Optimal Mission Path Planning (MPP) For An Air Sampling Unmanned Aerial System

Luis Felipe Gonzalez
School of Engineering System, Queensland University of Technology, Brisbane QLD 4001 Australia.
felipe.gonzalez@qut.edu.au
DongSeop Lee
International Center For Numerical Methods In Engineering (CIMNE/UPC)
Edificio C1, Gran Capitan, s/n. 08034 Barcelona – Spain. ds.chris.lee@gmail.com
Rodney Walker
School of Engineering System, Queensland University of Technology, Brisbane QLD 4001 Australia.
ra.walker@qut.edu.au
Jacques Periaux
International Center For Numerical Methods In Engineering (CIMNE/UPC)
Edificio C1, Gran Capitan, s/n. 08034 Barcelona – Spain. jperiaux@gmail.com

Abstract

This paper presents advanced optimization techniques for Mission Path Planning (MPP) of a UAS fitted with a spore trap to detect and monitor spores and plant pathogens. The UAV MPP aims to optimise the mission path planning search and monitoring of spores and plant pathogens that may allow the agricultural sector to be more competitive and more reliable. The UAV will be fitted with an air sampling or spore trap to detect and monitor spores and plant pathogens in remote areas not accessible to current stationary monitor methods.

The optimal paths are computed using a Multi-Objective Evolutionary Algorithms (MOEAs). Two types of multi-objective optimisers are compared; the MOEA Non-dominated Sorting Genetic Algorithms II (NSGA-II) and Hybrid Game are implemented to produce a set of optimal collision-free trajectories in three-dimensional environment. The trajectories on a three-dimension terrain, which are generated off-line, are collision-free and are represented by using Bézier spline curves from start position to target and then target to start position or different position with altitude constraints. The efficiency of the two optimization methods is compared in terms of computational cost and design quality. Numerical results show the benefits of coupling a Hybrid-Game strategy to a MOEA for MPP tasks. The reduction of numerical cost is an important point as the faster the algorithm converges the better the algorithms is for an off-line design and for future on-line decisions of the UAV.

1 Introduction

An Unmanned Aerial System (UAS) can be fitted with an air sampling device or spore trap to detect and monitor spores or plant pathogens. Australia for instance is a large country, and the landscape is not always accessible and has agricultural areas that are in remote locations, where topography and climate conditions could make traditional monitoring and surveillance methods almost impossible. An UAS fitted with an air sampling device flying an optimal path can monitor and reduce the risk of pest introduction from international trade and, at the same time, will capture a wide range of plant health information in a cost-effective way so as to cover international and domestic market demands. QUT, in conjunction with the CRC for Plant Biosecurtiy, Department of Agriculture and Food, Western Australia.s Murdoch University, Queensland Department of Primary Industries and Fisheries, is currently involved in a project to develop an unmanned vehicle (UAV) to monitor inaccessible cultivation areas and look for either unwanted spores or other plant pathogens.

The UAS been developed for this work is one of Queensland University of Technology UAS – a 3.5 Silverstone Flamingo UAV airframe fitted with a Micropilot Heli 2128 and a QUT developed avionics as shown in Figure 1.

One important aspect of a UAS is the design of an optimal path plan and trajectory. Traditionally optimal path plans are found using deterministic optimisers but this may be trapped in local minima [Tang et al., 2005]. Other techniques such as evolutionary algorithms are robust to find global solutions but suffer from large computational expense; therefore, one of the main objectives in optimal path planning is to develop effective and efficient optimization techniques in terms of computational cost and solution quality [Lee et al., 2008a] and [Lee et al., 2008b].

Figure 1. ARCAA/Queensland University of Technology Flamingo UAS.

This paper investigates two different game strategies for multi-objective Mission Path Planning (MPP) optimization; the first method is a well known Non-Dominated Sorting Genetic Algorithm NSGA-II [Deb et al., 2000]. The second optimization method is developed based on NSGA-II using the concept of Nash equilibrium [Wang and Periaux 2001] and [Periaux et al., 2006] and Pareto optimality [Deb et al., 2001] and [Lee, 2008] (Hybrid-Game). In this paper, a concept of Hybrid-Game strategy is applied to a multi-objective optimiser; NSGA-II however it can be implemented to other MOEA optimiser.

The Hybrid-Game on NSGA-II consists of several Nash-Players and one Pareto-Player. Each Nash-Player optimises its own local criteria using its own strategy to speed up the search for a global design or Pareto-front.

The evolutionary optimization methods NSGA-II and Hybrid-Game are applied to produce a set of useful optimal trajectories in a three-dimensional environment. The trajectories on a three-dimension terrain are represented using Bézier spline curves from start position to target and then target to start position or a different position under altitude constraints. Results from both optimization techniques are compared in terms of design quality and computation expense.

The rest of paper is organized as follows; Section 2 presents the methodology for NSGA-II and Hybrid-Game applied to NSGA-II. A Mission Path Planning is described in Section 3. Section 4 considers a MPP design problems using NSGA-II and Hybrid-Game. Conclusions and forthcoming work are described in Section 5.

