Lee, D-S. and Gonzalez, L. F. and Periuax, J. and Bugeda, G. (2009) ***Design optimisation using advanced artificial intelligent system coupled to hybrid-game strategies.*** In: Proceedings of the 3rd International Workshop on Artificial Intelligence in Science and Technology (AISAT 2009), 23 - 24 November 2009, University of Tasmania, Hobart, Tasmania.

Design Optimization using Advanced Artificial Intelligent System Coupled to Hybrid-Game Strategies

D.S. Lee[1], L.F. Gonzalez[2], J. Periaux[1] and G. Bugeda[1]

[1]*International Center for Numerical Methods for Engineering (CIMNE), UPC, Barcelona, Spain.*
E-mail: ds.chris.lee@gmail.com, jperiaux@gmail.com, bugeda@cimne.upc.edu
[2]*Engineering Systems, Queensland University of Technology, Brisbane, Australia.*
E-mail: felipe.gonzalez@qut.edu.au

Abstract

One of the main aims in artificial intelligent system is to develop robust and efficient optimisation methods for Multi-Objective (MO) and Multidisciplinary Design (MDO) design problems. The paper investigates two different optimisation techniques for multi-objective design optimisation problems. The first optimisation method is a Non-Dominated Sorting Genetic Algorithm II (NSGA-II). The second method combines the concepts of Nash-equilibrium and Pareto optimality with Multi-Objective Evolutionary Algorithms (MOEAs) which is denoted as Hybrid-Game. Numerical results from the two approaches are compared in terms of the quality of model and computational expense. The benefit of using the distributed hybrid game methodology for multi-objective design problems is demonstrated.

1. Introduction

One of the main purposes of Multi-Objective (MO) or Multidisciplinary Design Optimisation (MDO) using Evolutionary Algorithms (EA) is to develop effective and efficient optimisation techniques in terms of computational cost and solution quality [1-4]. This paper investigates two different game strategies for multi-objective design optimisation; the first method is one well known MOEA; the Non-Dominated Sorting Genetic Algorithm NSGA-II [4]. The second optimisation method called Hybrid-Game can be coupled to any MOEAs; in this case NSGA-II is coupled. The method hybridises the concept of Nash equilibrium [5, 6] and Pareto optimality [4, 7]. The Hybrid-Game method consists of several Nash-Players and one Pareto-Player. Nash-Players optimise local criteria using their own strategy to accelerate the searching speed for global designs which are seeded to the Pareto-Player. The evolutionary optimisation methods NSGA-II and NSGA-II with Hybrid-Game are applied to mathematical multi-objective design problem. Results from both optimisation techniques are compared in terms of design quality and computation expense. The rest of paper is organised as follows; Section 2 presents the methodology, Section 3 considers benchmark multi-objective mathematical test problems. Conclusions and forthcoming work are described in Section 4.

2. Methodology

In this section, two evolutionary optimisation methods NSGA-II and NSGA-II with Hybrid-Game are presented. The first method NSGA-II is a modified version of a well known non-domination based genetic algorithms, and NSGA to have a better sorting algorithm, incorporates elitism. NSGA-II uses Pareto tournament to produce Pareto non-dominated solutions. In the second method NSGA-II is hybridised 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) [8] and polynomial mutation [9]. As a reference, Figure 1 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 with its design bounds, and objective/fitness functions.
Step2: Initialise random population

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.

endwhile

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 based on the non-domination rank of the individual. If individuals have the same non-domination rank then individual with large crowding distance will be selected.

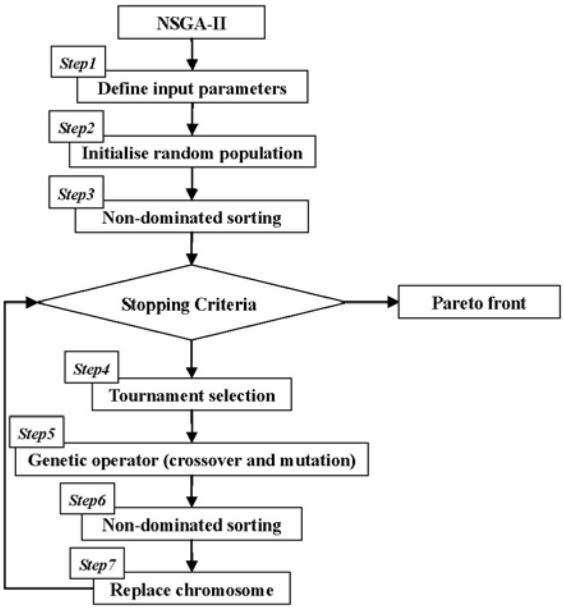

Figure 1. Algorithm for NSGA-II.

