To appear in: International Conference on Intelligent Computing, Sep. 15-18, Shanghai, China

Evolutionary Optimization with Dynamic Fidelity Computational Models

Dudy Lim[1], Yew-Soon Ong[1], Yaochu Jin[2], and Bernhard Sendhoff[2]

[1] Division of Information Systems, School of Computer Engineering
Nanyang Technological University, Nanyang Avenue, Singapore 639798
[2] Honda Research Institute Europe GmbH
Carl-Legien-Strasse 30, Offenbach/Main 63073, Germany

Abstract. In optimization, it is now a common practice to use lower fidelity computational models in place of the original model when dealing with problems with computationally expensive objective functions. In this paper, we present a study on evolutionary optimization with *dynamic fidelity* computational models capable of acclimatizing to localized complexity, for enhancing design search efficiency. In particular, we propose an evolutionary framework for model fidelity control that decides, at runtime, the appropriate fidelity level of the computational model, which is deemed to be computationally less expensive, to be used in place of the exact analysis code as the search progresses. Empirical study on an aerodynamic airfoil design problem based on a Memetic Algorithm with Dynamic Fidelity Model (MA-DFM) demonstrates that improved quality solution and efficiency are obtained over existing evolutionary schemes.

1 Introduction

In science and engineering, computational models are commonly created to simulate and analyze a set of processes or phenomena observed in the physical system so as to gain new insights. Computational models serve as efficient and convenient alternatives for conducting studies on the original system that are otherwise deemed to be too costly or hazardous to construct. They are typically expressed in the form of mathematical equations and then implemented as computer simulation codes. A motivating real world example for us is the aerodynamic aircraft wing analysis and design where various multi-fidelity models ranging from low-fidelity simple-physics models to high-fidelity detailed simulation models have been studied [1–6]. In this field, Navier-Stokes and Euler equations are examples of two computational models with different fidelity, where the latter is obtained by removing the viscosity terms from the Navier-Stokes equation. The term “fidelity” here refers to the extent to which a model is capable to mimic the original physical system of interest. A common assumption is that higher fidelity models are generally more accurate at the expense of a higher computational cost. The complexity and level of details of a physical system comes in many forms, from variable mathematical models [7], variable parametric formulations [4, 8–10], variable operating conditions [11] to variable residual tolerance levels [12].

In recent years, a rising trend in science and engineering is on the use of increasingly high-fidelity accurate analysis codes in the analysis and design process. For example, modern Computational Structural Mechanics, Computational Electro-Magnetics, Computational Fluid Dynamics, and Computational Quantum Mechanics solvers have been shown to be astonishingly accurate. Such analysis codes play a central role in the design process since they aid designers and scientists in validating designs and also enable them to study the effects of altering key design parameters on product performance. However, the move towards using accurate high-fidelity analysis codes results in high computational cost and complexity in the design optimization process, which consequently leads to longer design cycle times. It is worth noting that with the use of high-fidelity computational models, minutes to hours of supercomputer time are often necessary to evaluate or simulate a potential design.

One common approach to enhancing search efficiency in conventional multidisciplinary optimization is via the use of lower fidelity model(s) in place of the original computationally expensive model. From a survey of the literature, it is worth noting that most efforts spend on dealing with optimization problems with computationally expensive model, particularly in the field of surrogate-assisted optimization, have generally focused on the use of data-fitting approximation models, such as Kriging, Radial Basis Function, Multivariate Adaptive Regression Splines, Polynomial Regression, Artificial Neural Networks and several others. Even though data-fitting models are generally easy and computationally inexpensive to construct, they are often disregarded by domain experts since they are generally black-box mathematical models that do not provide meaningful scientific interpretation of the underlying processes that is modeled. In contrast to the use of black-box data-fitting approximation models as a replacement of the computationally expensive model, we seek for alternative variable fidelity computational analysis models in the present work.

In the context of evolutionary optimization, significant studies on enhancing search efficiency via the use of variable fidelity computational analysis models have been made in recent years. It is possible to categorize existing evolutionary algorithms [13] that employ variable fidelity computational models into *1)* non-adaptive, i.e., one or more fixed low fidelity model(s) in tandem with the original expensive computational model, or *2)* deterministic adaptive approaches, i.e., pre-defined adaptation rule(s) is(are) used to manage the variable fidelity models employed. In the first category, [9] employs a fixed lower resolution computational model during the evolutionary design of a communication antenna. Subsequently, the obtained solution is then refined by space mapping. On the other hand, instances of deterministic adaptation can be found in [4, 8, 10]. In [4], the evolutionary search employs computational models of different grid mesh that varies from coarse to fine grids at pre-defined stages of the GA search. In a multi-island GA optimizer [8, 10], multiple subpopulation of individuals are equipped with models of variable fidelity levels. Individuals then migrate across subpopulation which are subsequently evaluated based on the computational model of the respective subpopulation. Last but not least, [7] uses a pre-defined adaptation rule to regulate the proportions of individuals in the population with different fidelity models in the evolutionary search.