2 Methodology

In this section, two evolutionary optimization methods; NSGA-II and Hybrid-Game applied to NSGA-II are presented. The first method NSGA-II is a modified version of a well-known non-domination based genetic algorithms NSGA to have a better sorting algorithm, incorporates elitism. NSGA-II uses Pareto tournament to produce Pareto non-dominated solutions [Deb, 2001]. In the second method NSGA-II is hybridized by applying the concept of Nash-equilibrium coupled to Pareto optimality.

2.1 NSGA-II

NSGA-II uses a binary tournament selection, Simulated Binary Crossover (SBX) and polynomial mutation [Deb and Agrawal, 1995] and [Deb, 2001] and . Figure 2 describes the algorithm for NSGA-II which has seven main steps:

```
Step1: Define population size, the number of generations as stopping criteria, dimension of decision variables and design bounds, and objective/fitness functions.
Step2: Initialise a random population of candidate paths
Step3: Sort non-dominated solutions from initial random population with individual rank and crowding distance corresponding to fitness values or position in front.
while Stopping Criteria (generation number)
    Step4: Do tournament selection based on individual rank and crowding distance.
    Step5: Do genetic operation which consists of crossover and mutation to generate an offspring population.
    Step6: Sort non-dominated solutions from combined population (Parent population + offspring population).
    Step7: Replace the best solutions based on its rank and crowing distance to parent population.
end
```

In *Step3*, each individual in population will be assigned with a non-domination rank as well as its crowding distance. The tournament selection (*Step4*) will be through based on the non-domination rank of individual. If individuals have same non-domination rank then individual with large crowding distance will be selected. NGA II has been compared with more traditional deterministic methods and have shown robustness to find global solutions for very hard problems, Reference 8 shows a comparison and results of NSGA and deterministic methods for very hard multimodal, multi point, with multiple local minima problems.

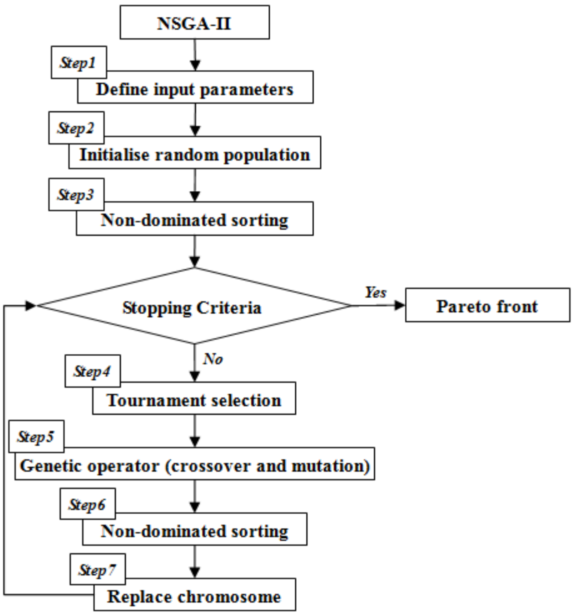

Figure 2. Algorithm for NSGA-II.

2.2 Hybrid-Game Applied To NSGA-II

This method couples the concept of Nash-game and Pareto optimality to NSGA-II and hence it can simultaneously produce Nash-equilibrium and Pareto non-dominated solutions. The Pareto solutions contain a set of possible trajectories whilst the Nash players have an optimal path each. The Hybrid-Game consists of several Nash-Players corresponding to the objectives of problem. Each Nash-Player has its own optimization criteria and uses its own strategy. A Nash-equilibrium is obtained when each Nash-Player cannot improve its objective.

In the context of path planning if the overall objective is to minimise the path distance between two points; start and target and target to start the optimization can be split in two with one player optimising the path: start to target and the other player optimising target to start. Note that the context of an evolutionary optimiser the paths start to target and target to start may be different as both players will be optimising with different set of populations and genetic material. In the context of Artificial Intelligent (AI) sampling, the UAV can sample air during flight or can be tasked to fly an optimal trajectory from the launching area (start) to the potentially infected area (target).

The reason for implementation of Nash-game coupled to Pareto optimality is to accelerate the search for one of the global solutions [Lee, 2008]. The elite design from each Nash-Player will be seeded to a Pareto-Player at every generation. The algorithm of NSGA-II with hybrid game is shown in Figure 3 where eight main steps are;

Step1: Define population size, the number of generations as stopping criteria, dimension of decision variables and design bounds, and objective/fitness functions for Nash Players and Pareto player.

Step2: Initialize three random populations of paths; One for Pareto-Player, one for Nash-Player1 and one for Nash-Player2.

Step2-1: Transfer elite design variable from the Pareto-Player to Nash-Player1.

Step2-2: Transfer elite design variable from the Nash-Player1 to Nash-Player2

Step3: Sort non-dominated solutions from initial random population with individual rank and crowding distance corresponding to fitness values or position in front.

while Stopping Criteria (generation number)

Step4: Do tournament selection based on individual rank and crowding distance.

Step5: Do genetic operation in each population which consists of crossover and mutation to generate an offspring population.

Step5-1: Seed elite design from the Nash-Player1 and the Nash-Player2 to the Pareto-Player population if only if the first offspring of each generation is considered.

Step5-2: Send and use elite design from each Nash-Player to the other Nash-Player.

Step6: Sort non-dominated solutions from combined population (Parent population + offspring population) on the Pareto-Player.

Step7: Replace the best solutions based on their rank and crowing distance to parent population.