2.2. Hybrid-Game applied to NSGA-II

Traditionally, Pareto and Nash games are considered independently when solving a MO problem. In this work, the Hybrid Nash –Pareto approach is considered and detailed in Reference [10]. The Hybrid-Game consists of several Nash-Players corresponding to each objective 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. The reason for this hybrid game implementation of Nash-game coupled to Pareto optimality is to accelerate the search for one of the global solution. The elite design from each Nash-Player will be seeded to a Pareto-Player at every generation and hence it can simultaneously produce Nash-equilibrium and Pareto non-dominated solutions. The algorithm of NSGA-II with hybrid game is shown in Figure 2. The eight main steps are;

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

Step2: Initialise three random populations one for Pareto-Player, one for Nash-Player1 and one for Nash-Player2.

Step2-1: Transfer elite design variable which corresponds to Nash-Player2 objective from the Pareto-Player to the Nash-Player1

Step2-2: Transfer elite design variable from the Nash-Player1 to the 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: Perform genetic operations in each population which consists of crossover and mutation to generate an offspring population.

Step5-1: Seed the elite design from Nash-Player1 and Nash-Player2 to the chromosome of Pareto-Player as a first offspring of each generation.

Step5-2: Send and use elite design on 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: Take the best non-dominated solutions (from *Step 6*) based on their rank and crowing distance and replace to new Parent population on the Pareto-Player.

Step8: Update the elite design obtained by Nash-Player1 and Nash-Player2

endwhile

For example, if a problem considers two objectives ($f_1 = x^2 y$, $f_2 = xy^2$) to minimize 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 use x and y as design variables to minimize both f_1 and f_2 while Nash Player1 will only use x to minimize f_1 and having design variable y_{elite} fixed by Nash-Player2. Nash-Player2 will only use y to minimize f_2 using x_{elite} fixed by Nash-Player1.

In *Step2*, the Pareto-player initialises a random population for f_1 and f_2, and sends the best elite design variable (y_{elite}) that minimizes f_2 to the Nash-Player1. Nash-Player1 initializes a random population using this y_{elite} and sends the elite design value (x_{elite}) that minimizes f_1 to Nash-Player2. Nash-Player2 initializes its random population using this x_{elite} value from Nash-Player1.

In *Step5-1*, the Pareto-Player uses elite design variables (x_{elite}, y_{elite}) and evaluate if only if the first offspring of each generation is considered. In addition, this elite design will be replaced if it is not dominated by any candidates/individuals in Pareto-Player. Nash-Players 1 and 2 will use their elite design at each generation (*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-Player, Nash-Player1, Nash-Player2). The Hybrid-Game can employ more than two players if there are more than two objective functions or if the problem considers complex multi-objective design problem as described in Reference [13].

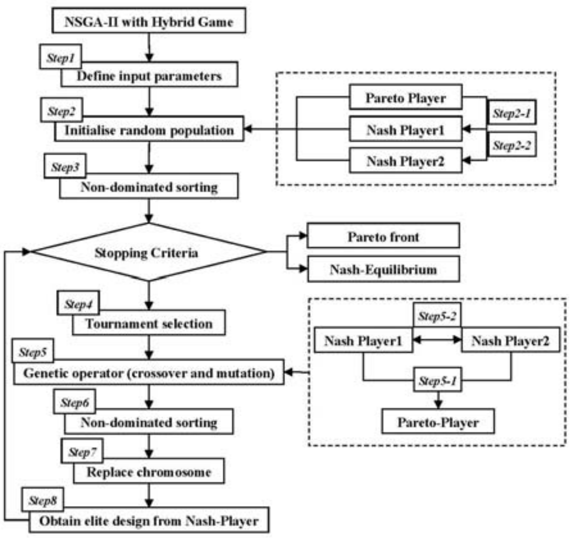

Figure 2. Hybrid-Game on NSGA-II.

