A Hybrid Multi-Objective Genetic Algorithm for Topology Optimization

K. R. Olympio*
The Pennsylvania State University
Aerospace Engineering Department
229 Hammond Building
University Park, PA16802

ABSTRACT

A new tool is developed in order to solve computationally expensive multi-objective topology optimization problems related to the design for flexible active and passive skins for morphing aircraft. The approach used is based on a multi-objective genetic algorithm coupled with a local search algorithm to create a hybrid multi-objective algorithm. The ability of the developed algorithm to find efficiently Pareto fronts of problems with two and three objectives is evaluated using three test problems. A multi-objective topology optimization problem related to the design of flexible skins is then solved as a proof of concept for more advanced models of flexible skins. It is shown that the hybrid multi-objective algorithm performs significantly better than the multi-objective algorithm it is based on. Additionally, the algorithm is able to solve a larger number of problems. The topology optimization results gives some very promising solution and the approach used lays the ground-work for more advanced topology optimizations.

Keywords

hybrid multi objective genetic algorithm,NSGA2, topology optimization, morphing aircraft, flexible skin,

1. INTRODUCTION

By design, typical aircraft usually perform very well in one flight configuration only (high speed maneuver, climb, dash, cruise or loiter) and present poor to acceptable performance in other flight configurations (Fig. 1). In the recent years, the high demand for better and multi-mission aircraft has lead design engineers to develop more versatile aircraft by expanding their flight envelop or increase efficiency and capability of a single aircraft for multiple distinct flight configurations.

*Graduate Research Assistant

It is well known that the performance of an aircraft in distinct flight configurations is highly dependent on the wing airfoil and most importantly on the wing geometry as seen in Fig. 1. Thus, one way to expand an aircraft flight envelop is to change actively the wing area of the aircraft for the different phases of flight as a bird would do (Fig. 2). This approach led to the concept of morphing aircraft or aircraft that can change smoothly and actively shape in flight.

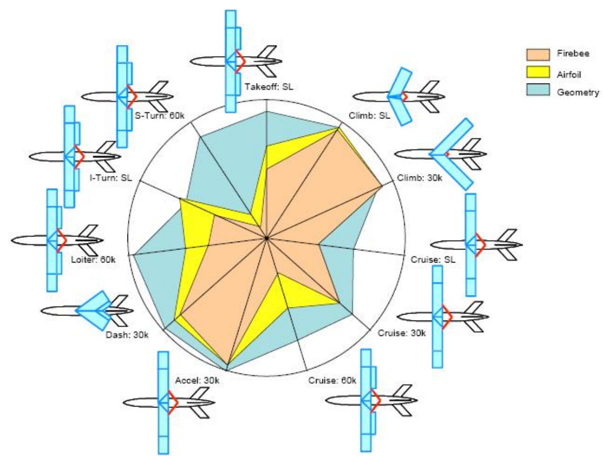


Figure 1: Spider plot comparing predicted performance of the fixed-geometry Firebee, a morphing airfoil Firebee and a morphing planform Firebee (Joshi et. al. [13]). The radius represents the efficiency of the aircraft for the type of flight considered

However, morphing aircraft present some unique challenges. One of them is related to the skin of the morphing structure (generally the wings). Indeed, a typical aircraft skin is either made of a metal alloy or a composite material. These materials allow little elastic deformation and thus are not suitable for morphing aircraft where *large* and *recoverable* deformations are sought. Some of these challenging requirements are:

1. As a component of an aircraft it should be light weight
2. It should have low in-plane stiffness so morphing does not require too much effort.
3. During deformation, the local stresses should stay below the material equivalent yield stresses or elastic

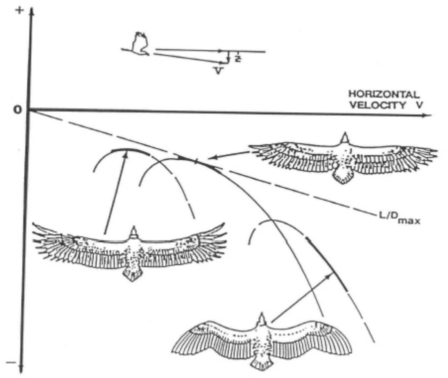


Figure 2: Bird flying in different configurations. The bird deforms its wings depending on the flight speed (Manzo et. al.[16])

limit.

4. The skin should also have low out-of-plane deflection due to the air loads in order to keep a smooth surface and avoid buckling under compressive loading.

No material which simultaneously meet all these criteria currently exist. Some morphing aircraft have used a polyurethane membrane under high pre-strain as a flexible skin (for example MFX-1 from *NextGen Aeronautics*). The disadvantages are that it can require a lot of energy to deform, especially to stretch it, it can wrinkle, creep and it is sensitive to weather conditions. Olympio[18, 19] studied the possibility of using micro-cellular structures and particularly structures made of hexagonal cells for passive 'flexible' skins under one-dimension and two-dimension extensional morphing. Olympio's work mainly involved parametric studies and simple gradient-based optimizations. Recently, Reich et. al[22] used a simple topology approach to find a solution to semi-active flexible skin for morphing aircraft. Their work was based on Sigmund's[20] 99 lines code with two-dimensional linear finite elements and an OC or MMA optimizer. Also a single objective was used. Nevertheless, their work suggested that very different solutions can be obtained by changing only the type of material used.

So far, research efforts on flexible skins showed that micro-structures proved to have a lot of potential, but some issues still remain. The objective of this research effort is to lay the ground-work of a solution procedure where the design space is opened to any micro-structures. The problem is defined by several conflicting objectives and a topology optimization approach is used to find the best micro-structures which meet all the conflicting requirements described above.

In topology optimization, conventional mathematical programming methods such as gradient-based methods are very popular. However these types of algorithm can be difficult to implement in a real multi-objectives problem. Genetic algorithms, due to their structure are more suited to solve these types of problems. Indeed, different individuals can explore a different region of the design space through evolutionary mechanisms and a set of best solutions called the 'Pareto-optimal' set can be found. Obtaining the Pareto-optimal set of solution can prevent the designer to introduce bias (or make early choice in what the solution will look like) as he does not need to use a constraint approach or a weighted sum approach. The designer knowledge is then shifted to defining the problem and selecting the best solution from the 'Approximated Pareto' set of solutions.