Step8: Replace the elite design by Nash-Player1 and Nash-Player2

end

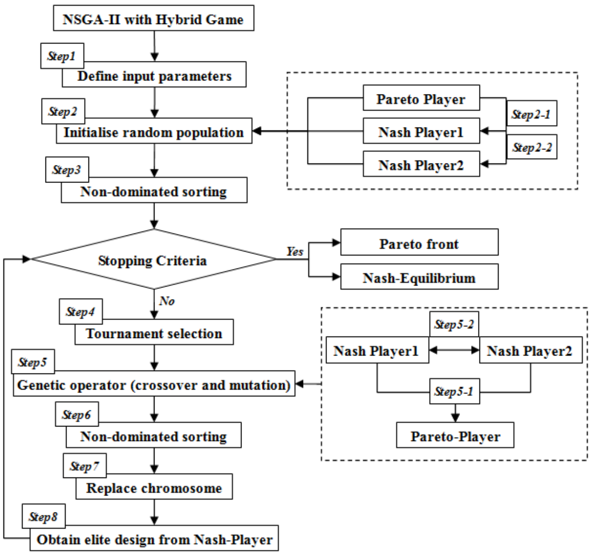

Figure 3. Algorithm for Hybrid-Game.

For example, a problem considers two objectives ($f_1 = x^2 y$, $f_2 = xy^2$) to minimise f_1 and f_2 where design variables are x and y. A Hybrid-Game will consist of one Pareto Player and two Nash Players. The Pareto-Player will optimise x and y to minimise both f_1 and f_2 while Nash Player1 will only optimise x to minimise f_1 using design variable y fixed by Nash-Player2. Nash-Player2 will only optimise y to minimise f_2 using design variable x fixed by Nash-Player1.

In *Step2*, Pareto-player initializes a random population for

f_1 and f_2, and sends the elite design variable (y_{elite}) for f_2 to the Nash-Player1. Nash-Player1 initializes a random population using y_{elite} from Pareto-Player and sends elite design (x_{elite}) for f_1 to Nash-Player2. Nash-Player2 initializes its random population using x_{elite} from Nash-Player1.

At *Step5-1*, the Pareto-Player uses elite design variables (x, y) if only if the first offspring of each generation is considered. Nash-Players 1 and 2 will use their elite design at each offspring (*Step5-2*).

The difference between NSGA-II with Hybrid-Game applied to NSGA-II is that NSGA-II uses only one-type of population to generate Pareto optimal front while Hybrid-Game on NSGA-II considers three-types of populations (Pareto, Nash-Player1, Nash-Player2).

The Hybrid Game strategy has been compared to other optimization methods in Reference 2. Results show that the Hybrid Game is capable of capturing global optimal solutions for very hard problems. Figure 4, for example, shows the solutions to a very hard problem; a multi-objective Non-Uniformly Distributed Convex Optimisation defined by Deb [Deb, 2001]. It can be expressed using equations 1 and 2. It is seen how the method captures the global solution to this hard problemas. A deterministic method will fail to find a solution in this type of problems.

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

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

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

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

$$h(f_1, g) = \begin{cases} 1 - \left(\dfrac{f_1}{g}\right)^{\alpha} & if\ f_1 \le g \\ 0 & otherwise \end{cases}$$

$$\alpha = 4$$

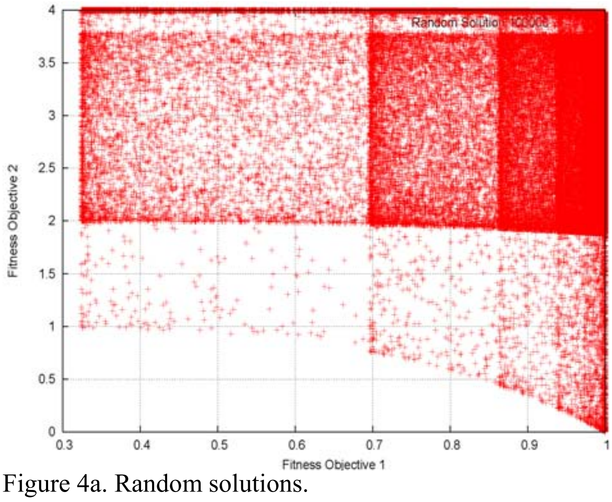

Figure 4a. Random solutions.

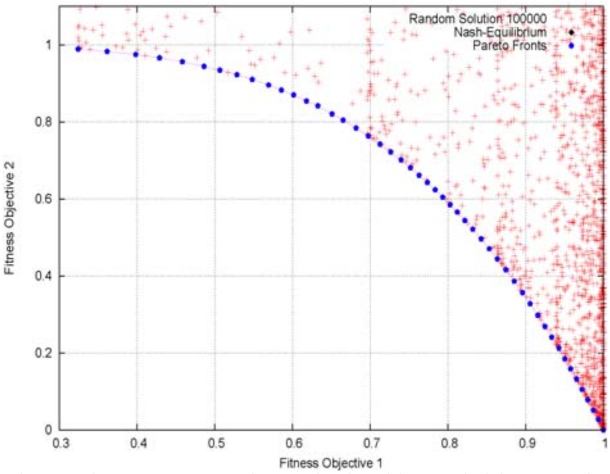

Figure 4b. True Pareto front obtained by Hybrid-Game for non-uniformly distributed non-convex design.