In this paper, our objective is to present an investigation on evolutionary framework that employs *dynamic fidelity* computational models for solving problems with computationally expensive objective functions. In particular, we propose a evolutionary framework for model fidelity control that decides, at runtime, the complexity level of the localized computational model to be used in place of the exact analysis code as the search progresses. Based on [13], the proposed evolutionary framework falls under the category of dynamic adaptation. In contrast to non-adaptive and deterministic adaptive approaches, where domain knowledge on the appropriate choice of fidelity level to use or established heuristics/rules that are known to work well on the problems of interest are assumed to be available, dynamic adaptive approaches require minimum intervention of external mechanisms, and have been shown to be more effective in many contexts [14, 15]. Further, we would like to note that the present study represents a first attempt on dynamic adaptation in evolutionary optimization where dynamic localized fidelity computational models have been deployed for solving problems with computationally expensive objective functions.

The remaining of this paper is organized as follows. Section 2 describes a multi-fidelity airfoil computational model used in the present study for aerodynamic design. Subsequently, Section 3 introduces the memetic evolutionary framework with dynamic fidelity computational models for optimizing the computationally expensive aerodynamic design problem. Section 4 summarizes the empirical results on the aerodynamic design problem and analyzes the results with comparison to existing evolutionary optimizers. Finally, Section 5 provides a brief conclusion of this paper.

2 Multi-Fidelity Models in the Aerodynamic Airfoil Design Problem

In this section, we provide an example of multi-fidelity computational models, prevalent in many real-world problems, using the aerodynamic airfoil design problem. For illustration purpose, Fig. 1 depicts an aircraft together with an airfoil, which is simply the cross-section of the aircraft wing (see Fig. 1). Particularly, we consider the parametric design optimization of 2D airfoil structure using a subsonic inverse pressure design problem. In the inverse design problem, the aim is to minimize the difference between the surface pressure P of a given airfoil with the desired pressure profile P_d of a baseline shape. If W is the flow variables and S the shape design variables, the inverse pressure design problem can be formulated as a minimization problem of the form:

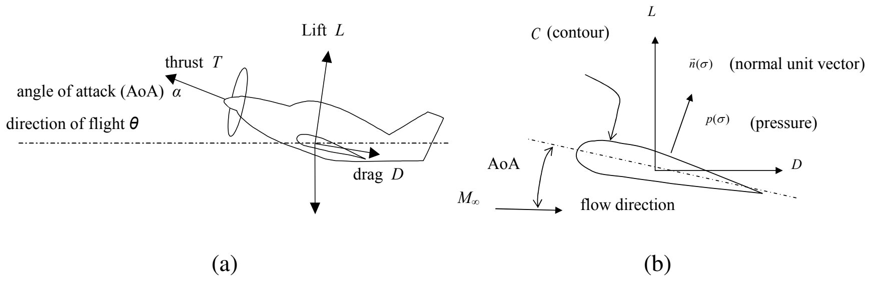
$$f\left(W, S\right) = \frac{1}{2}\int_{wall} \left(P - P_d\right)^2 d\sigma \tag{1}$$

For the purpose of this study, the target pressure profile is generated from the NACA 0015 airfoil[3] as the baseline shape. The airfoil geometry used here is characterized using 24-parameter Hicks-Henne representation[4] as described in Fig. 2. Since we only

[3] The NACA 0015 is one of the four-digit series airfoils. The first two digits ‘00’ implies the none existence of camber. The next two digits, i.e. ‘15’ indicates a 15% thickness to chord length ratio.

[4] The Hicks-Henne representation describes the upper and lower surfaces of the airfoil as linear combinations of a finite number of basis functions.

consider compressible non-viscous flow, a finite-volume Euler solver with body-fitted grid and explicit time-stepping is employed for the purpose of this study. The free-stream conditions in this problem are subsonic speed of Mach=0.5, and angle of attack (AOA)=2.0, corresponding to symmetric pressure profiles on the upper and lower walls.


Lift L
thrust T
angle of attack (AoA) α
direction of flight θ
drag D
L
C (contour)
$\vec{n}(\sigma)$ (normal unit vector)
$p(\sigma)$ (pressure)
AoA
D
M_∞
flow direction

(a) (b)

Fig. 1. (a) An aircraft and (b) an airfoil, the cross-section shape of the wing.

