Generative Transfer Optimization for Aerodynamic Design

Zhendong Guo  
Xi'an Jiaotong University  
ericzhendong@163.com  
Xi'an, Shaanxi, China

Wei Sun  
Aero Engine Academy of China  
swsunwei000000@163.com  
Beijing, China

Liming Song  
Xi'an Jiaotong University  
songlm@mail.xjtu.edu.cn  
Xi'an, Shaanxi, China

Jun Li  
Xi'an Jiaotong University  
junli@mail.xjtu.edu.cn  
Xi'an, Shaanxi, China

Zhenping Feng  
Xi'an Jiaotong University  
zpfeng@mail.xjtu.edu.cn  
Xi'an, Shaanxi, China

ABSTRACT

Transfer optimization, one type of optimization methods, which leverages knowledge of the completed tasks to accelerate the design progress of a new task, has been in widespread use in machine learning community. However, when applying transfer optimization to accelerate the progress of aerodynamic shape optimization (ASO), two challenges are encountered in sequence, that is, (1) how to build a shared design space among the related aerodynamic design tasks, and (2) how to exchange information between tasks most efficiently. To address the first challenge, a data-driven generative model is used to learn airfoil representations from the existing database, with the aim of synthesizing various airfoil shapes in a shared design space. To address the second challenge, both single- and multi-fidelity Gaussian processes (GPs) are employed to carry out optimization. On one hand, the multi-fidelity GP is used to leverage knowledge from the completed tasks. On the other hand, mutual learning is established between single- and multi-fidelity GP models by exchanging information between them in each optimization cycle. With the above, a generative transfer optimization (GTO) framework is proposed to shorten the design cycle of aerodynamic design. Through airfoil optimizations at different working conditions, the effectiveness of the proposed GTO framework is demonstrated.

INTRODUCTION

Aerodynamic shape optimization (ASO) has drawn wide attentions in the aerospace community, as it has been shown quite effective to reduce the design cycle of aircraft. Usually, ASO starts by parameterization of the design space. Non-uniformed rational B-Spline (NURBS) (Du et al., 2020), free-form deformation (FFD) (He et al., 2019) and etc. are representatives of conventional approaches for the parameterization task. The general process of these conventional parameterization approaches are as follows. First, the contour of a referenced aerodynamic shape is fitted by NURBS or FFD, etc. Then, the design space is defined by manually selecting the active control points on the fitted contour and their changing range as well. Since no explicit rule about which variable should be used to solve the problem most efficiently, the effectiveness of ASO with the above parameterization process can be easily influenced by a designer’s experience (Constantine et al., 2013). Moreover, such manually defined design space has to be specified for each ASO problem but may not be able to be reused for similar design tasks (Masters et al., 2016).

To address the above issues, people resort to learning the shape variability from the historically collected dataset with deep learning based generative models, which is a novel concept known as generative model based optimization (GMO) (Li et al., 2020) in recent few years. Specifically, with the advent of huge volume dataset, the deep learning based generative models can be used to learn a mapping from a latent space with dozens variables to the complex aerodynamic shapes. Therein, the design space of ASO for a wide range of applications is automatically encoded in this learnt latent space.

The advantage of GMO has been demonstrated in a handful of recent studies (Chen et al., 2019; Du et al., 2020). However, the existing studies of GMO are mainly focused on the optimization of a single task from scratch. In contrast, we humans routinely take full advantage of the past experiences when solving a new task. Motivated by this, a novel concept known as transfer optimization (TO) (Gupta et al., 2017), which leverages useful knowledge from the related tasks to
accelerate the optimization process, has been shown quite effective in the machine learning community. More importantly, unlike the conventional parameterization methods, the latent space of GMO provides an avenue to apply TO, by building a shared design space for similar ASO problems. Hence, instead of carrying out optimization from scratch, we propose generative transfer optimization (GTO) to shorten the design cycle of aerodynamic shape design.

In GTO, a shared design space is first trained by using the generative adversarial net (GAN) (Goodfellow et al., 2014) with an existing dataset. Then, we propose to carry out aerodynamic optimization with both single- and multi-fidelity Gaussian processes (GPs) (Rasmussen and Williams, 2005; Bonilla et al., 2008), where the multi-fidelity GP is used to leverage useful knowledge from the completed ASO problems. Further, as the single- and multi-fidelity GP may contain different information of the real optimal design, we propose to make the optimization process that uses the single- and multi-fidelity GP, respectively, serve as a helper task of the other, with knowledge transfer between them in each optimization cycle. Thereby, our proposed GTO can be expected to achieve the optimal aerodynamic design most efficiently.