3. Multi-Objective Mathematical Design Optimisation

This section compares the true Pareto convergences obtained by NSGA-II and Hybrid-Game on NSGA-II. Both NSGA-II and Hybrid-Game use same optimization parameters; population size = 100, maximum number of generations = [50:500], crossover rate = 0.9 and mutation probability = 1/n where n is the number of decision variables. Six multi-objective mathematical test cases are conducted including non-convex, non-uniformly distributed non-convex, discontinuous, a non-linear goal programming of mechanical design, ZDT4, ZDT6.

3.1. Non-Convex MO Design

This problem which is described in Reference [7] considers minimization of equations (1) and (2). Random solutions (100,000) are shown in Figure 3.

$$f_1(x_1) = 4x_1 \qquad (1)$$

$$f_2(x_1, x_2) = g(x_2) \cdot h\left(f_1(x_1), g(x_2)\right) \qquad (2)$$

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

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

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

$$\alpha = 0.25 + 3.75\left(g(x_x) - 1\right)$$

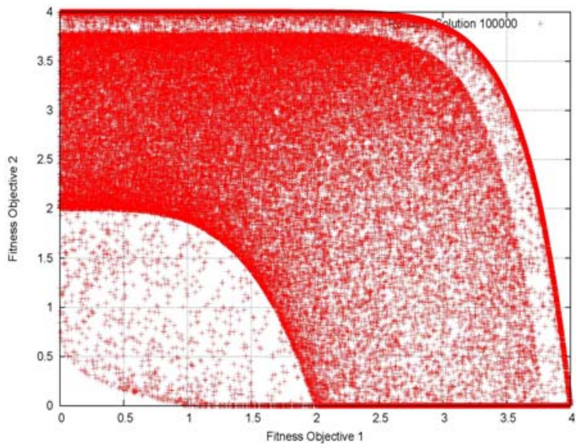

Figure 3. Random solutions (100,000 points).

Figure 4 compares the convergence obtained by NSGA-II and Hybrid-Game coupled to NSGA-II. The optimization is stopped after 50 generations with a population size of 100. It can be seen that the NSGA-II will require more function evaluations (marked with red circle) while the Hybrid-Game has capture the true Pareto front.

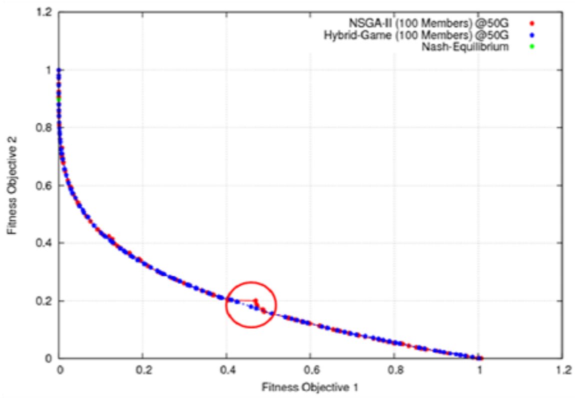

Figure 4. Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 50 generations.

3.2. Non-Uniformly Distributed Non-Convex Design

This problem defined in Reference [7] considers a non-uniformly distributed non-convex problem. It is an extended version of a non-linear problem where the objective is to minimise equations (3) and (4). Random solutions are shown in Figure 5.

$$f_1(x_1) = 1 - \exp(-4x_1)\sin^4(5\pi x_1) \quad (3)$$

$$f_2(x_1, x_2) = g(x_2) \cdot h(f_1(x_1), g(x_2)) \quad (4)$$

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 6 compares the convergence obtained by NSGA-II and Hybrid-Game coupled to 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 Hybrid-Game has already capture the true Pareto front.

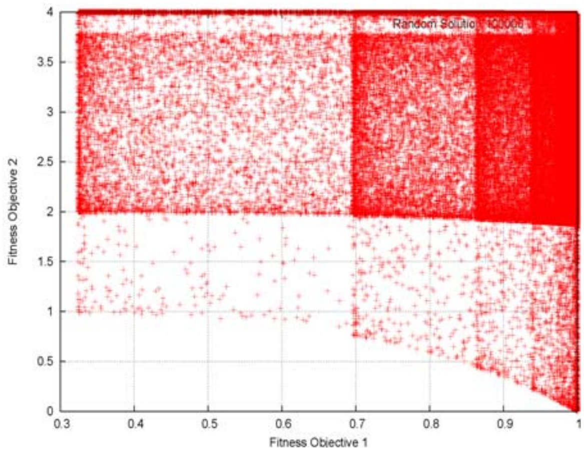

Figure 5. Random solutions (100, 000 points).