Figures 4c and 4d show the initial population obtained by NSGA-II and Hybrid-Game on NSGA-II. It can be seen that NSGA-II found 9 Pareto members with better fitness values for the objective 1 when compared to Pareto-Player in Hybrid-Game which found 7 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 Figure 4d. 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.

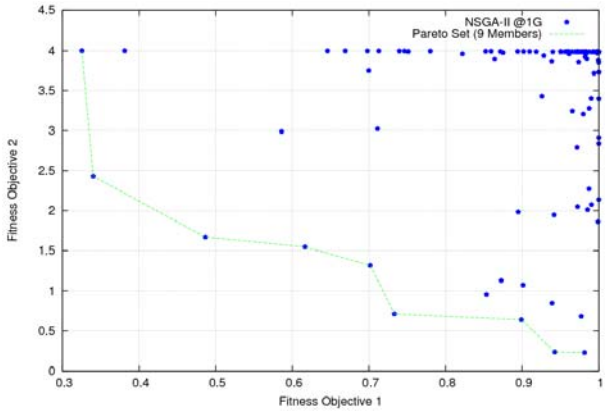

Figure 4c. Initial population obtained by NSGA-II.

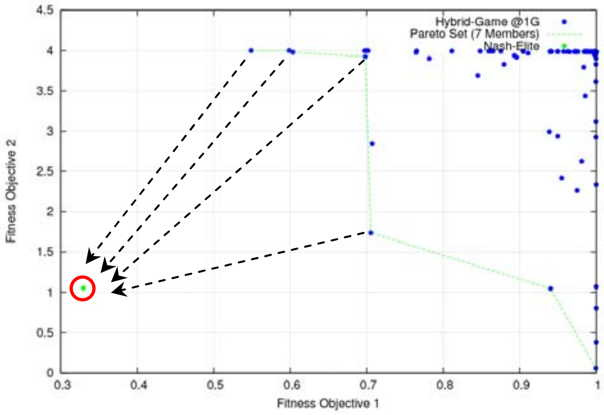

Figure 4d. Initial population obtained by Hybrid-Game on NSGA-II.

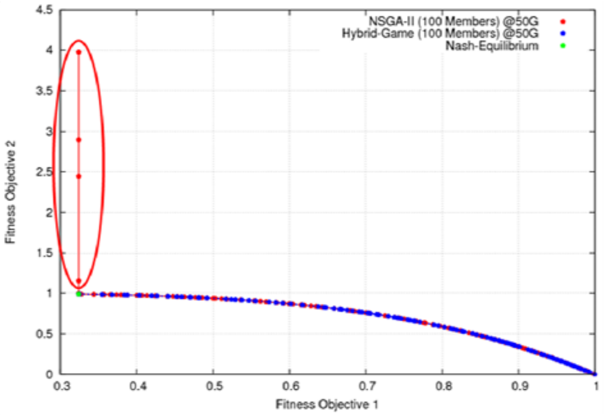

Figure 4e. Pareto fronts obtained by NSGA-II and Hybrid-Game on NSGA-II after 50 generations.

Figure 4e 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.

In this paper, each Nash-Player will optimise either the paths from the start position to the target position or from the target position to the start/end position.

3 Mission Path Planning

In this work, a we consider a UAS fitted with an air sampling device or spores trap will survey, monitor a mountainous area and avoid collision with known fixed obstacles from a start position to a target position or from a start point to a target point and to a different end point.

Results obtained by NSGA-II and Hybrid-Game will be compared in terms of solution quality and computational expense.

3.1 3-D Terrain

The terrain is represented by meshing three dimensional surfaces with obstacles and altitude constraints. There can be two types of terrain; the first is a short distance with a small number of obstacles and hazard zone as shown in Figure 5a and the second is a long distance terrain with a large number of obstacles as shown in Figure 5b. In this paper, long distance terrains are considered.

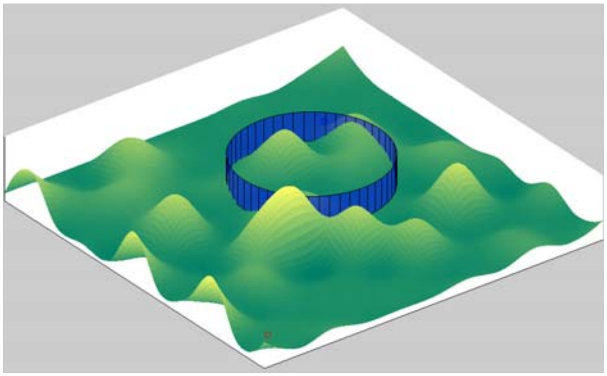

Figure 5a. Short distance terrain.

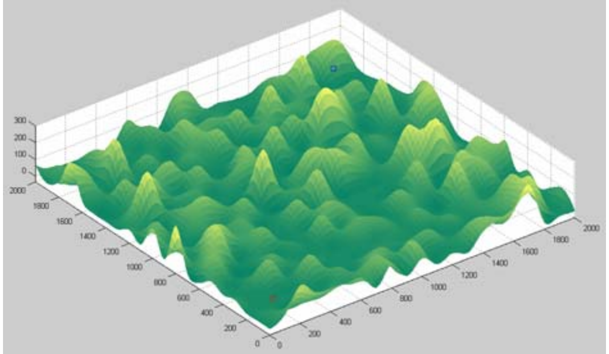

Figure 5b. Long distance terrain.