The remainder of this paper is organized as follows: Section II presents preliminaries of several related topics. And then, Section III illustrates the details of GTO. After that, Section IV shows the experimental studies on the airfoil design at different working conditions. And Finally, we draw conclusions in Section V.

PRELIMINARIES
In order to illustrate the contributions of this work more clearly, we first provide preliminaries on several related topics, such as aerodynamic shape optimization, transfer optimization and Bayesian optimization in this section.

Aerodynamic Shape Optimization
The contour of an aerodynamic application is often represented by a sequence of scatter points when importing it into a CAD or a CFD software for further process or performance evaluation, as shown in Fig. 1 (a). In particular, the effective dimension of the scatter points are often set to be greater than 100 in order to accurately depict the shape contour. Then, Direct optimization of such ultra high-dimensional design space can be quite a challenging task (Masters et al., 2016). To address the above problem, a reference design is often first selected in conventional parameterization approaches, of which contour is fitted by NURBS, FFD and etc, as shown in Fig.1(b). Then, a pool of design candidates for subsequent aerodynamic optimization can be generated by adjusting the control points of the fitted contour, as shown in Fig.1 (c). Correspondingly, the dimension of optimization space is reduced to dozens or even fewer, which is more accessible by using genetic algorithms (Zhou et al., 2007) or surrogate model based optimization (Liu et al., 2019). While tremendous success has been achieved by using the above conventional parameterization methods, we may face the following issue. That is, the active control points and the changing range of related variables needs to be set by the designer (see Fig.1(b)). Further, in order to satisfy the increasing design requirements and include the desirable solution in the optimization searching space (Constantine et al., 2013), the dimension of the design space (that described by active control points) can be set very large, which may trigger the issue of “cold start” (Shahriari et al., 2015) when using Bayesian optimization algorithms as the optimizer. Luckily, the real-world problems are seldom in isolation. For instance, the airfoils working under some certain range of Mach number and Reynolds number may share similar optimal shape contour. Therefore, instead of carrying out aerodynamic optimization from scratch, we propose to leverage the useful knowledge from the related tasks to warm up the optimization progress, which is an emerging topic known as transfer optimization (Gupta et al., 2017) to be illustrated in the next subsection.

Transfer Optimization
Typically, transfer optimization can be grouped into three categories, namely sequential transfer optimization (STO), multitasking optimization (MTO) and multiform optimization (MFO) (Gupta et al., 2017), as shown in Fig.2. Specifically, STO focuses on optimizing for a single target problem, with the assistance of related source tasks, and the knowledge transfer is often conducted in an unidirectional way, i.e., from the completed tasks to the new task of interest. Differently, MTO is carried out by optimizing several tasks simultaneously, where the knowledge transfer is carried out bidirectionally with equal treatment for each task. Further, MFO inherits the spirit of STO and MTO. However differently, the so-called “tasks” in MFO are alternate formulations of a single task of interest, Fig.2(c) shows the scheme of multitasking MFO.

More specifically, the differences between STO, MTO and MFO can be seen from the following applications. As reported in (Valenzuela del Rio and Mavris, 2015), when optimizing the engine shaft horsepower of a UH-60A modification with a Fenestron tail, they make use of the observations from the UH-60A with a conventional tail that previously designed. This is the typical case of STO. Further, in the stage of preliminary design, as it can be difficult to decide to whether equip UH-60A with a Fenestron tail or a conventional tail, the optimizations for UH-60A with Fenestron tail and conventional tail can be carried out concurrently, by using the MTO strategy shown in Fig.2(b). Alternatively, when optimizing UH-60A with a Fenestron tail, if both the Euler equation solver and 3D RANS solver are available for the performance evaluation, the optimization can be carried out with both Euler equation solver and RANS solver simultaneously, by using the MFO strategy shown in Fig.2(c). Note that the Euler equation solver and RANS solver can be regarded as two alternate formulations of the objective function. Since the low-fidelity Euler equation solver is much cheaper, the MFO by combining Euler equation
Variable range of control points are set by the experience (a) (b) (c)

Represented by sequence points with 100+ effective dimension

Figure 1 Issue of conventional parameterization for aerodynamic design optimization, (a) airfoil represented by 100+ sequence points, (b) issues of conventional parameterization method, (c) design pool of conventional parameterization method

and RANS solver can help to accelerate the optimization progress.