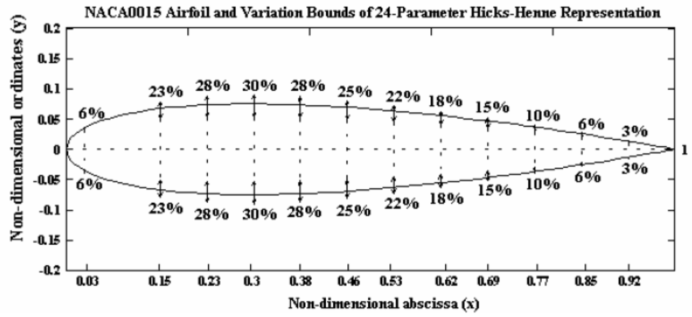


Fig. 2. Airfoil geometry characterized using 24 design variables with the NACA 0015 as baseline.

In practice, the evaluation of a single individual design (refer to equation 1) is an iterative process, usually terminated after a certain residual error threshold or a maximum number of iterations has been reached. Since this is an inverse pressure problem, the residual error here refers to the pressure difference between the respective design and the baseline airfoil and is expected to reach a value close to zero. In the case of limited computational budget, it is also possible to treat the maximum number of iterations as the effective termination condition. Hence, variable fidelity models can be obtained at different levels of residual error tolerance and/or maximum iterations. Generally, lower residual error tolerance and greater number of iterations provide higher fidelity outputs.

The correlation factor (r) and root mean square error ($rmse$) of the models at variable levels of residual error tolerance (i.e., fidelity) to the original high fidelity airfoil analysis model are reported in Figs. 3 and 4, respectively. In this study, a set of design points, obtained from space-filling Design of Experiment (DOE) sampling, are partitioned into separate local clusters based on their proximity in the search space of the airfoil model. The correlation factor, r_k^j, and root mean square error, $rmse_k^j$ for data cluster j at fidelity level k, are defined as follows:

$$r_k^j = \frac{n\sum_{i=1}^{n} f_i\hat{f}_i - \sum_{i=1}^{n} f_i \sum_{i=1}^{n} \hat{f}_i}{\sqrt{[n\sum_{i=1}^{n} f_i^2 - (\sum_{i=1}^{n} f_i)^2][n\sum_{i=1}^{n} \hat{f}_i^{\,2} - (\sum_{i=1}^{n} \hat{f}_i)^2]}}. \tag{2}$$

$$rmse_k^j = \sqrt{\frac{\sum_{i=1}^{n} |f_i - \hat{f}_i|^2}{n}} \tag{3}$$

where n is the number of data in cluster k, f_i is the output from the original analysis code for data i, and $\hat{f}_i$ is the output from the fidelity level k of data i.

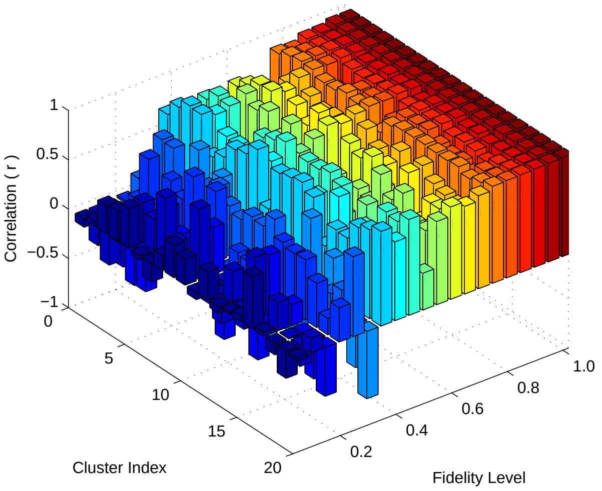


Fig. 3. Correlation factor of the 20 clustered localized models for variable levels of residual error tolerances, while taking the original airfoil analysis code as the reference model (i.e., with a fidelity of 1)

The large $rmse$ values in Fig. 4 highlighted large discrepancies between the low fidelity models and the original analysis code. On the other hand, Fig. 3 displays good fitness landscape correlation between the low fidelity localized models and the original model at early stages of the convergence. In the context of optimization, the absolute fitness of a design is generally non-crucial, rather it is the relative fitness of the solutions that is important in leading the search towards the global optimal solution. In this

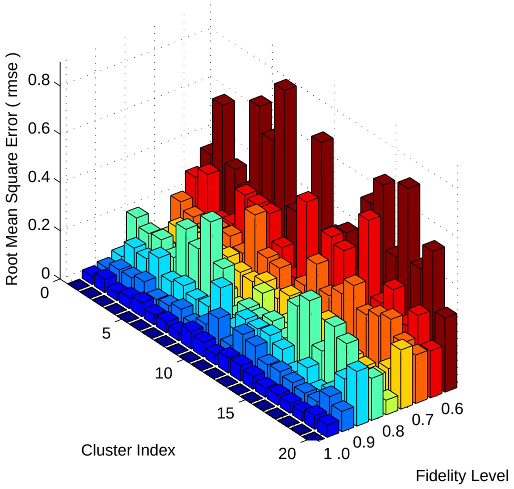


Fig. 4. $rmse$ of the 20 clustered localized models for variable levels of residual error tolerances, while taking the original airfoil analysis code as the reference model (i.e., with a fidelity of 1)

case, this implies the correlation between the predicted fitness values is of utmost importance. For greater details, the reader is referred to [16–18], where it was shown that the correlation metric represents a more useful metric for estimating the quality of an approximation model over one that is based on absolute error.