In section 2 the hybridization of an efficient multi-objective genetic algorithm (MOGA) is discussed. In the following section (section **??**), the topology optimization problem is defined. In section 4, the performances of the new algorithm are examined using several metrics and test problems. The topology optimization results are discussed in section 5 and a conclusion follows in section 6.

2. HYBRIDIZATION OF A MOGA

This section describes how a Multi-Objective Genetic Algorithm (MOGA) and a local search method are coupled to create a Hybrid Multi-Objective Genetic Algorithm (HMOGA). The motivation for developing a HMOGA is presented first. Then the fundamental elements composing the HMOGA along with discussion about issues related to the hybridization process are exposed. A flow chart summarizing the HMOGA strategy is shown in Fig. 5.

2.1 Motivation

It is a known fact that, for a given problem, a genetic algorithm (GA) is outperformed by any specialized solution scheme tuned to the problem. Moreover, slow convergence of GAs toward an accurate solution is a well-known draw-back that can make them unsuitable for practical engineering applications. This is mainly due to the fact that GAs do not exploit any local information as individuals approach global optima. However, GAs have generally a good global behavior so regions containing global optima are generally found. Local searchers, on the other hand, can converge to an optimum with high accuracy but their global behavior is generally poor. This can prevent them from converging to global optima.

Thus a combination of both methods is very likely to perform better than either method alone[11, 15]. Indeed, if the two methods are well coupled, the local search will provide knowledge to the "blind searcher" or MOGA about the space surrounding individuals ; and the MOGA will expand the access of the local search to the entire design space thanks to its good global behavior. Moreover, GA are usually better suited to handle multiple objectives than typical local search method. By hybridizing a MOGA, we seek to improve its efficiency: converge faster (smaller number of function evaluations) and better (closer to the true Pareto front).

Several researchers have tried to develop a theory of hybridization with simple genetic algorithms (SGA) [8, 9, 15] and in the case of MOGA, the issues are similar. Each of them will be discussed in the following section. Among the most important issues, one needs to:

- Properly balance local and global search so that one method does not outweigh the other but complements it.

- Perform the local search on the right individuals.
- Keep diversity (and avoid premature convergence) even if the problem has big attractors which would take advantage of local search weakness.

The following sections present the algorithmic choices made to develop an efficient HMOGA.

2.2 Global search procedure: MOGA

In the recent years, several efficient MOGA have been developed. They include *Strength Pareto Evolutionary Algorithm 2* ($SPEA2$)[23] and *Non-Dominated Sorted Genetic Algorithm 2* ($NSGA$-II)[3, 6]. Kollat and Reed[14] improved $NSGA$-II to better handle multi-objectives problems with high-order Pareto surfaces. This was achieved with several additional features: ϵ-dominance for population archiving, dynamic population sizing and automatic termination. This 'improved' algorithm was named ϵ-$NSGA$-II. It is worth noting that these additional features do not add to the complexity of parameterizing the MOGA as they are generally part of the problem itself or the set of solutions sought.

Kollat and Reed showed that ϵ-$NSGA$-II can be very competitive compared to other algorithms such as $SPEA2$ for high order multi-objective problems[14]. Moreover, the dynamic population sizing allows to perform substantial search with small populations, thus a large part of the design space can be explored without excessive number of function evaluations which makes the algorithm very attractive for computationally intensive problems. For these reasons, ϵ-$NSGA$-II is chosen to be hybridized in order to create a highly efficient algorithm for solving multi-objective structural optimization problems.

2.3 Local search procedure

One can distinguish two distinct classes of local search procedures: gradient-based methods and non gradient-based methods. Gradient-based methods consist of: Newton's method, Gauss-Newton's method, steepest descent, conjugate gradient and variations of Newton's method. Non gradient-based methods include but are not limited to: downhill simplex method of Nelder-Mead[17], Powell's method, random walk and evolutionary algorithms. However, in this study, evolutionary algorithms will not be considered for hybridizing a MOGA as we seek to incorporate more knowledge into the search and an evolutionary algorithm would simply introduce more "blind search".

Several researchers have used non gradient methods to hybridized a GA. For example, Ishibuchi[12] and Deb[11] used a random walk approach with real and binary coded variable respectively. Other researchers such as Chelouah and Siarry[1] and Fan and Zahara[10] used Nelder-Mead simplex method as their local search procedure. There is no definite answer about which local search procedure to use as no algorithm is better than another for all classes of problems (*No Free Lunch Theorem*). However, the objective of the hybridization is to widen the range of efficiency for both the local search procedure and the GA. Thus, the local search procedure should be adapted to the type of problem considered. For example, if the problem is nonlinear with constraints, gradient based approaches might perform better. If, the problem has some discontinuity or the variables are discrete, a non gradient approach might be more appropriate.

In this study, the local search was used for two main purposes:

1. Help the GA starts with a few nearly optimum solutions. This is especially helpful for constrained problems and problems with weakly dominated front (such as DTLZ6) or multiple "local Pareto fronts" (such as DTLZ3).
2. Improve the quality of the solutions which are already near global optima. So, it seems natural to apply the local search to the best individuals (or archived individuals in the case of ϵ-$NSGA$-II) only.

These two points will be discussed in section 2.5.

2.4 Objective function for the Local search procedure

Local search procedures can be difficult to implement with multiple objectives. Thus, it is necessary to assemble all the objective functions into one single function. Many approaches exist: *weighting sum method*, *Min-max method*, *Global Criterion Method* among others. In this project, only the *weighting sum method* was considered (Eq. 1).