Furthermore, though in different forms, all of the three categories of TO algorithms shown in Fig. 2 aim to improve the optimization via knowledge transfer with the the knowledge base $M(t)$. More specifically, in order to exchange information effectively, the so-called “knowledge” should be embedded in a shared design space among the the related tasks. However, the optimization searching space is specified for each aerodynamic design when using conventional parameterization method, as shown in Fig. 1. In other words, the design space defined by conventional parameterization method may not be able to be shared among similar tasks, and thus making the transfer optimization for aerodynamic design questionable. To address the above issue, we propose to use generative models (Goodfellow et al., 2014) to build a shared design space for similar tasks, which will be discussed in detail in the next section.

Bayesian Optimization

Bayesian optimization (BO) (Shahriari et al., 2015) is well-known to be sample-efficient, which is most frequently used for solving expensive black-box problems such as aircraft design (Li et al., 2020). The general procedure of BO is as follows. First, it uses Gaussian process (GP) to build a surrogate of the objective function. Second, it employs an acquisition function which incorporates the GP surrogate to select the next promising sample to query, and simulations are conducted to evaluate the new sample candidate. Third, the new sample is added to the training set for the next iteration.

As a key component of BO, GP (Rasmussen and Williams, 2005) is a stochastic process to approximate the input-output relation, which formulates the function prediction $Y(x)$ as a Gaussian random variable:

$$ Y(x) \sim \text{GP}(m(x), k(x, x')) $$

where $m(x)$ denotes the mean function, and $k(x, x')$ is the covariance function between samples $x$ and $x'$. Usually, we take $m(x) \equiv 0$ and use the popular squared exponential (SE) covariance function to describe the covariance correlations. The marginal likelihood function is used to tune the GP hyperparameters. For standard GP with samples from a single source, the posterior estimates of the mean function prediction ($\hat{y}$) and the corresponding mean squared error ($\hat{\sigma}^2$) at a point $x$ are expressed as:

$$ \hat{y}(x) = k(x, X)(K + \sigma_n^2 I)^{-1} y $$

$$ \hat{\sigma}^2(x) = k(x, x) - k(x, X)(K + \sigma_n^2 I)^{-1}k(X, x) + \sigma_n^2 $$

(2)
where, $\sigma^2_z$ is the Gaussian noise, $K$ is the covariance matrix of the input variables, and $k$ is the cross-correlation vector between $x$ and $X$.

As another key component of BO, the acquisition function acts as a surrogate that determines which sample to be evaluated in BO. Among various acquisition functions, expected improvement (EI) (Jones et al., 1998) is most frequently used. Assuming that the function prediction $Y(x)$ comes from a GP with the posterior estimate as $Y(x) \sim N(\hat{y}(x), \sigma^2(y(x)))$, the basic idea of EI is to measure how much objective improvement can be attained with respect to the current best solution $f_{\text{best}}$, which is formulated as:

$$EI(x) = (f_{\text{best}} - \hat{y}(x)) \Phi(u) + \sigma(y(x)) \phi(u)$$

$$u = \frac{f_{\text{best}} - \hat{y}(x)}{\sigma(y(x))}$$

(3)

where, $\Phi(\cdot)$ and $\phi(\cdot)$ denote the standard normal distribution function and density function, respectively. In each optimization cycle, we select the next point to sample by maximizing the EI:

$$x^* = \arg \max_x EI(x)$$

(4)

**METHODOLOGY**

When applying transfer optimization to accelerate the progress of aerodynamic design, two problems needs to be solved, i.e., (1) how to build a shared design space among the related aerodynamic design tasks, and (2) how to efficiently exchange information between the related tasks. We propose a novel framework, namely generative transfer optimization (GTO), to address the above issues, which is illustrated in detail in this section.

**Generative Model-Based Shape Parameterization**

Instead of parameterizing the design space in a two-step way as the conventional parameterization methods do (see Fig.1), we propose to learn the synthesis of airfoil contours from the historically collected dataset in a one-step way, and we achieve such a goal by using the most popular generative model, i.e., generative adversarial net (GAN) (Goodfellow et al., 2014), as shown in Fig.3. Specifically, GAN is consisted of two components as the generator and the discriminator. With the input of latent variable $z$, the generator is trained to synthesize airfoil contour that follows the characteristics of the real dataset $X$. In the meantime, the discriminator is trained to distinguish the synthesized airfoil $x'$ of Generator (also denoted as $G(z)$ in Eq.(5)) from the real airfoil $x$ that sampled from the dataset $X$, with the following adversarial training loss:

$$\min_{G} \max_{D} V(D,G) = E_{x \sim p_{\text{data}}} [\log D(x)] + E_{z \sim p_z} [\log (1 - D(G(z)))]$$

(5)

where, $D(x)$ and $D(G(z))$ are the probabilities that the samples come from the real dataset $X$ or the generator, respectively. After training, the airfoil design space over a wide range of applications is implicitly encoded in the generator of GAN. In other words, various airfoil shapes can be synthesized by varying the latent vector $z$.