3 Memetic framework with Dynamic Fidelity Computational Models

Without loss of generality, we consider a general nonlinear programming problem of the following form:

$$\begin{aligned} &\text{minimize: } f(\mathbf{x}), \\ &\text{subject to: } x_i^l \leq x_i \leq x_i^u, \end{aligned} \tag{4}$$

where $i = 1, 2, \ldots, d$, d is the dimensionality of the search problem, and x_i^l, x_i^u are the lower and upper bounds of the i^{th} dimension of vector $\mathbf{x}$, respectively. In this paper, we are interested in cases where the evaluation function of $f(\mathbf{x})$ is computationally expensive and it is desired to obtain near optimal solution on a limited computational budget.

Based on previous analysis in Section 2, here we present an evolutionary memetic framework aimed at improving the efficiency of multi-fidelity evolutionary optimization, which is labeled here as Memetic Algorithm with Dynamic Fidelity Models (MA-DFM). The idea behind the proposed MA-DFM is that the complexity level of the localized computational model to be used in place of the exact analysis code, is dynamically determined at run time as the search progresses. In the process, a user specific correlation factor, η, is utilized to determine the minimum level of fidelity that is considered to match well with the original model. The pseudo codes of the standard MA and MA-DFM used in the present work are outlined in Algorithm 1 and 2, respectively.

Algorithm 1 Standard Memetic Algorithm (MA)

```
**Initialization:** Generate a population of design vectors.
**while** computational budget is not exhausted **do**
    Evaluate all individuals in the population using the exact fitness function.
    **for** each individual $\mathbf{x}$ in the population **do**
        • Apply local search using exact fitness function with probability of $P_{ls}$ and intensity
          of $I_{ls}$ to find an improved solution, $\mathbf{x}_{opt}$.
        • Replace $\mathbf{x}$ with the locally improved solution, i.e. $\mathbf{x} = \mathbf{x}_{opt}$ and $f(\mathbf{x}) = f(\mathbf{x}_{opt})$.
    **end for**
    Apply standard evolutionary operators to create a new population.
**end while**
```

The MA-DFM (refer to Algorithm 2) begins with the initialization of a population of design points. During the database building phase, the search operates as a traditional evolutionary algorithm for the initial few generations. In this stage, only the original analysis code is used as the fitness function while at the same time, all partial results obtained in the evaluation of the computational model are archived in database, $\Im$. Subsequently, the algorithm proceeds to the memetic search phase. Henceforth, for each individual $\mathbf{x}$, an appropriate fidelity level, φ, of the computational model to use in the local learning phase may then be determined based on the m nearest points in database $\Im$. For instance, a user-specified confidence based on correlation, η, may be introduced to obtain the minimum fidelity model to be used in place of the original model in the local search. Upon local convergence based on the low fidelity model, the locally optimized solution x_{opt} is then validated using the original model and replaces the original starting individual if improved solution is attained, in the spirit of Lamarckian Learning (refer to line 10 of Algorithm 2). The entire search cycle is then repeated until the maximum allowable computational budget is exhausted.

It is worth highlighting on the novel use of dynamic localized computational model in the proposed framework. Particularly, local models are used in favor of global models since constructing accurate global models is fundamentally flawed due to the curse of dimensionality [18–21]. Further, this allows more precise estimation on the unique characteristics of the problem landscapes, thus leading to the prediction on the appropriate level of localized model fidelity over the use of the original computationally expensive model.

4 Empirical Study

In this section, we present an empirical study using the inverse pressure problem described in Section 2. We validate the efficacy of the proposed MA-DFM against the

Algorithm 2 MA with Dynamic Fidelity Models (MA-DFM)

```
Initialization: Generate a database, $\Im$ containing a population of designs, archive all exact evaluations made in the database.
while computational budget is not exhausted do
    if database building phase then
        • Evolve the population using standard EA, archive all exact evaluations at different iterations into the database.
    else
        for each individual $\mathbf{x}$ in the EA population do
            • Find $m$ nearest points to $\mathbf{x}$ in database.
            • Based on the $m$ points, find the minimum fidelity level $\varphi$, at which the correlation measure between fidelity level of $\varphi$ and the original analysis code, has reached $\eta$, i.e. $r_{\varphi} \geq \eta$
            • Apply local search on $\mathbf{x}$ using fitness function at fidelity level $\varphi$, $f_{\varphi}(.)$ with probability of $P_{ls}$ and intensity of $I_{ls}$ to find an improved solution, $\mathbf{x}_{opt}$.
            • Replace $\mathbf{x}$ with the locally improved solution, i.e. $\mathbf{x} = \mathbf{x}_{opt}$ and $f(\mathbf{x}) = f(\mathbf{x}_{opt})$.
            • Archive all new exact function evaluations made in database $\Im$.
        end for
        Apply standard evolutionary operators to create a new population.
    end if
end while
```