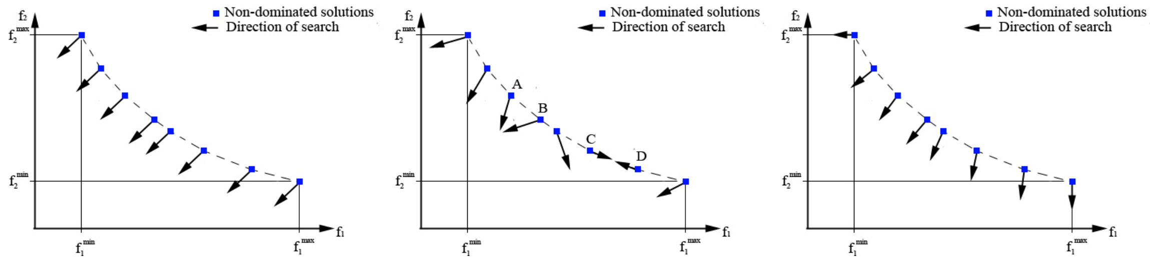
$$\begin{cases} \tilde{f}(x) & = \sum_{i=1}^{M} w_i f_i(x), \text{M: number of objectives} \\ \sum_{i=1}^{M} w_i & = 1, w_i \geq 0 \end{cases} \tag{1}$$

This approach has been implemented successfully in many studies[4, 12] and the weights were defined in various ways. Weights can be(Fig. 3): constant, random or being a function of the location of the solution in the objective space. Constant weights greatly restrict the search as only one direction of improvement is sought. With random weights, the search is statistically performed in all directions but two individuals may also be directed toward the same value (point A and B in Fig. 3) or toward each other (point C and D in Fig. 3) which is likely to decrease diversity and local search efficiency (because several local search runs will give the same answer). Deb[4] suggests using weights according to the position of the individual in the objective space (Eq. 2):

$$w_i = \frac{f_i^{max} - f_i(x)}{f_i^{max} - f_i^{min}} \left[\sum_{k=1}^{M} \frac{f_k^{max} - f_k(x)}{f_k^{max} - f_k^{min}} \right]^{-1} \quad i \in 1, ..., M \tag{2}$$

where f_i^{max} and f_i^{min} are the approximated maximum and minimum values of the i^{th} objective function respectively. They can be found by looking at the individuals in the first front (or 'approximated' Pareto front). With weights calculated as in Eq. 2, one explicitly seeks to move solutions toward the true Pareto, expand the first front and attempt to avoid distinct individuals from converging toward the same values.

Also, to avoid scaling issues, the objective functions are scaled between 0 and 1 as shown in Eq. 3.



(a) Constant weights (b) Random weights (c) Calculated weights

Figure 3: Schematic representation of the choice of weight in defining the objective function for the local search in the case of the two-objective problem

$$\begin{cases} \bar{f}_i(x) & = \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}},\ i \in 1, ..., M \\ \tilde{f}(x) & = \sum_{i=1}^{M} w_i \bar{f}_i(x), \text{M: number of objectives} \\ \sum_{i=1}^{M} w_i & = 1, w_i \geq 0 \end{cases} \quad (3)$$

Additionally, if the problem has some constraints defined by $g_i(s) \leq 0$, large random numbers can be used as penalty parameters and the fitness function becomes:

$$\begin{cases} \tilde{\tilde{f}}(x) & = \tilde{f}(x) + \sum_i \lambda_i \max(g_i(s), 0) \\ \lambda_i : & \text{Penalty parameter for the } i^{th} \text{ constraint} \end{cases} \quad (4)$$

2.5 Combining local and global search

Linking a local search algorithm to a genetic algorithm can be viewed as adding learning capability to individuals in a population. But several issues need to be considered. In sub-section 2.5.3, How the local search is affecting the individual is discussed . In sub-section 2.5.2, we discussed when the local search should be applied. In sub-section 2.5.4, details about the strategy used to limit the computational resources used by the local searcher are exposed.

2.5.1 Preparation

It is a known fact that introducing good solutions in a population will only help a GA to converge. This is the purpose of the *preparation phase* in the HMOGA developed in this study.

However, if the solutions are not close enough to the global optima or the solutions are on local optima, then the algorithm may also fail to converge to a global optimum in a finite number of generations. Thus, the search for good solution must be as successful as possible. In this study, this preparation phase aims at finding the optimal Pareto-set corner. Thus, if the problem has M objectives, each corner will correspond to the minimization of M-1 objectives. As discussed in sub-section 2.4, weights need to be determined to calculate the fitness function the local searcher should minimize. Thus, for each corner sought $i \in \{1, M\}$, $w_j = \frac{1}{M-1}, j \neq i$ and $w_i = 0$.

Additionally, since a local searcher is used, and to increase the probability that a corner of the true Pareto is found, several searches will be done for each corner by changing the initial guess. These initial guesses correspond to specific points in the decision space. Let be N_D the number of equally spaced values taken in each decision variable range, so for the i^{th} decision variable bounded by $[min_i, max_i]$, the values are: $\frac{(max_i - min_i)(k-1)}{N_D - 1}, k \in 1, 2, ..., N_D$. Note that only a small part of the discretized space is considered, thus for a given individual, k is identical for all decision variable. Figure 4 shows an example for 2 decision variables (x_1,x_2) and $N_D = 6$.

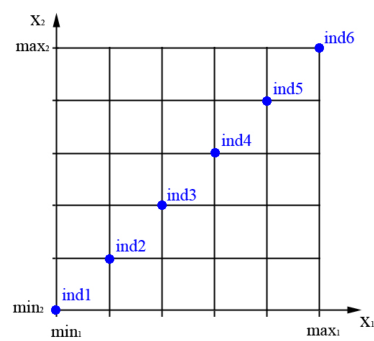


Figure 4: Discretization of the search space for the initial local searches

Therefore, the preparation phase involves a population of $M * N_D$ individuals on which the local search is applied. The best individuals are then included in the initial archive (before injection in the initial population) using ϵ-domination. Note that by doing so, the HMOGA can become more robust, unless the local search reintroduce some stochastic processes (as it is the case for the simple method, for which the initial simplex size is chosen randomly).

This approach is similar to what would be done if only a local search were used in order to check that a global solution is obtained.