Figure 4 compares the airfoils generated by GAN and the conventional parameterization method, NURBS. Specifically, Fig.4(a) shows the airfoils randomly drawn from UIUC dataset1, which collects more than 1500 airfoils that designed for a wide range of applications. Fig.4(b) exhibits the airfoils randomly synthesized by the generator of GAN, where the UIUC dataset is used as the training set. For comparison purpose, Figs.4(c) and (d) show the airfoils generated by the NURBS, where the NACA2424 and NACA0012 are used as the reference design, respectively.

Obviously, the shape variability of airfoils in Fig.4(c) and (d) are very limited. Moreover, there is no shape overlap for the airfoils exhibited in Figs.4(c) and 4(b). It means, the optimal airfoil designs that attained from the design space shown in Fig.4(c) cannot be transferred to the design space shown in Fig.4(d), and vice versa. In other words, the knowledge transfer between these two design spaces is almost impossible.

In contrast, the airfoils shown in Fig.4(b) exhibit sufficient shape variability, which covers most of the airfoils in the UIUC database. It implies that, when optimizing over the latent space of GAN, the optimal solution of a completed task can be directly used as a starting point for the optimization of a similar task to accelerate the design process. In other words, the knowledge transfer between similar tasks can be facilitated by the latent space of GAN, and therefore we propose GAN-based generative transfer optimization to shorten the optimization cycle of aerodynamic designs.

**Sequential Transfer Optimization with Multi-Fidelity GP**

While the latent space of GAN provides an avenue to enable knowledge transfer between the related tasks, how to build the knowledge base $M(t)$ (see Fig.2) and thus leverage the knowledge adaptively can be another key issue. Therefore, a multi-fidelity GP (MFGP) is devised in this work to leverage the knowledge that drawn from the completed task.

Specifically, let $Z_T$ and $Z_S$ denote the samples that coming from the target problem and completed tasks, respectively; and $y(Z_T)$ and $y(Z_S)$ the objective function values of the target and completed tasks. In the meantime, the functional

1https://m-selig.ae.illinois.edu/ads.html
Figure 3 Airfoil shape parameterization by using generative adversarial net

Figure 4 Comparison of airfoil design pools, (a) airfoils randomly drawn from the UIUC dataset, (b) airfoils randomly generated by the generator of GAN, (c) airfoils sampled by using NURBS, where the NACA2424 is used as the reference design, (d) airfoils sampled by using NURBS, where the NACA0012 is taken as the reference design.

correlations between the target and completed task is measured by the coefficients $\rho_{ij}$. Then, the correlation vector $k_M$ and correlation matrix $K_M$ of MFGP can be formulated as below (Bonilla et al., 2008):

$$k_M(z, Z^*) = [\rho_{11}k(z, Z_T), \rho_{12}k(z, Z_S)]$$

$$K_M(Z^*) = \begin{bmatrix} \rho_{11}K(Z_T) & \rho_{12}K(Z_T, Z_S) \\ \rho_{21}K(Z_S, Z_T) & \rho_{22}K(Z_S) \end{bmatrix} + D$$

where, $k$ and $K$ are the correlation vector and covariance matrix formulation of the input variables in Eq.(2); and $D$ is a $2 \times 2$ diagonal matrix with diagonal elements $\{\sigma^2_{n,j}\}_{1 \leq j \leq 2}$ where $\sigma^2_{n,j}$ is the noise term associated with the source and target samples, respectively. Thereby, the prediction at an unobserved point $z$ over the latent space of GAN can be expressed as:

$$\hat{y}_{MFGP}(z) = k_M(z, Z^*) K_M^{-1}(Z^*) \begin{bmatrix} y(Z_T) \\ y(Z_S) \end{bmatrix}$$

$$\sigma^2_{MFGP}(z) = k_M(z) - k_M(z, Z^*) K_M^{-1} k_M(Z^*, z) + \sigma^2_{n,2}$$

Correspondingly, the EI acquisition function for the Bayesian optimization with MFGP is shown as below:

$$EI_{MFGP}(z) = (f_{\text{min}} - \hat{y}_{MFGP})\Phi(\frac{u_{MFGP}}{\sigma_{MFGP}}) + \sigma_{MFGP}(z)\phi(\frac{u_{MFGP}}{\sigma_{MFGP}})$$

$$u_{MFGP} = (f_{\text{min}} - \hat{y}_{MFGP}(z))/\sigma_{MFGP}(z)$$

Note that, $\rho_{ij}$ in Eq.(6) will be tuned in each optimization cycle to optimally capture the functional correlations between the completed and target samples. And hence, the knowledge gained from the completed tasks can be most effectively used in Eq.(8) to facilitate the optimization search of target problem.

