitude and altitude - are defined taking into consideration the mission requirements as well as airspace and class restrictions. The chosen set of waypoints is then used in the 'solver' component of the optimisation process to determine the fitness function of this particular set of waypoints.

The upper and lower bounds for altitude, in metres, is calculated simply using a specified altitude margin, Δalt. In this paper, the value of Δalt was chosen to be 100m.

On the other hand, defining the bounds for latitude and longitude of a waypoint is more complex. This is due to the fact that, realistically, when an aircraft is flying, the distance between two waypoints is not the straight line route, but the line vertically above the straight line route following the earth's surface, known as a 'great circle'. Therefore, spherical-triangle calculation is used for range- and bearing-related computations along a great circle [24], such as those required for defining the bounds for latitude and longitude of a waypoint. In this paper, the upper and lower bounds of the latitude and longitude are defined and calculated as 10% of the distance between the waypoint and the preceding waypoint, found using a rearranged form of the Haversine Formula [25].

5.5.2 Physical Constraints

Several physical constraints were incorporated into the ASM, shown in Fig. 3, to make the simulation model more realistic. The majority of these constraints were implemented as part of the *Flight Control* block in the ASM, and these are:

- The throttle control must be within a range from 1% to 100% ($0.01 \leq TC \leq 1$).
- The rudder deflection must be between $-20°$ and $+20°$ ($-20° \leq \delta_r \leq +20°$).
- The aileron deflection must be between $-10°$ and $+10°$ ($-10° \leq \delta_a \leq +10°$).
- The elevator deflection must be between $-20°$ and $+20°$ ($-20° \leq \delta_e \leq +20°$).
- The airspeed is controlled at 20m/s when the aircraft at level flight and climbing, and increased to 30m/s when descending.

All these constraints are applied in the ASM and are part of the simulation process.

5.6 Implementation Design and Optimisation Rationale

In the implementation of the HPUAS mission waypoint optimisation, the HAPMOEA optimiser is set up with only one layer. This layer has a population size of 10, two parents in recombination, a buffer length of 12 and a tournament-in-buffer ratio of 2.0 (refer to §3). The definition of the lower and upper bounds vectors, *lb* and *ub* respectively, are calculated as per §5.5.1.

The HAPMOEA optimiser calls the ASM, using a fundamental sample time, Δt, set at 0.1 seconds, to evaluate each set of candidate waypoints which has been generated in the process. The outputs of the ASM are the fuel consumption over the mission duration and the time required to execute the mission. These form the objectives which are to be minimised by the HAPMOEA optimiser.

The optimisation rationale is displayed in Fig. 10.

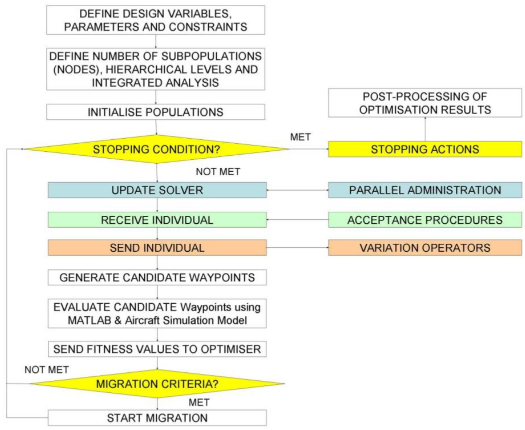

Fig. 10 The HAPMOEA Optimisation Rationale Flow Diagram.

# Pareto Front Members	Fuel Consumption (kg)	Mission Time Required (hrs)	Time Taken (hrs)
1 Pareto Front Member	0.703125	6.023	96
Baseline	0.639029	6.55	-

Table 1 Table of Statistics for Multi-Objective Optimisation.

5.7 Optimisation Results and Post-Processing of Optimal Solutions

The stopping condition used in the HAPMOEA optimiser in the HPUAS mission waypoint optimisation is a total runtime of 96 hours on one machine only. Table 1 shows the run statistics and the fitness values for a selected number of Pareto front members. *This table is a placeholder only, for the time being, as results are still being compiled.*

6 Conclusions

The basic conepts of a hierarchical, asynchronous parallel multi-objective EA used to solve the problem of optimising a Hybrid UAS mission were presented in this paper. Even though more computationally expensive, an EA optimiser can provide an UAS mission planner extended benefits in terms of improved endurance and/or range, and extra payload capacity. The method can be used as an alternative option to satisfy some of the needs for robust multi-objective and multidisciplinary design optimisation problems. The method is easily coupled, particularly adaptable, easily parallelised, and required no gradient of hte objective function(s). The methodology is integrated in a single framework that allows:

- Solving of single- and multi-objective, non-linear, deceptive, discontinuous, and multi-model problems;
- Incorporation of different game strategies - Pareto, Nash, Stackelberg;
- Implementation of multi-fidelity approaches;
- Parallel computations; and
- Asynchronous evaluations.

Further extensions of the ASM, and the developing and conduction of flight experiments with an UAS are presently under investigation.

7 Acknowledgments

The authors gratefully acknowledge Mourad Sefrioui of Dassault Aviation and Eric Whitney of Boeing Australia for fruitful discussions on Hierarchical EAs. Additional thanks goes to Pillar Eng of QUT and Australian Research Centre for Aerospace Automation (ARCAA) for assistance in the implementation of the *Flight Planner* module within the ASM, and Richard Glassock for assistance in the incorporation of the HEP in the ASM.