2.5.2 Local search gap

In this study, the local search is also apply on the archived individuals created by the MOGA.

Deb studied several hybrid techniques for an engineering shape design[4, 11]. Two main distinctions were made in the way the local search procedure used (hill-climbing) and the MOGA were coupled:

1. *Posteriori approach*: the MOGA is ran for a fixed

number of generation, then the local search method is applied on the solutions obtained using an aggregated objective function.

2. *Online approach*: every solution created by the genetic operators is modified by the local search procedure.

From Deb's results, it appears that the posteriori approach has better diversity and convergence than the online approach. The number of function evaluations was not considered but it is clear that a large part of the computational effort is given to the local searcher in the online approach.

Therefore, in order to decrease the computational resources taken by the local search and from Deb's results[4, 11], the local search is applied only to the archived individual after each run (inter-run local search).

2.5.3 Lamarckian versus Baldwinian

It is also important to know how the local search is affecting the individuals. Two approaches exist: Lamarckian or Baldwinian approach. With the former, the genotypic as well as the phenotypic information are changed by the local search. The improvements are then passed from generation to generation in the evolutionary search. In the latter, only the fitness is changed and the most capable individuals are copied into the next generation.

Thus, a Lamarckian approach can be greedier than a Baldwinian approach. But, it can also cause premature convergence by reducing the diversity in the population[9]. A Baldwinian approach results in function evaluations which are not fully exploited because the genotypic information is lost.

In this study, a Lamarckian approach is used because high efficiency is sought. To prevent premature convergence or loss in diversity, ϵ-domination is performed between the populations before and after the local search is performed. Additionally, the local search can be applied to only a few random individuals $R_A * A$, where A is the size of the archive and R_A is the proportion of the archive which will be "improved" by local search.

2.5.4 Duration of local search

To prevent the local searcher from taking all the computational resources, the number of iterations or function evaluations need to be fixed in case convergence is not reached. Several approaches have been used so far[11, 12, 9, 8] to achieve this.

The local search can terminate in three different ways:

1. The maximum number of iterations N_I has been reached.
2. No more improvement is possible. This happens when a local or global optimum is reached.
3. The termination criteria is met.

In this study, the stopping criteria for the local search is based on the 'efficiency' of the local search and the global search. The 'efficiency' of a search is defined as:

$$e = \frac{f^{old} - f^{new}}{f^{old}} \tag{5}$$

For the local search, f^{old} and f^{new} are simply the values of the fitness function at iteration $k-1$ and k respectively. For the MOGA, f^{old} and f^{new} should be the distance of the first front from the origin of the objective space between two successive generations. So Eq. 5 gives an indication of how much a front has moved toward minimization using either the local search or the MOGA. For example, if $f^{old} = f^{new}$, the algorithm (either MOGA or local search) has made no progress at all and $e = 0$. If $f^{new} = 0$, the algorithm has made maximum progress and $e = 1$. Ideally, the local search and the MOGA would complement each other perfectly over each {MOGA+local search} run and the 'efficiency' would be $e_0 \simeq 1$. So let us assume this is the case then:

$$e_0 = \left(\frac{f^{old} - f^{new}}{f^{old}}\right)_{GA,\infty} + \left(\frac{f^{old} - f^{new}}{f^{old}}\right)_{LS,\infty} \tag{6}$$

Therefore, the local search iterations are stopped when,

1. "Sufficient" improvement was reached:

$$\left(\frac{f^{old} - f^{new}}{f^{old}}\right)_{LS} \geq e_0 - \left(\frac{f^{old} - f^{new}}{f^{old}}\right)_{GA}$$

2. The local searcher is unable to find a good solution fast enough: $k \geq N_I$ where N_I is a hard limit on the maximum number of iterations done by the local search and k is the number of local search iterations.
3. The local search has converged to an optimum.

Additionally, if a local search run is not sufficient enough (according to the previous definition of efficiency), the next inter-run local search is skipped to allow the MOGA to innovate without doing inefficient local searched (and thus consuming function evaluations). If this is the case, it might also be useful to increase the mutation probability in order to innovate more. But this has not been considered in the study.

3. TOPOLOGY OPTIMIZATION PROBLEM

3.1 Problem formulation

To validate the concept of solving a topology optimization problem with a HMOGA, we consider a simple topology optimization problem. The problem consists in finding the best designs of a cantilever beam under tip shear load in terms of minimum mass and minimum deflection.

A material distribution approach is chosen to represent the micro-structure(Fig. 6). Therefore, the design variable can take any values between 0 and 1, but it is preferable to have either 0 or 1 so it seems natural to use binary coded variables. The number of binary variables is equal to the number of elements used in the finite element discretization.

The topology optimization problem can be defined by Eq. 7.

$$\left.\begin{array}{lll} \text{Minimize} & \delta(\vec{F},\vec{s}) & \text{(Tip deflection of the cantilever beam)} \\ \text{Minimize} & m(\vec{s}) & \text{(Mass of the beam)} \\ \text{such that} & s_i \in \{0,1\}, \vec{s}=[s_1,s_2,...,s_N] & \end{array}\right\} \quad (7)$$

where $\vec{F}$ is the loading applied to the skin, $\vec{s}$ is a vector of the design variables and N is the number of design variables. In this problem, the loading is simply a tip shear force.