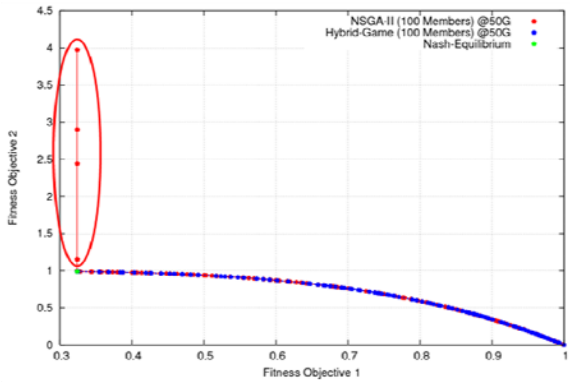

Figure 6. Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 50 generations.

3.3. Discontinuous MO (TNK) Design

The problem TNK proposed in Reference [11] considers minimisation of equations (5). Random solutions are shown in Figure 7.

$$f_1(x_1) = x_1 \text{ and } f_2(x_2) = x_2 \quad (5)$$

Subject to

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

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

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

Figure 8 compares the convergence obtained by NSGA-II and Hybrid-Game coupled to 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.

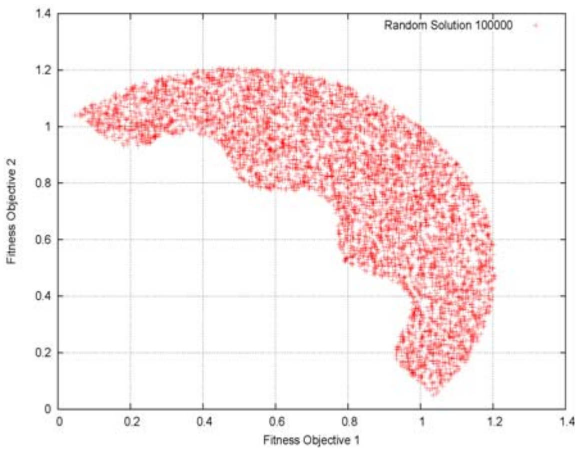

Figure 7. Random solutions (100,000 points).

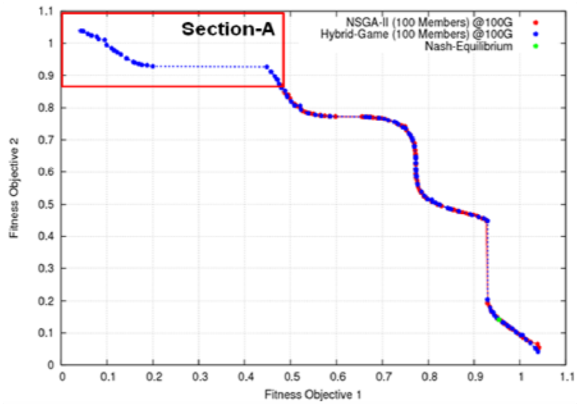

Figure 8. Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 100 generations.

3.4. ZDT6

This problem is designed by Zitzler, Deb and Thiele (ZDT) [12] is formulated by using Eq. (6)- (8).

ZDT6:

$$f_1(x_1) = 1 - \exp(-4x_1)\sin^6(6\pi x_1) \quad (6)$$

$$g(x_2) = 1 + 9\left(\left(\sum_{i=2}^{m} x_i\right)\Big/(m-1)\right)^{0.25} \quad (7)$$

$$f_2(f_1, g) = 1 - (f_1/g)^2 \quad (8)$$

where $m = 10$, $x_i \in [0,1]$.

Figures 9- 11 compare the ZDT6 convergences at 100, 300 and 500 generations obtained by NSGA-II and Hybrid-Game on NSGA-II. The optimization is stopped after 500 generations with a population size of 100.

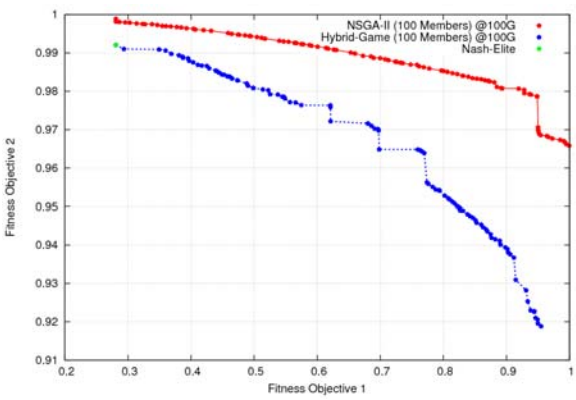

Figure 9. Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 100G.