The example terrain is shown in Figure 5b where there are obstacles in 90% of the area to survey. The red square is the starting position and the blue square is the target position. This artificial terrain is randomly generated however it could represent some real geographical data.

For the application considered in Section 4, a constraint is imposed on the UAV to fly below 60% of maximum altitude due to limitations in the air sampling/spores trap or due to regulatory constraints as represented by the pink surface as shown in Figure 6.

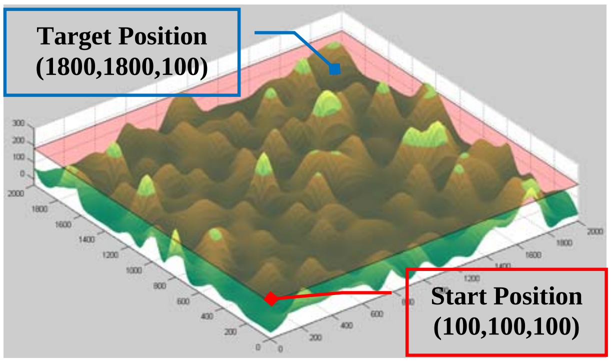

Figure 6. Baseline terrain with altitude constraints (Test 1).

The UAV may be tasked to fly higher however the UAV may need to fly at lower altitudes due to;

1. Air Traffic Control and separation.

2. Sampling objectives cannot reach to high altitude. For instance, led will not be polluted upto 10,000 *ft*.

Therefore, to be a valid, collision free trajectory, the *z* coordinates of a candidate trajectory should be below this altitude constraint and should avoid obstacles in the x and y direction. Two terrains are considered in the applications; Test 1 (Figure 6) and Test 2 (Figure 7).

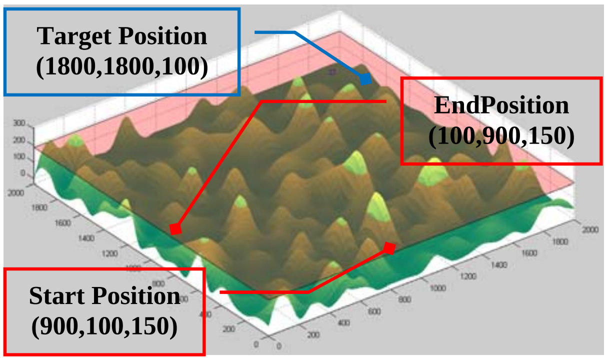

Figure 7. Baseline terrain with altitude constraints (Test 2).

The paper considers two applications;
Test case 1 considers minimization of two trajectories from the start position (100, 100, 100) to the target position (1800, 1800, 100) and then from the target to the start position.
Test case 2 considers minimization of two trajectories from the start position (900, 100, 150) to the target position (1800, 1800, 100) and then flying to the end position (100, 900,150).

3.2 Collision-Free Trajectory

The trajectory is generated using Bézier spline curves in three dimensional environment since the Bézier functions are useful in defining shapes and surfaces without sharp corners. The Bézier spline curves are computed from a parametric mathematical function which uses the control points (P_n) as parameters in terms of three dimensional Cartesian coordinate system i.e. (x, y, z). The start, target, end positions are fixed and the middle control points are variables. The coordinates of a trajectory are computed using equations 3, 4 and 5;

$$X(t)=\sum_{i=1}^{n}\left(\sum_{j=2}^{m}\left(\left(1-t_i\right)x_0 \ldots x_{j-1}+t_i x_1 \ldots x_j\right)\right) \qquad (3)$$

$$Y(t)=\sum_{i=1}^{n}\left(\sum_{j=2}^{m}\left(\left(1-t_i\right)y_0 \ldots y_{j-1}+t_i y_1 \ldots y_j\right)\right) \qquad (4)$$

$$Z(t)=\sum_{i=1}^{n}\left(\sum_{j=2}^{m}\left(\left(1-t_i\right)z_0 \ldots z_{j-1}+t_i z_1 \ldots z_j\right)\right) \qquad (5)$$

where *n* is the number of coordinates points in the trajectory and *m* represents the number of control points for Bézier spline curve. The parameter *t* is between 0 to 1 i.e $t\in[0,1]$.
Twenty two control points are considered to produce a detailed trajectory. The trajectory from the start position to the target position will be marked as a red line while a blue line will represent the trajectory from the target to the start position.

3.4 Fitness Functions and Penalty Abstract

The optimization consists of minimising the length of collision-free trajectories from the start to the target position and also the feasible return path to the start position or to an end position. The overall fitness function is therefore

$$f=\min\left(length(Path_{Total})\right)+Penalty \qquad (6)$$

where a *Penalty* will be applied when the *z*-coordinates of trajectories is lower than the *z*-coordinates of obstacles or higher than the altitude constraints. A penalty approach allows using good genetic material; part of a trajectory that may be optimal. Fitness function (6) represents the minimization to a single single-objective problem; however, the problem can be modified as a multi-objective problem if we consider two objectives one to minimise the length of the start to target path and a second one to minimise the target to start trajectory. The start position is (100, 100, 100) and the end position is (100, 900,150).
The minimum distance; without constraints, is the straight line from the start position to the target position and back.