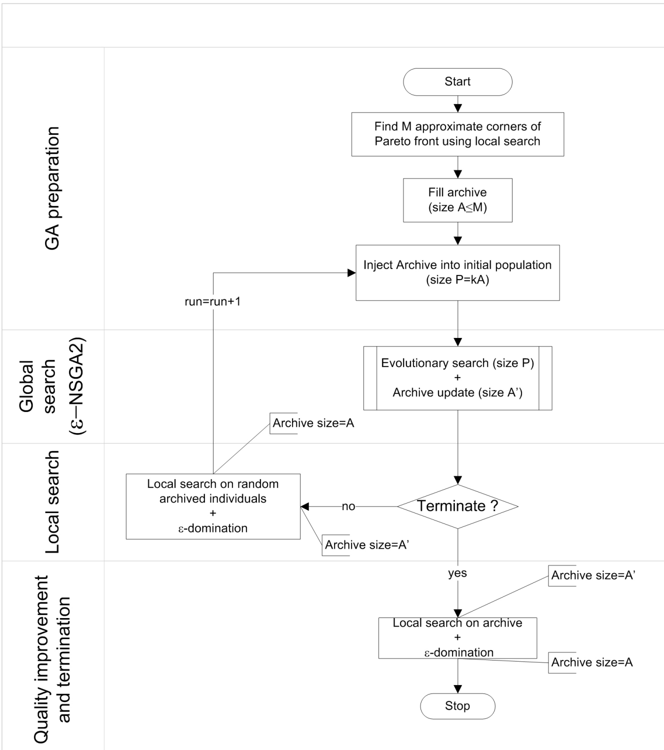


Figure 5: Flow chart describing the hybrid multi-objective algorithm. The global search block corresponds the ϵ-$NSGA2$ algorithm for one single run

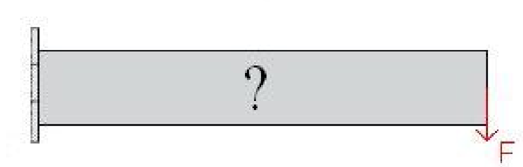


Figure 6: Cantilever beam under shear tip force

For the finite element part of the code, while Q4 elements[1] require less computational effort than higher order elements, they tend to create checkerboard patterns[20]. This is because the stiffness of a checkerboard pattern using this type of element is artificially large due to parasitic shear strain of the element. Thus, if such elements were used, the algorithm would be very likely to converge to such patterns or other meaningless designs. To avoid this issue, Q9 elements[2] should be used because unlike the Q4 element, they do not exhibit shear locking[2]. The disadvantage of this type of element is that the associated computational cost is much higher.

Since only the concept need to be validated in this study, we choose to use Q4 elements as they require less computational resources for the same finite element discretization.

4. TEST PROBLEMS

In this section, the hybrid ϵ-$NSGA$-II is compared to ϵ-$NSGA$-II for some analytical test problems: ZDT1, DTLZ3 and DTLZ6. These functions have been created from the bottom up which makes it easy to define the close form equation of the true Pareto front.[7]

To evaluate the performance of the hybrid algorithm, we use two or three different metrics(depending on the number of objectives): the *convergence metric*, the *diversity metric* and optionally the ϵ-*indicator metric*. The first metric indicates how close to the Pareto-optimal front the solution set is, the second indicates how diverse the final solution set is and the latter represents the smallest distance the first front must be translated to completely dominate the reference set[5]. These metrics will be plotted against the total number of function evaluations to assess speed of convergence and diversity.

Additionally, two local searches (Powell's method and Simplex method) are used in order to evaluate the effect of having an efficient algorithm(Powell's method) and a simple algorithm (Simplex method) which can give less accurate results but with fewer function evaluations[21].

It should also be noted that in this section, only the effect of adding a local search (for the same MOGA parameters) to the MOGA is investigated. Thus, better parameters may exist.

4.1 ZDT1

This is a simple convex problem with 2 objectives. The parameters used to solve the problem with both algorithms are shown in Tab. 1. Since it is a very simple problem, we use it as an opportunity to evaluate the effect of the *preparation phase* so we choose $R_A = 0$ (no inter-local search run).

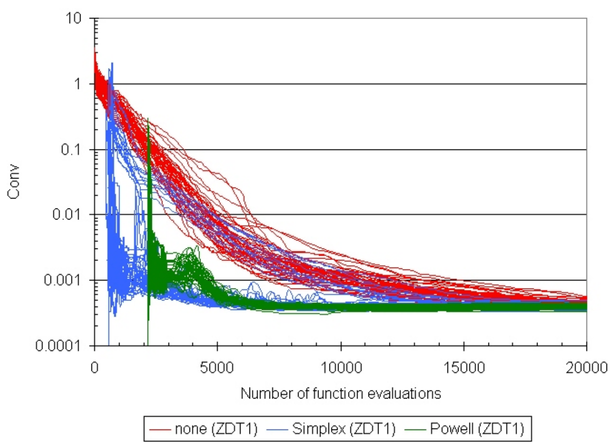
In Figs. 7, the offset that can be observed for the HMOGAs is due to the preparation phase. Fig. 7(a) shows the convergence of the three approaches.Fig. 7(b) presents how diverse the solution is for each approached versus the number of function evaluations.

It can be observed that the HMOGAs converge almost twice

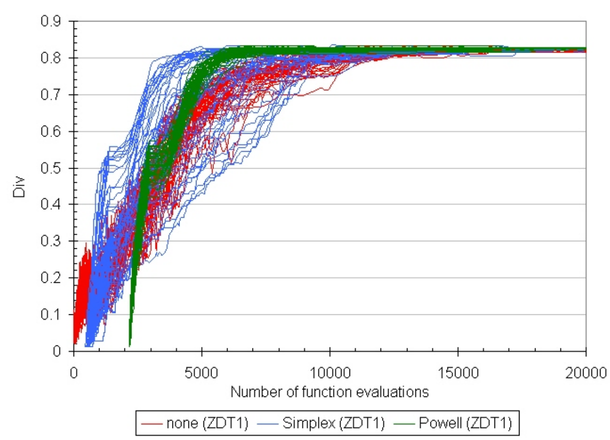
[1]Quadrilateral element with 1 node at each corner
[2]Quadrilateral element with 9 nodes: 1 node at each corner, 1 node at the center of each side and 1 node at the center of the element

	MOGA	HMOGA	HMOGA
Max nb of gen. per run	100	100	100
Max nb of func. eval	20000	20000	20000
Initial population size	8	8	8
Min/Max population size	8/10000	8/10000	8/10000
Injection scaling factor	0.25	0.25	0.25
Crossover probability	1	1	1
Mutation probability	0.033	0.033	0.033
ϵ(all objectives)	0.0075	0.0075	0.0075
R_A	NA	0%	0%
N_I	NA	50	50
N_D	NA	3	3
Local search	NA	Simplex	Powell

Table 1: Parameters used to solve the problem ZDT1. R_A is the proportion of the archive affected by the local search after each run. N_I is the maximum number of iterations allowed for the local search algorithm (it can be different to the number of function evaluations). N_D is the number of starting points for each corner of the Pareto front in the preparation phase



(a) Convergence metric



(b) Diversity metric

Figure 7: Metrics obtained during the solution of ZDT1 problem based on 50 different seeds

as fast as the MOGA. Additionally, the final solutions obtained with the HMOGA are as "diverse" as the solution obtained using the MOGA. These results confirm that by providing good solutions to the MOGA, a significant increase in convergence can be obtained.