standard GA, MA, and a non-adaptive MA that employs a fixed fidelity model or MA-FFM in short. In contrast to MA-DFM, note that MA-FFM performs local searches having a pre-defined fixed model of low fidelity (f_{ζ}), where $\zeta < 1$. The common parametric configurations of all 4 schemes used in the present experimental study are summarized in Table 1 and briefly described in what follows.

In the local search procedure, the well-established Feasible Sequential Quadratic Programming (FSQP) method [22] is employed. Although I_{ls} defines the maximum computational budget of each individual in the local learning phase, the actual time incurred do vary according to the fidelity level used. In the present study, the computation cost per evaluation is determined by the fidelity level of the model used, i.e., evaluating a computational model with a fidelity level of 0.7 translates to a compute cost of 0.7 evaluation count. On the other hand, note that each evaluation of the fixed low fidelity in the local search for MA-FFM is performed at a constant cost of ζ, which is here assumed as 0.5 in our experiment setting, which is half the computational expense of the original model. Since model accuracy is highly dependent on the sufficiency of the m data points used for model building, the size of nearest neighboring points used is defined by $(d+1)(d+2)/2$, where d is the dimensionality of the optimization problem and is 24 for the airfoil problem considered here. A maximum computational budget of 5000 function evaluations is used in the experimental study.

The obtained average convergence trends for each of the 4 algorithms, across 10 independent runs, are summarized in Fig. 5. It is also worth noting that in all the algorithms considered, the first 1000 exact evaluations represent the results of the standard GA, hence similar initial search trends are shown in the figure. Nevertheless, the search trends begin to differ after 1000 evaluations where it can be observed that all

Table 1. Setting of experiments for GA, MA, MA-FFM, and MA-DFM.

General Parameters	
Population size (N_{pop})	100
Crossover probability (P_{cross})	0.9
Mutation probability (P_{mut})	0.1
Maximum number of exact evaluations	5000
MA, MA-FFM, and MA-DFM - specific Parameters	
Local search probability (P_{ls})	0.2
Local search intensity (I_{ls})	10 local search iterations
MA-FFM and MA-DFM - specific Parameters	
Database building phase	1000 evaluations
Number of nearest neighbours (m)	(d+1)(d+2)/2
MA-FFM - specific Parameters	
Fixed lower fidelity level used (ζ)	0.5
MA-DFM - specific Parameters	
Minimum correlation required (η)	0.75

the memetic schemes (MA, MA-FFM, and MA-DFM) studied outperform the standard GA significantly at 95% level of confidence, see statistical test given in Table 3. This demonstrates the ability of the MAs in converging to good quality solution more efficiently than a standard GA [23, 24].

Next, note that among the two multi-fidelity MA schemes, while both MA-FFM and MA-DFM employs lower fidelity model as the fitness function, the latter is observed to perform better. The $t-test$ results reported in Table 3, calculated based on statistical data from Table 2 and Fig. 6, confirms this significant difference in performance. Subsequently, the best airfoil shapes and the upper and lower surface pressure coefficients obtained by 4 schemes compared in the experiments are plotted in Figs. 7 and 8, respectively. It is shown from these figures that the best shape/pressure profile obtained by MA-DFM is closer to the shape/pressure profile of the NACA0015 baseline airfoil. The results obtained on the real world aerodynamic problem thus demonstrate the ability of the MA-DFM in model fidelity control that decides, at runtime, the appropriate fidelity level of the computational model to be used in place of the exact analysis code as the search progresses.

Table 2. Mean and standard deviation of GA, MA, MA-FFM, and MA-DFM search at the end of 5000 function evaluations.

Scheme	**Mean ± Std. deviation**
GA	3.5776e-3 ± 9.5973e-4
MA	2.1232e-3 ± 4.9329e-4
MA-FFM	2.6417e-3 ± 5.0709e-4
MA-DFM	1.5131e-3 ± 4.0862e-4

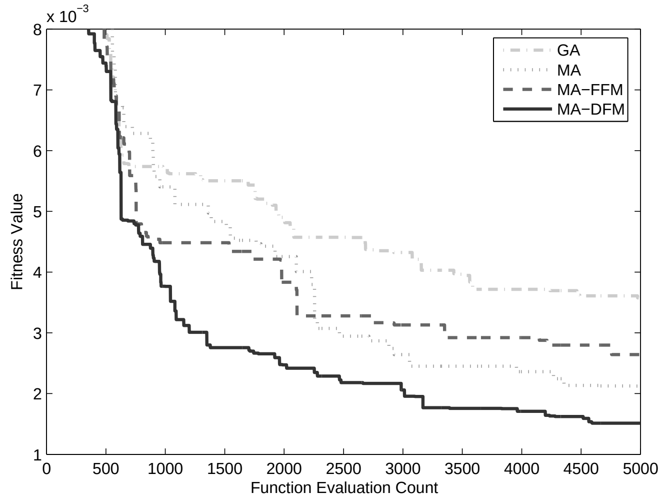


Fig. 5. Search convergence traces of GA, MA, MA-FFM, and MA-DFM on the aerodynamic airfoil problem.