4 Real World Design Optimisation

Two applications are conducted for MPP design optimization using NSGA-II and Hybrid-Game. The Hybrid-Game employs one Pareto-Player and two Nash-Players that optimise the one direction trajectory; Test 1: Nash-Player1 (Start-Target) and Nash-Player2 (Target-Start), Test 2: Nash-Player1 (Start-Target) and Nash-Player2 (Target-End).

4.1 Test1:START-TARGET-START

Problem Definition: The test is to minimise total trajectory length from the start position (100, 100, 100) to the target position (1800, 1800, 100) and back to the start position. This scenario occurs when a UAV is launched and is asked to fly an optimal trajectory, collect some samples and return back to lunching point and The fitness functions are;

$$f_1=\min\left(length(Path_{S-T})\right)+Penalty$$
$$f_2=\min\left(length(Path_{T-S})\right)+Penalty$$

Subject to;
z-coordinates $< Altitude_{radar}$
z-coordinates $> Altitude_{Obstacles}$

Stopping criteria;
$\mu_i \geq Pop_{Total}$ and $F_{\mu i} \leq Path_{min+10\%}$
or $Elaps_{Time} \geq 3$ hours

where μ_i is the number of individuals in total population ($Pop_{Total}=20$) and $F_{\mu i}$ is the fitness value of individuals.

The stopping criteria is when the number of feasible solutions is equal or greater than total population size, and all paths lengths in the population have a length lower than 10% over the length of the straight line path. The optimization will be terminated if the elapsed time is more than three hours.

This test was run five times using both NSGA-II and Hybrid-Game to compare the computational cost and solution quality.

Design Variables: The trajectory is generated using 20 control points for Start-Target and Target-Start. The y and z-coordinates are variables ($y \in [0,2000]$, $z \in [0,300]$) while the x-coordinates are predefined.

Interpretation of Numerical Results: The computational cost and numerical cost obtained by NSGA-II and Hybrid-Game are compared in Figures 8a and 8b. It can be seen in the last column (AVG: average computational cost of five tests) that the Hybrid-Game takes 13.5 minutes while the computational cost of NSGA-II is 65.2 minutes for five tests (T1 ~ T5). In other words, the Hybrid-Game reduces the computational cost of a standard MOEA such as NSGA-II by 80%.

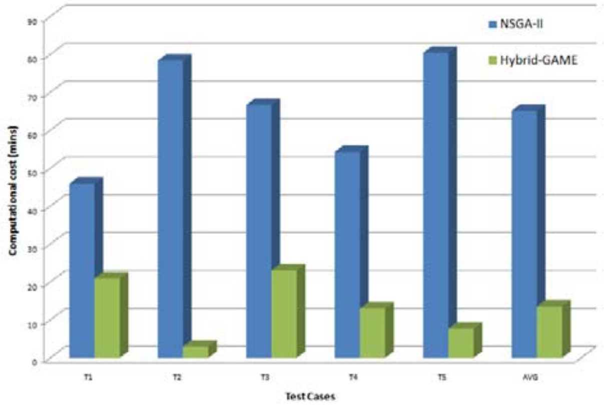

Figure 8a. Comparison of performance between NSGA-II and Hybrid-Game applied to NSGA-II.

Note: AVG represents the average computational cost of NSGA-II and Hybrid-Game for five tests.

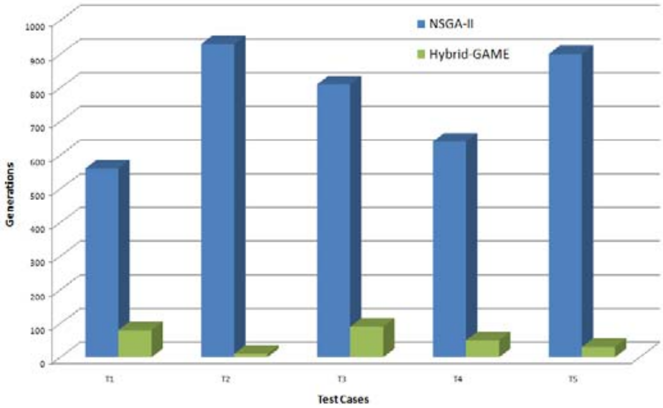

Figure 8b. Comparison of performance (generations) between NSGA-II and Hybrid-Game applied to NSGA-II.

The trajectories obtained by NSGA-II for this test are shown in Figure 9 where the red lines represent the trajectory from the start position to the target position while blue lines are for the trajectory from the target to the start position. The average distance of collision-free trajectories is 4,989 *m* which is only 3.7% higher than the minimum distance ($Path_{min=}$4,989 *m*).

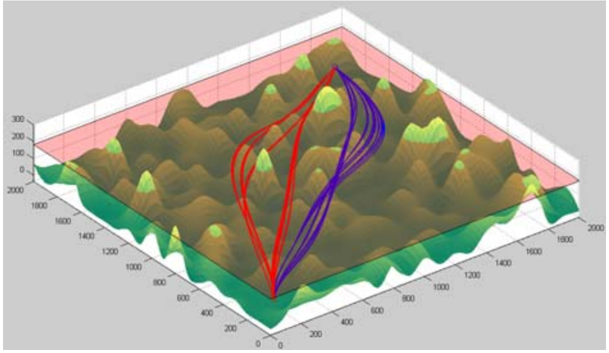

Figure 9. Collision-free trajectories obtained by NSGA-II (Test 1).