However, if the preparation phase only yields local optima, the performances might change drastically. This is investigated with DTLZ3.

4.2 DTLZ3

This function tests the ability of the HMOGA to converge to the global Pareto front. Indeed, this problem has several "local Pareto front" which can render the local search useless. For this problem, Deb et al.[7] suggest using k=10 so we have $n = 12$ decision variables and the problem has $3^k - 1 = 59048$ local Pareto optimal front and only one global Pareto-optimal front.

The parameters used to solve the problem with both algorithms are shown in Tab. 2 and the results are presented in Figs. 8.

	MOGA	HMOGA	HMOGA
Max nb of gen. per run	150	150	150
Max nb of func. eval	100000	100000	100000
Initial population size	12	12	12
Min/Max population size	12/10000	12/10000	12/10000
Injection scaling factor	0.25	0.25	0.25
Crossover probability	1	1	1
Mutation probability	0.033	0.033	0.033
ϵ(all objectives)	0.075	0.075	0.075
R_A	NA	100%	100%
N_I	NA	300	100
N_D	NA	3	3
Local search	NA	Simplex	Powell

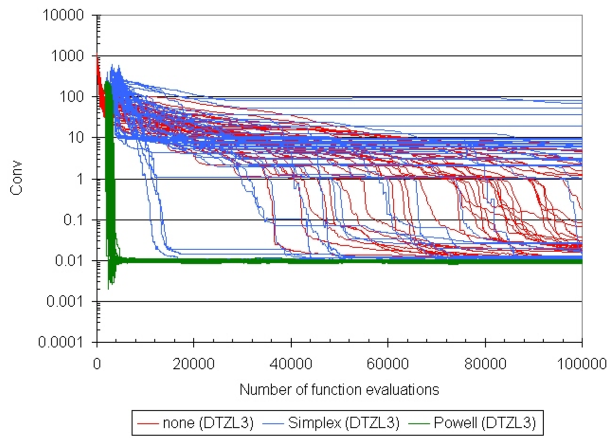
Table 2: Parameters used to solve the problem DTLZ3

Fig. 8(a),8(b) and 8(c) show the convergence metric, diversity metric and ϵ-indicator versus the number of function evaluations for each approach.

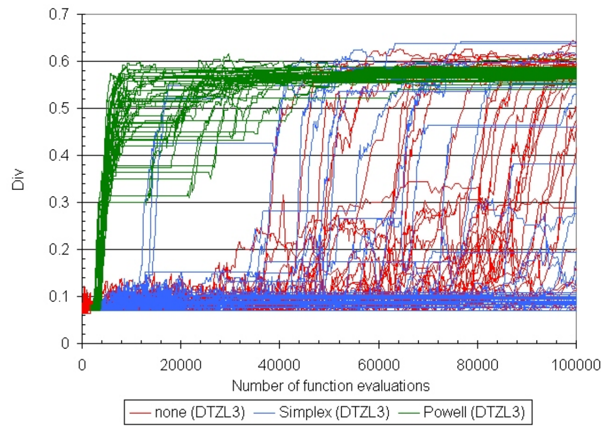
It is seen that when the Simplex approach is used, the algorithm may not necessarily find an global optimum in the preparation phase. In that situation two cases arise: either the MOGA runs manage to improve the solution by mutation or crossover, or the MOGA fails to do so. Note that in the case of DTLZ3, the inter-local search runs can only move the solution to the closest optimum (local or global Pareto front) as the numerous "plateaux" suggest. To avoid failure of the MOGA to converge, it might be useful to dynamically change the mutation probability (not considered here).

If a more powerful local search approach is used (such as Powell's method), good solutions are generally found and the HMOGA converge to the global Pareto front without much difficulty.

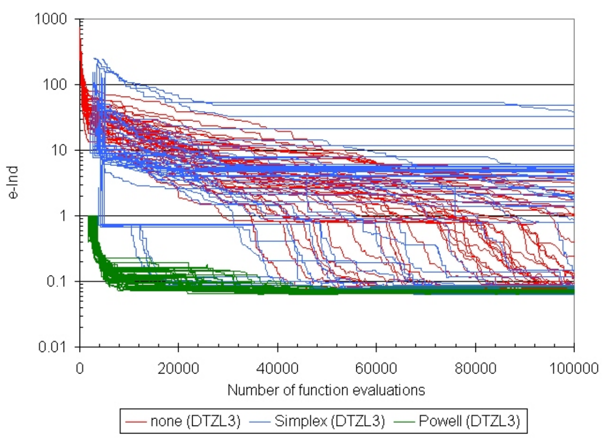
These results suggest that it can be worthwhile to spend a significant amount of resources finding good solutions. If



(a) Convergence metric



(b) Diversity metric



(c) ϵ-indicator metric

Figure 8: Metrics obtained during the solution of DTLZ3 problem based on 50 different seeds

the problem as many local Pareto front, and the local search fail to find at least one global solution, then both algorithms (MOGA and local search) may become powerless. To avoid this issue, dynamic mutation probability might help.

4.3 DTLZ6

This function tests the ability of the HMOGA to converge to a curve in an objective space of dimension greater than 3. For this problem, it is also suggested to take $k = 10$ so there are $n = 12$ decision variables.

The parameters used to solve the problem with both algorithms are shown in Tab. 3 and the results are presented in Figs. 9.