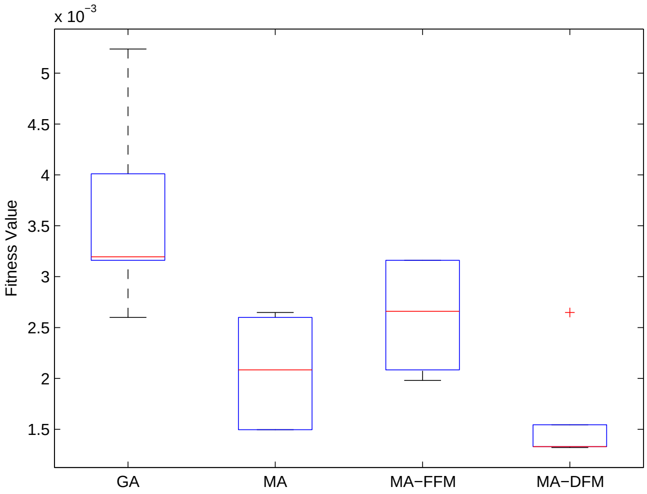


Fig. 6. Fitness boxplot of GA, MA, MA-FFM, and MA-DFM search at the end of 5000 function evaluations.

Table 3. *t-test* at 95% confidence level for GA, MA, MA-FFM, and MA-DFM search ($s+$, $s-$, or $\approx$ at row i and column j indicates that results of scheme at row i are significantly better, significantly worse, or indifferent compared to those at column j).

Scheme	**GA**	**MA**	**MA-FFM**	**MA-DFM**
GA	$\approx$	$s-$	$s-$	$s-$
MA	$s+$	$\approx$	$s+$	$s-$
MA-FFM	$s+$	$s-$	$\approx$	$s-$
MA-DFM	$s+$	$s+$	$s+$	$\approx$

5 Conclusions

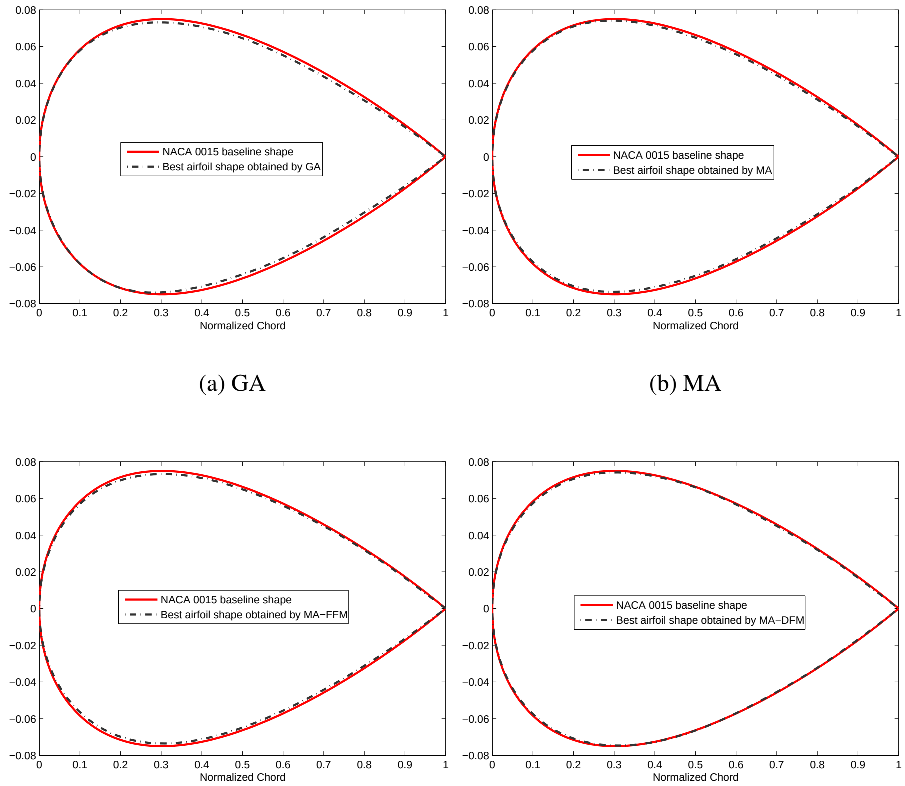
In this paper, we have proposed and studied a multi-fidelity evolutionary optimization technique, which we call the Memetic Algorithm with Dynamic Fidelity Model (MA-DFM). In contrast to existing techniques in the field which uses only non-adaptive or deterministic adaptive fidelity models, the MA-DFM dynamically acclimatizes to the local complexity for enhancing computational efficiency. In particular, the evolutionary framework studied is equipped with model fidelity control that decides, at runtime, the appropriate fidelity level of the computational model, which is deemed to be computationally less expensive, to be used in place of the exact analysis code as the search progresses. Empirical study on the aerodynamic airfoil design problem using standard evolutionary optimizers such as GA, MA, and MA-FFM demonstrates that MA-DFM is capable to arrive at improved quality solution and efficiency due to its ability in employing appropriate fidelity level of the computational model in the evolutionary search.