Figure 10 shows the paths obtained by the Hybrid-Game approach where the average distance of collision-free trajectories is 4,980 *m* which is only 3.5% higher than the minimum distance ($Path_{min}$).

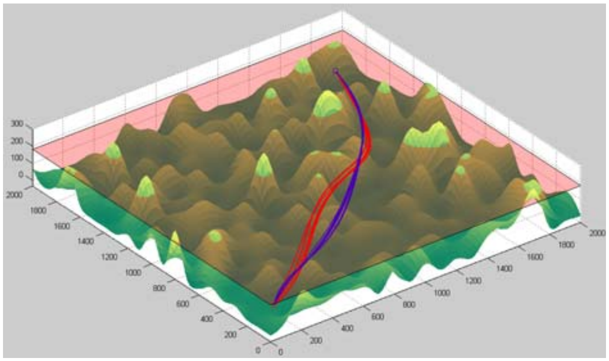

Figure 10. Collision-free trajectories obtained by Hybrid-Game on NSGA-II (Test 1).

Even thought, the paths obtained by Hybrid-Game are not as diverse as the paths obtained by NSGA-II, the computational cost of Hybrid Game is much lower and have the shorter collision-free trajectories.

4.2 TEST2: START-TARGET-END

Problem Definition: The purpose of this test is to minimise the total trajectory length from the start position (900, 100, 150) to the target position (1800, 1800, 100) and then to the end position (100, 900,150). This scenario occurs when we want to survey multiple fields. The fitness functions are;

$$f_1 = \min\left(length(Path_{S-T})\right) + Penalty$$
$$f_2 = \min\left(length(Path_{T-E})\right) + Penalty$$

Subject to;
z-coordinates < $Altitude_{radar}$
z-coordinates > $Altitude_{Obstacles}$

Stopping criteria;

$$\mu_i \geq Pop_{Total} \text{ and } F_{\mu i} \leq Path_{min+10\%}$$
$$\text{or } Elaps_{Time} \geq 3 \text{ hours}$$

The test case considers the same stopping criteria but smaller population which is only 10 members.

<u>*Design Variables*</u>: The trajectory is generated using 20 control points for Start-Target and Target-Start. The *y* and *z*-coordinates are variables ($y \in [0,2000]$, $z \in [0,300]$) while the *x*-coordinates are predefined.

Interpretation of Numerical Results: Figures 11a -c show the trajectories obtained by NSGA-II. The red lines represent the trajectory from the start position to the target position while the blue lines are for the trajectory from the target to the end position. NSGA-II has failed to find collision-free trajectories for Start-Target and Target-End. It can be seen that the trajectories obtained by NSGA-II collide with a fixed obstacle (Section-A) near the target position as shown in Figure 11b -c.

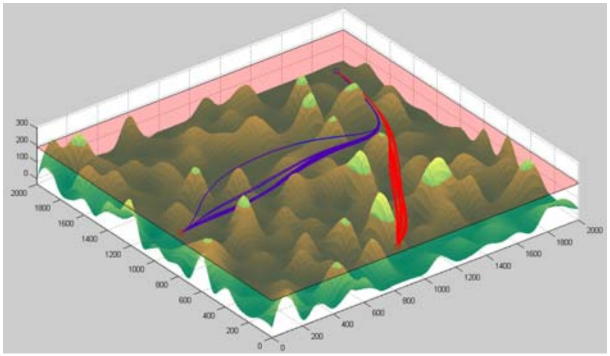

Figure 11a. Trajectories obtained by NSGA-II (Test2).

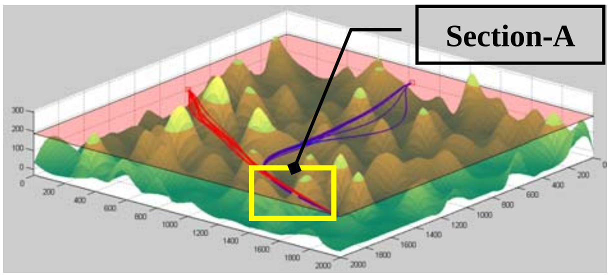

Figure 11b. Collision to the obstacle (Section-A).

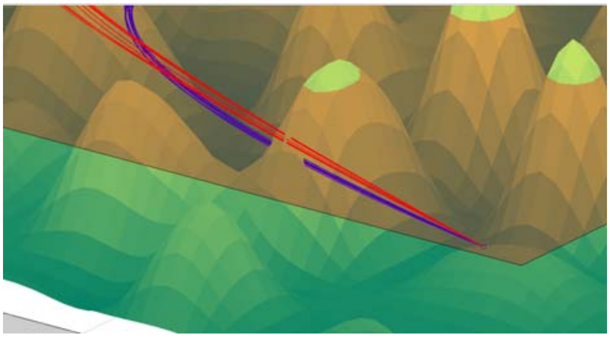

Figure 11c. Zoomed Section-A (Figure 11b).

Figures 12a and 12b show the collision-free trajectories obtained by Hybrid-Game. The average distance of trajectories of Hybrid-Game is 4,182 m which is only 8.6% longer than the minimum distance ($Path_{min}$).