	MOGA	HMOGA	HMOGA
Max nb of gen. per run	250	250	250
Max nb of func. eval	100000	100000	100000
Initial population size	12	12	12
Min/Max population size	12/10000	12/10000	12/10000
Injection scaling factor	0.25	0.25	0.25
Crossover probability	1	1	1
Mutation probability	0.033	0.033	0.033
ϵ(all objectives)	0.0075	0.0075	0.0075
R_A	NA	100%	100%
N_I	NA	100	50
N_D	NA	3	3
Local search	NA	Simplex	Powell

Table 3: Parameters used to solve the problem DTLZ6

Fig. 9(a),9(b) and 9(c) show the convergence metric, diversity metric and ϵ-indicator versus the number of function evaluations for each approach.

Due to the weakly dominated front and the general lack of accuracy of a standard GAs, this problem is often difficult for MOGA. As seen in Figs. 9, not only the convergence metric is high, but the diversity also stays low which suggest clustering of solution on the weakly dominated front. However, with a local search, at least one global solution is often found and convergence is extremely fast (less than 2000 functions evaluations). When this happens, the MOGA only needs to spread the solution on the Pareto front and it can be seen that the diversity increases fast.

This problem clearly illustrate the superiority of a HMOGA over a MOGA because high accuracy is needed.

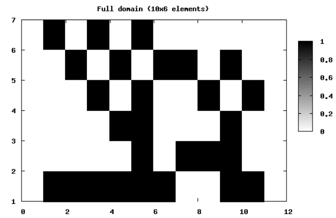
4.4 Summary

The last three test problems illustrated the general better behavior of a HMOGA over a MOGA. However, a HMOGA may become less successful if the *preparation phase* is not carried on properly and the problem has many local solutions. Therefore, it seems that spending a non negligible amount of function evaluations in the preparation phase can be very rewarding.

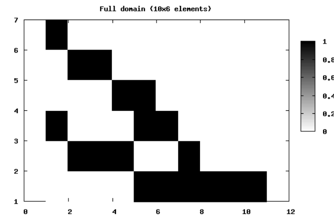
Although other strategies for the preparation phase must exist, the one used in this study (with Powell's algorithm) appears to work well. Thus it will be used in the following topology optimizations.

5. TOPOLOGY OPTIMIZATION RESULTS

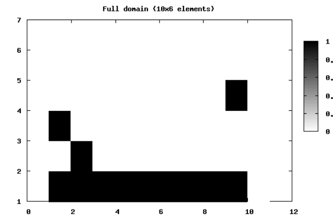
Besides the topology-related issues mentioned in section 3, the algorithm gives mathematically valid designs but meaningless in the real world. To solve this issue, a load path or a similar approach will have to be used in order to obtain



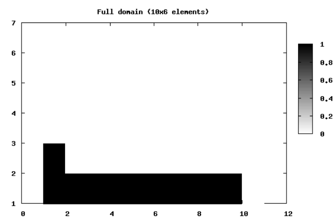
(a) Solution 1: checkerboard pattern due to Q4 elements



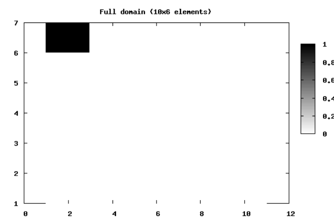
(b) Solution 2: checkerboard pattern due to Q4 elements and bad connectivity



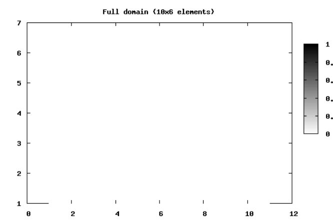
(c) Solution 4: bad connectivity



(d) Solution 5: bad connectivity



(e) Solution 13: bad connectivity



(f) Solution 15

Figure 11: Solutions on the first front. All solutions are mathematically correct, however from the solutions shown, only 1 and 15 are meaningful (based on the assumptions made)

Max nb of gen. per run	150
Max nb of func. eval	300000
Initial population size	12
Min/Max population size	12/10000
Injection scaling factor	0.25
Crossover probability	1
Mutation probability	0.033
ϵ_1	5
ϵ_2	1
R_A	100%
N_I	50
N_D	3
Local search	Powell

Table 4: Parameters used to solve the topology optimization of a cantilever beam

solution with fully connected elements. Moreover, filtering techniques will have to be used in order to make the solutions mesh-independent.

6. CONCLUSIONS

A multi-objective genetic algorithm has been coupled with a local search algorithm. Three test problems have helped to validate the hybridization process. The hybrid algorithm seems to converge to a more accurate solution and in fewer number of function evaluations than the MOGA it is based on provided the local search is efficient.

Additionally, a simple multi-objective topology optimization problem permitted to validate the methodology general topology optimization problems and more spcifically for flexible skin design of morphing aircraft. But some improvements are still needed.

However, since geometric nonlinearities are sought, a nonlinear finite elements still needs to be implemented. Furthermore, classical improvement for genetic algorithm such as multi-processing can be used to increase the computational resources given to the algorithms and in turn increase either the speed or number of iterations done by the algorithm. Additionally, some filtering techniques and connectivity correction will have to be implemented so that meaningful solutions will be found using the approach described in this paper.

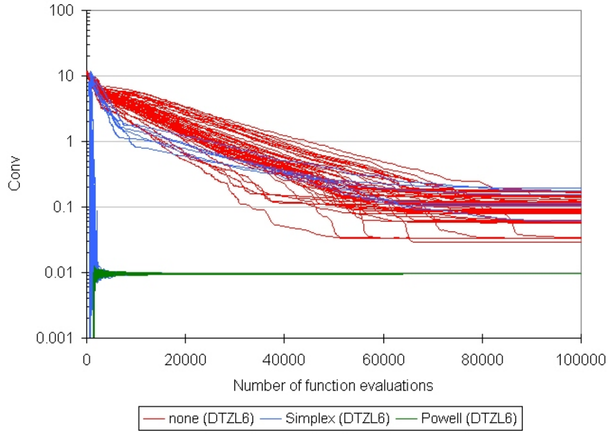
7. REFERENCES

[1] R. Chelouah and P. Siarry. Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. *European Journal of Operational Research*, 148:335–348, 2003.