Acknowledgment

D. Lim and Y.S. Ong would like to thank Honda Research Institute Europe GmbH for sponsoring this work and members of Nanyang Technological University, Singapore for providing the computing resources.

References

1. M. G. Hutchinson, E. R. Unger, W. H. Mason, B. Grossman, and R. T. Haftka, “Variable-complexity aerodynamic optimization of a high speed civil transport wing”, *Journal of Aircraft*, 31:110-116, 1994.
2. W. H. Mason, D. L. Knill, A. A. Giunta, B. Grossman, L. T. Watson, and R. T. Haftka. “Getting the full benefits of CFD in conceptual design”, *16th AIAA Applied Aerodynamics Conference*, Albuquerque, AIAA-1998-2513, June 1998.
3. D. P. Raymer. Aircraft Design: A Conceptual Approach. Educational Series. AIAA, Reston, Virginia, 3rd edition, 1999.
4. T. Ray, H. Tsai, and C. Tan, “Effects of Solver Fidelity on a Parallel Search Algorithm’s Performance for Airfoil Shape Optimization Problems”, *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference*, 2002, Atlanta, Georgia.
5. A. J. Keane, “Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods”, *Journal of Aircraft*, 40(4):741-750, 2003.



(a) GA

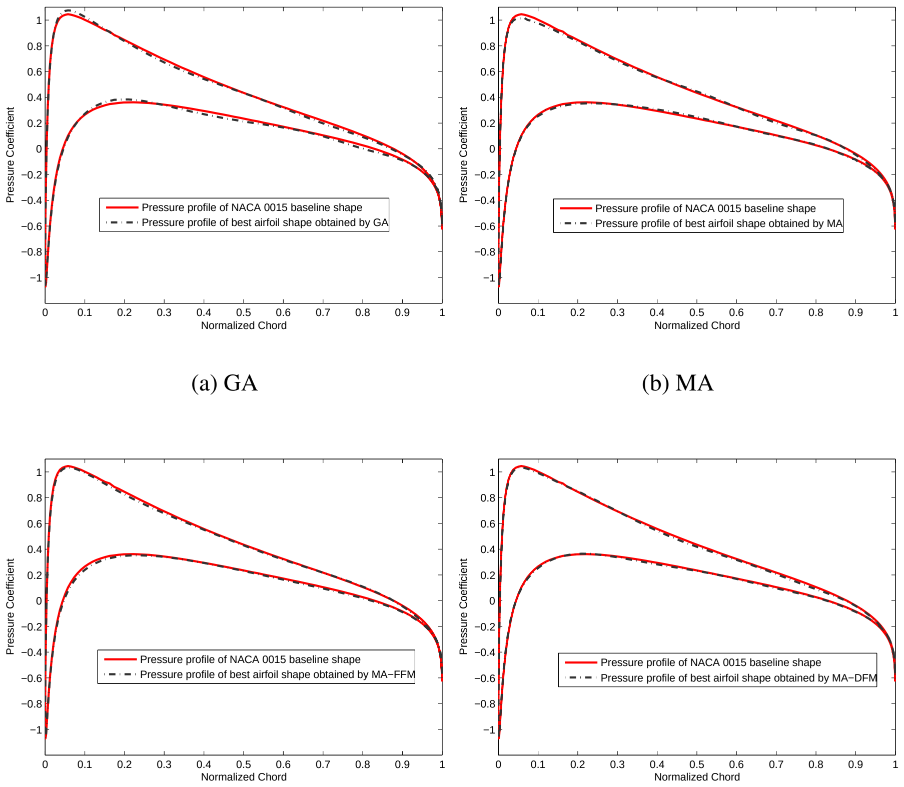
(b) MA

(c) MA-FFM

(d) MA-DFM

Fig. 7. Converged airfoil shapes obtained by GA, MA, MA-FFM, and MA-DFM.