8 Copyright Statement

References

[1] *Jane's Unmanned Aerial Vehicles and Targets*, [online database], URL: http://juav.janes.com/public/juav/index.shtml [cited March 2007].

[2] Glassock, R., Hung, J.Y., González, L.F. and Walker, R.A. (2009). "Multimodal hybrid powerplant for unmanned aerial systems (UAS) robotics," *Twenty-Fourth Bristol International Unmanned Air Vehicle Systems Conference*, 30 March - 1 April, 2009, Bristol, United Kingdom.

[3] Kroo, I., Altus, S., Braun, R., Gage, P, and Sobieski, I. (1994). "Multidisciplinary optimisation methods for aircraft preliminary design," AIAA-94-4325, *Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, 7-9 September, 1994, Panama City, Florida.

[4] González, L., Whitney, E., Srinivas, K., Armfield, S. and Périaux, J. (2005). "A robust evolutionary technique for coupled and multidisciplinary design optimisation problems in aeronautics," *Computational Fluid Dynamics Journal*, Vol. 14, No. 2, pp. 142-153.

[5] Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, Wiley: UK.

[6] Lee, D.S., González, L.F., Srinivas, K. and Périaux, J. (2008). "Robust evolutionary algorithms for UAV/UCAV aerodynamics and RCS design optimisation," *Computers & Fluids*, Vol. 37, pp. 547-564.

[7] LaValle, S.M. (2006). *Planning Algorithms*, Cambridge: UK.

[8] Sefrioui, M. and Périaux, J. (2000). "A hierarchical genetic algorithm using multiple models for optimisation," *Parallel Problem Solving from Nature - PPSN VI*, edited by M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo and H.-P. Schwefel, Springer: New York, pp. 879-888.

[9] Obayashi, S. (1998). "Multidisciplinary design

optimisation of aircraft wing planform based on evolutionary algorithms," *1998 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, pp. 3148-3153.

[10] Hansen, N. and Ostermeier, A. (2001). "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, Vol. 9, No. 2, pp. 159-195.

[11] Wakunda, J. and Zell, A. (2000). "Median-selection for parallel steady-state evolution strategies," *Parallel Problem Solving from Nature - PPSN VI*, edited by M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo and H.-P. Schwefel, Springer: New York, pp. 405-414.

[12] Michalewicz, Z. (1992). "Genetic algorithms + data structures = evolution programs," *Artificaial Intelligence*, Springer-Verlag: New York.

[13] Van Veldhuizen, D.A., Zydallis, J.B. and Lamont, G.B. (2003). "Considerations in engineering parallel multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, pp. 144-173.

[14] González, L.F., Whitney, E., Périaux, J., Sefrioui, S. and Srinivas, K. (2004). "A robust evolutionary technique for inverse aerodynamic design and control of aerospace systems using tools from nature," *Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering, Volume II*, 24-28 July, 2004, Jyvaskyla, Finland.

[15] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. (1994). *PVM: Parallel Virtual Machine. A user's guide and tutorial for networked parallel computing*, Massachusetts Institute of Technology: Cambridge, Massachusetts.

[16] *AeroSim Blockset, Aeronautical Simulation Blockset, Ver. 1.2 User's Guide*, Unmanned Dynamics, URL: http://www.u-dynamics.com.

[17] McManus, I., Clothier, R. and Walker, R. (2005). "Highly autonomous UAV mission planning and piloting for civilian airspace operations," *Proceedings of the Eleventh Australian International Aerospace Congress, AIAC-11, First Australasian Unmanned Air Vehicles Conference*, 13-17 March, 2005, Melbourne, Australia.

[18] Hung, J.Y., González, L.F., Walker, R.A. and Périaux, J. (2009). "Mission optimisation and multi-disciplinary design of hybrid unmanned aerial systems (UAS) using advanced numerical techniques," *Proceedings of the Thirteenth Australian International Aerospace Congress, AIAC-13, Third Australasian Unmanned Air Vehicles Conference*, 9-12 March, 2009, Melbourne, Australia.

[19] Glassock, R., Hung, J.Y., González, L.F. and Walker, R.A. (2008). "Design, modelling and measurement of a hybrid powerplant for unmanned aerial systems," *Australian Journal of Mechanical Engineering*, Vol. 6, No. 2, pp. 69-78.

[20] *HP220/25 Gear Motor datasheet*, Plettenberg Elektromotoren, URL: http://www.plettenberg-motoren.com/Datenblaetter/220_25_GEAR/HP22025a3p45_1.pdf.

[21] *Air Thunder 5000mAh 6-cell Lithium Polymer (Li-Po) Battery Pack*, URL: http://www.airthunder.com.

[22] Yeo, H. and Kim, H. (2002). "Hardware-in-the-loop simulation of regenerative braking for hybrid electric vehicles," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 216, No. 11, pp. 855-864.

[23] Oudijk, M.F. (2005). *Optimization of CVT control: For hybrid and conventional drive lines*, Master Thesis Report, Eindhoven University of Technology, Eindhoven, The Netherlands; University of California, Davis, California.

[24] Kayton, M. (1997). "The navigation equations," *Avionics Navigations Systems*, edited by M. Kayton and W.R. Fried, John Wiley & Sons: New York, pp. 21-54.

[25] Sinnot, R.W. (1984). "Virtues of the Haversine," *Sky and Telescope*, Vol. 68, No. 2, p. 159.