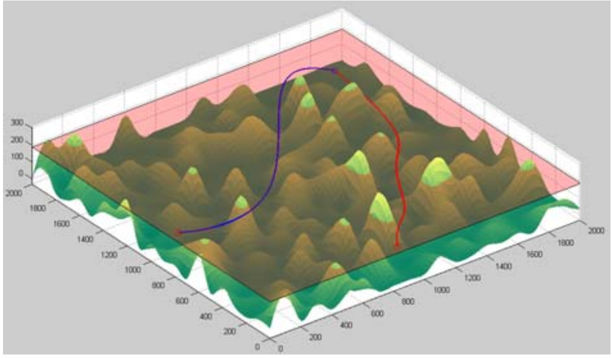

Figure 12a. Trajectories obtained by Hybrid-Game on NSGA-II (Test 2).

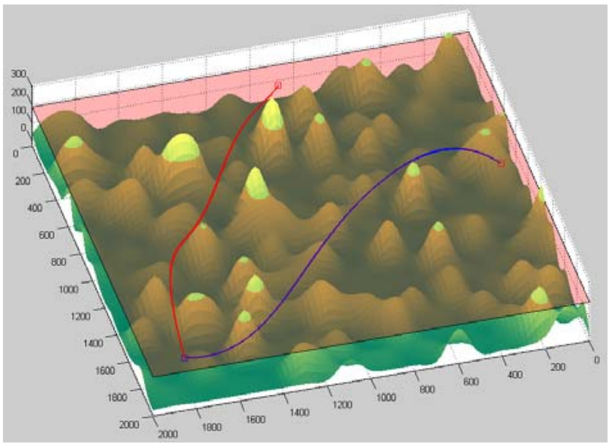

Figure 12b. Trajectories in another view point (Figure 12a).

Once these collision-free trajectories are obtained the next task is to translate them as way points on the UAS autopilot. The optimal solutions are very important since they are not only representing the shortest distance path ways but also collision-free trajectories.

5 Conclusions

This paper illustrated the benefit of coupling a Hybrid-Game strategy to a Multi-Objective Evolutionary Algorithm optimiser. The numerical techniques NSGA-II and Hybrid-Game on NSGA-II were compared in terms of performance efficiency and solution quality for a Mission Path Planning (MPP) problems. The coupling to NSGA-II was presented as it represents one of the well know MOEA optimisers. Nonetheless the Hybrid game strategy could be applied to other methods to improve their convergences. In this paper we considered off-line only applications future work will focus on on-line algorithms, the Nash-Game is fast enough to find on-line collision-free trajectories. Ongoing work focuses on exploring other trajectory generation techniques rather than the Bezier curves, work is also underway to define a real geographic terrain, other aircraft in the same region and how to input the trajectory as waypoints into the UAS autopilot.

References

[Deb et al., 2000] Deb, K., Pratap, Amrit., Agarwal, S., Meyarivan, T., A Fast Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): p. 182-197, 2002.

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

[Deb and Agrawal, 1995] Deb, K., Agrawal, R. B., Simulated binary crossover for continuous search space. *Complex Systems*, 9(2): p. 115–148, 1995.

[Lee, 2008] Lee, D.S., Uncertainty Based Multiobjective and Multidisciplinary Design Optimization in Aerospace Engineering, The Univ. of Sydney, Sydney, NSW, Australia, section 10.7, p.p. 348-370, 2008.

[Lee et al., 2008a] Lee, D.S., Gonzalez L.F., Srinivas, K. and Periaux, J., Robust Evolutionary Algorithms for UAV/UCAV Aerodynamic and RCS Design Optimisation, *An International Journal Computers and Fluids*. Vol 37. Issue 5, pages 547-564, ISSN 0045-7930, 2008.

[Lee et al., 2008b] Lee D.S, Gonzalez, L.F, Srinivas, K. and Periaux, J., 2008, Robust Design Optimisation using Multi-Objective Evolutionary Algorithms, An International Journal Computers and Fluids. Vol 37. Issue 5, pages 565-583, ISSN 0045-7930, 2008 .

[Periaux et al., 2006] Periaux, J., Gonzalez, L.F., Whitney, E.J., Srinivas, K., MOO Methods for Multidisciplinary Design Using Parallel Evolutionary Algorithms, Game Theory and Hierarchical Topology: Practical Application to the Design and Optimisation of UAV Systems (Part 1) Von Karman Institute (VKI) Lecture Series, Introduction to Multidisciplinary to Optimization and Multidisciplinary Design: Applications to Aeronautics and Turbomachinery, March 6-10, 2006.

[Sefrioui et al., 2000] Sefrioui, M., Periaux, J., 2000, Nash Genetic Algorithms: Examples and Applications. Proceedings of the 2000 Congress on Evolutionary Computation CEC00, IEEE Press, La Jolla Marriott Hotel La Jolla, California, USA, ISBN:0-7803-6375-2, pg :509-516, 2000.

[Tang et al., 2005] Tang, Z., Périaux, J., Désidéri, J.-A., 2005, 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, p. 487-493, 19-22 October 2005.

[Wang and Periaux 2001] Wang, J. F. and Périaux, J., Multi-Point Optimization using GAs and Nash/Stackelberg Games for High Lift Multi-airfoil Design in Aerodynamics. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on Vol. 1, Issue 2001 Page(s):552 – 559, 2001.