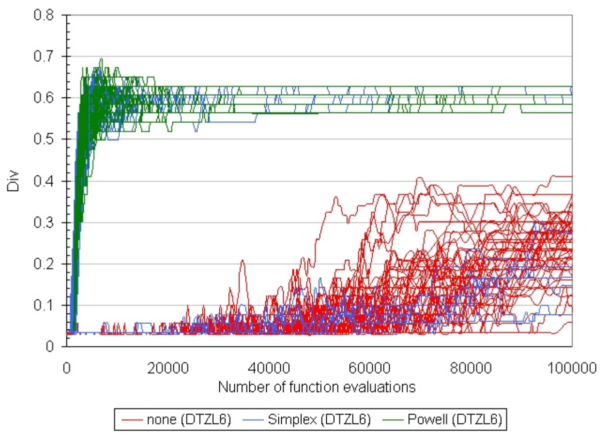
[2] R. D. Cook. *Concepts and applications of finite element analysis.* John Wiley& Sons, Inc., 4th edition, 2002.

[3] K. Deb, S. Agrwal, A. Pratap, and T. meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Parallel Problem Solving from Nature PPSN VI*, pages 849–858, 2000. In M. S. et al. (Ed.), Springer.

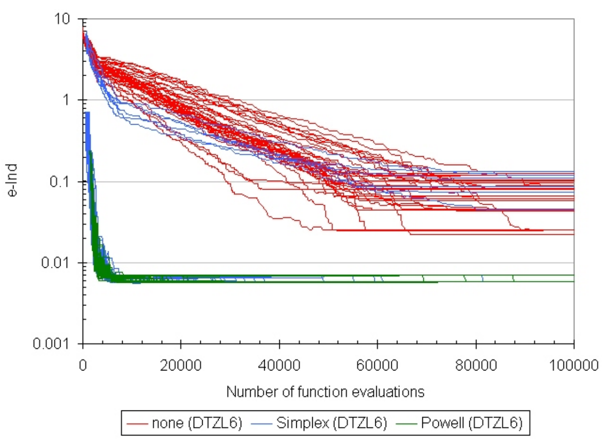
[4] K. Deb and T. Goel. A hybrid multi-objective evolutionary approach to engineering shape design. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 385–399, 2000.



(a) Convergence metric



(b) Diversity metric



(c) ϵ-indicator metric

Figure 9: Metrics obtained during the solution of DTLZ6 problem based on 50 different seeds

[5] K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. Technical Report KanGAL 2002004, Indian Institute of Technology, 2002.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitisit multiobjective genetic algorithms: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 2002.

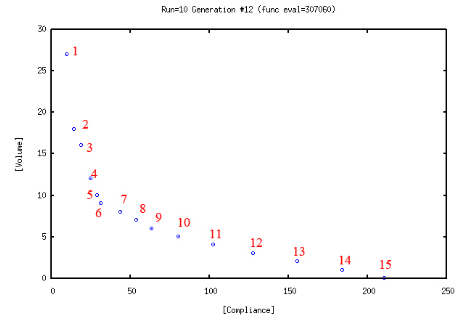


Figure 10: First front obtained for the topology optimization of a cantilever beam

[7] K. deb, L. Thiel, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. TIK-Technical Report 112, Insitut fur Technische Informatik und Kommunikationsnetze, July 2001.

[8] F. P. Espinoza, B. S. Minsker, and D. E. Goldberg. Optimal settings for a hybrid genetic algorithm applied to a groundwater remediation problem. *Proceedings of the World Water and Environmental Resources*, 2001.

[9] F. P. Espinoza, B. S. Minsker, and D. E. Goldberg. Adaptive hybrid genetic algorithm for groundwater remediation design. *Journal of Water Resources Planning and Management*, 131:14–24, 2005.

[10] S.-K. S. Fan and E. Zahara. A hybrid simplex search and particle swarm optimization for unconstrained optimization. *European Journal of Operational Research*, 2006.

[11] T. Goel and K. Deb. Hybrid methods for multi-objective evolutionary algorithms, 2001.

[12] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling-part c: Applications and reviews. *IEEE Transactions on systems, man and cybernetics*, 28(3):392–403, August 1998.

[13] S. P. Joshi, Z. Tidwell, W. A. Crossley, and S. Ramakrishnan. Comparison of morphing wing strategies based upon aircraft performance impacts. *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2004.

[14] J. B. Kollat and P. M. Reed. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Advances in Water Resources*, 29:792–807, 2006.

[15] F. Lobo and D. E. Goldberg. Decision making in a hybrid genetic algorithm. *Proc. 1997 IEEE Conference on Evolutionary Computation*, pages 121–125, 1997.

[16] J. Manzo, E. Garcia, A. Wickenheiser, and G. C. Horner. Adaptive structural systems and compliant skin technology of morphing aircraft structures. *Proc. of SPIE. Smart Structures and Materials 2004: Smart Structures and Integrated Systems*, 5390, 2004.

[17] D. M. Olsson and L. S. Nelson. The nelder-mead

simplex procedure for function minimization. *Technometrics*, 17(1), 1975.

[18] K. R. Olympio. Design of a passive flexible skin for morphing aircraft structures. Master's thesis, The Pennsylvania State University, 2006.

[19] K. R. Olympio. Zero-ν cellular honeycomb flexible skins for one-dimensional wing morphing. *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, April 2007.

[20] M. .P.Bendsoe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer-Verlag, 2nd edition, 2003.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[22] G. W. Reich, B. Sanders, and J. J. Joo. Development of skins for morphing aircraft applications via topology optimization. *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, April 2007.

[23] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: improving the strength pareto evolutionary algorithm. Technical Report TIK-103, Computer Engineering and Networks Laboratory (TIK),Department of Electrical Engineering, Swiss Federal Institute of Technology, ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.