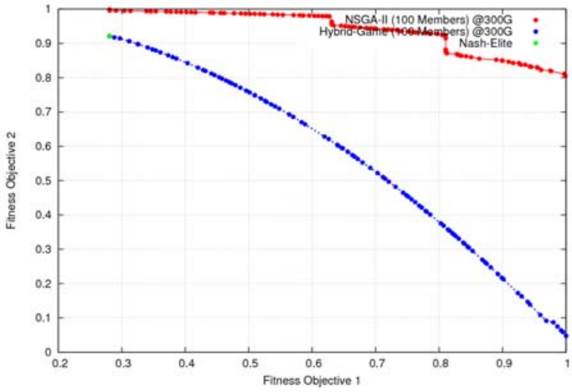

Figure 10. Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 300G.

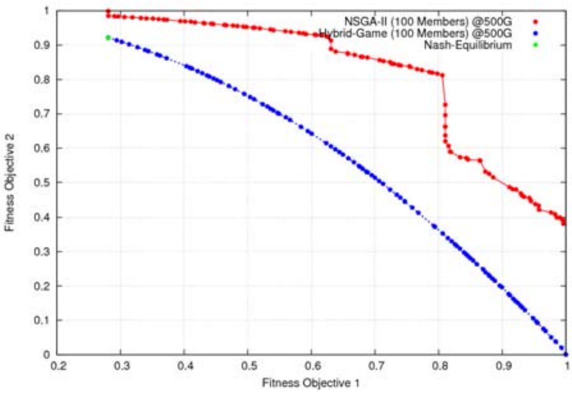

Figure 11. Pareto front obtained by NSGA-II (red dots) and Hybrid-Game (blue dots) after 500G.

As shown in Figure 11, the NSGA-II needs to run more function evaluations to find true Pareto front for ZDT6 while the Hybrid-Game coupled to NSGA-II successfully captures the true Pareto front.

4. Conclusions

The optimisation techniques NSGA-II and NSGA-II with Hybrid-Game are implemented and their numerical results are compared in terms of performance efficiency and solution quality. Results from practical test cases show the broad applicability of Hybrid-Game and represent the benefit of using the combination of Nash and Pareto-game strategies. These methods provide alternative choices to design engineer for multi-objective design optimisation problems including linear or non-linear. Ongoing work focuses on coupling the Hybrid-Game technique to a robust design technique Hierarchical Asynchronous Parallel Multi-Objective Evolutionary Algorithm (HAPMOEA) for time consuming robust multi-objective/multidisciplinary design optimisation problems.

5. References

[1] D.S. Lee, L.F. Gonzalez, J. Periaux, K. Srinivas, *Evolutionary Optimisation Methods with Uncertainty for Modern Multidisciplinary Design in Aerospace Engineering*, 100 Volumes of 'Notes on Numerical Fluid Mechanics' Heidelberg: Springer-Berlin, ISBN 978-3-540-70804-9, pages 271-284, Ch. 3. 2009.

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

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

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

[5] M. Sefrioui, and J. Periaux, 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.

[6] Nash, J., *Non-cooperative Games*. The Annals of Mathematics 54. pg 286-295.

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

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

[9] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26(4): p. 30–45, (1996).

[10] D.S. Lee, *Uncertainty Based Multiobjective and Multidisciplinary Design Optimization in Aerospace Engineering*, The Univ. of Sydney, Sydney, NSW, Australia, section 5.3, p.p. 112-121, 2008.

[11] K. Deb, "Nonlinear goal programming using multi-objective genetic algorithms". *Journal of the Operational Research Society*, 52(3), pp 291-302, 2001.

[12] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[13] D.S. Lee, L. F. Gonzalez, J. Periaux and K. Srinivas. Coupling Hybrid-Game Strategies With Evolutionary Algorithms For Multi-Objective Design Problems In Aerospace. Evolutionary Methods For Design Optimization And Control. Eds. T. Burczynski and J. Périaux,. (EUROGEN09). Cracow, Poland. 15-17 June, 2009.