6. H. Y. Wu, S. Yang, F. Liu, H. M. Tsai, "Comparison of Three Geometric Representations of Airfoils for Aerodynamic Optimization", *16th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida, 2003.
7. M. A. El-Beltagy and A. J. Keane, "A comparison of various optimization algorithms on a multilevel problem", *Engineering Applications of Artificial Intelligence*, 12(5):639-654, 1999
8. D. Edy, R. C. Averill, W. F. III Punch, and E. D. Goodman, "Evaluation of Injection Island GA Performance on Flywheel Design Optimization", In: I.C. Parmee (ed.), *Adaptive Computing in Design and manufacture*, Springer-Verlag, 1998.
9. M. F. Pantoja, P. Meincke, and A. R. Bretones, A hybrid genetic-algorithm space-mapping tool for the optimization of antennas, *IEEE Transactions on Antennas and Propagation*, 55(3) Part 1:777-781, March 2007.
10. I.C. Kampolis, A.S. Zymaris, V.G. Asouti, and K.C. Giannakoglou, "Multilevel optimization strategies based on metamodelassisted evolutionary algorithms, for computationally expensive problems", *IEEE Congress on Evolutionary Computation*, pp. 4116-4123, 2007.



(a) GA

(b) MA

(c) MA-FFM

(d) MA-DFM

Fig. 8. Pressure profiles at the upper and lower surfaces of the converged airfoil shapes obtained by GA, MA, MA-FFM, and MA-DFM.

11. L. Elliott, D. B. Ingham, A. G. Kyne, N. S. Mera, M. Pourkashanian, and C. W. Wilson, “An informed operator based genetic algorithm for tuning the reaction rate parameters of chemical kinetics mechanisms”, *GECCO*, LNCS 3103, pp. 945-956, 2004.
12. A. I. J. Forrester, N. W. Bressloff, A. J. Keane, “Response surface model evolution”, *16th AIAA Computational Fluid Dynamics Conference 23-26 June 2003*, Orlando, Florida.
13. R. Hinterding, Z. Michalewicz, and A. E. Eiben, “Adaptation in evolutionary computation: a survey”, *IEEE Conference on Evolutionary Computation*, 1997.
14. J. Lis, “Parallel genetic algorithm with dynamic control parameter”, *IEEE Conference on Evolutionary Computation*, pp. 324-329, 1996.
15. D. Schlierkamp-Voosen and H. Muhlenbein, “Adaptation of population sizes by competing subpopulations”, *IEEE Conference on Evolutionary Computation*, pp. 330-335, 1996.
16. Y. Jin, M. Huesken, and B. Sendhoff, “Quality measures for approximate models in evolutionary computation”, *Proc. GECCO Workshops: Workshop on Adaptation, Learning and Approximation in Evolutionary Computation*, pp. 170-174, Chicago, Illinois, USA, 2003.

17. L. Gräning, Y. Jin, and B. Sendhoff. “Individual-based management of meta-models for evolutionary optimization with applications to three-dimensional blade optimization”, In: S. Yang, Y.-S. Ong, Y. Jin(eds.), *Evolutionary Computation in Dynamic and Uncertain Environments*, pp.225-250, Springer, 2007.
18. D. Lim, Y. S. Ong, Y. Jin and B. Sendhoff, ’A Study on Metamodeling Techniques, Ensembles, and Multi-Surrogates in Evolutionary Computation’, *Proc. of Genetic and Evolutionary Computation Conference*. London, UK, pp. 1288 - 1295, 2007.
19. K. C. Giannakoglou, Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence, *International Review Journal Progress in Aerospace Sciences*, 38(5):43-76, 2002.
20. Y. S. Ong, P. B. Nair, and A. J. Keane, “Evolutionary optimization of computationally expensive problems via surrogate modeling”, *American Institute of Aeronautics and Astronautics Journal*, 41(4):687-696, 2003.
21. Z. Zhou, Y. S. Ong, M. H. Nguyen and D. Lim, “A Study on Polynomial Regression and Gaussian Process Global Surrogate Model in Hierarchical Surrogate-Assisted Evolutionary Algorithm, *Proc. of IEEE Congress on Evolutionary Computation*, Edinburgh, United Kingdom, Vol. 3, pp. 2832-2839, 2005.
22. C. T. Lawrence and A. L. Tits, “A computationally efficient feasible sequential quadratic programming algorithm”, *Society for Industrial and Applied Mathematics*, 11(4):1092-1118, 2001.
23. Y. S. Ong and A. J. Keane, “Meta-Lamarckian Learning in Memetic Algorithm, *IEEE Transactions On Evolutionary Computation*, 8(2):99-110, April 2004.
24. Y. S. Ong, M. H. Lim, N. Zhu and K. W. Wong, “Classification of Adaptive Memetic Algorithms: A Comparative Study”, *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 36(1):141-152, 2006.