**Multitasking Multiform Optimization with Alternate Formulations**

While the MFGP-based transfer optimization can be quite effective to warm up the progress particularly that at the early stage of the optimization process, the accuracy of single-fidelity GP (SFGP) built with target samples alone can gradually become better than that of MFGP, with the increase of target samples in the iteration process (Guo et al., 2018).
Therefore, instead of carrying out optimization with a single formulation, we propose to conduct optimization with both MFGP and SFGP (see Eq. (9)) simultaneously. Moreover, the optimization with MFGP and SFGP are made to be working under multitasking environment (see Fig.2(c)), i.e., they exchange the promising solutions in each optimization cycle, as shown in Eq.(10). This way, the optimization with either MFGP or SFGP is severed as a helper task of the other, which allows us to leverage the unique advantage of each of them to achieve the optimal solution most efficiently. Furthermore, the sample search with Eq.(11) can be carried out in a parallel way, which can further accelerate the design process when the wall clock time (i.e., the number of total iterations rather than the number of total function calls) is concerned.

\[
EI_{\text{SOGP}}(z) = (f_{\min}^{(t)} - \hat{y}_{\text{SFGP}}(z))\phi(u_{\text{SFGP}}) + \sigma_{\text{SFGP}}(z)\phi(u_{\text{SFGP}})
\]

\[
u_{\text{SFGP}} = (f_{\min}^{(t)} - \hat{y}_{\text{SFGP}}(z))/\sigma_{\text{SFGP}}(z)
\]

\[
f_{\min}^{(t)} = \min\left\{f_{\min}^{(t)}(z_{\text{best}}^{\text{SFGP}}), f_{\min}^{(t)}(z_{\text{best}}^{\text{MFGP}})\right\}
\]

\[
Z^* = \arg\max_z EI_{\text{MFGP}}(z), \arg\max_z EI_{\text{SFGP}}(z)
\]

With the above, Fig.5 shows the flowchart of proposed GTO framework, which is consisted of two stages. In the first stage, a GAN model with the available dataset (e.g., UIUC database) is used to learn a shared design space. Note that the learnt design space can be reused for a wide a range of aerodynamic applications. Then, in the second, the STO that based on MFGP (see Eq.(8)) and the multitasking MFO with alternate formulations (see Eqs.(10) and (11)) are integrated in a parallel way for the optimization of a target problem.

**EXPERIMENTAL STUDIES**

To showcase the effectiveness of proposed GTO framework, we test it on two ASO problems in this section, where the airfoils that work in conditions with different Mach number and Reynolds number are selected as the source tasks.

**Baselines for Comparison and Experimental Setup**

To show the benefits of transfer optimization, the conventional approach without knowledge transfer is selected as a baseline, denoted by EGO. In the meantime, the sequential transfer optimization by using the multi-fidelity GP alone to
Table 1 Neural network layers setup of generator and discriminator of GAN for the airfoil parameterization

<table>
<thead>
<tr>
<th>Layers</th>
<th>Generator</th>
<th>Discriminator</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>Dense layer, ReLU, batch normalization</td>
<td>Convolutional layer, ReLU, batch normalization, dropout</td>
</tr>
<tr>
<td>L1</td>
<td>Dense layer, ReLU, batch normalization</td>
<td>Convolutional layer, ReLU, batch normalization, dropout</td>
</tr>
<tr>
<td>L2</td>
<td>Deconvolutional layer, ReLU, batch normalization</td>
<td>Convolutional layer, ReLU, batch normalization, dropout</td>
</tr>
<tr>
<td>L3</td>
<td>Deconvolutional layer, ReLU, batch normalization</td>
<td>Dense layer, ReLU, batch normalization</td>
</tr>
<tr>
<td>L4</td>
<td>Deconvolutional layer, ReLU, batch normalization</td>
<td>Dense layer, ReLU, batch normalization</td>
</tr>
<tr>
<td>L5</td>
<td>Deconvolutional layer, ReLU, batch normalization</td>
<td>Dense layer, ReLU, batch normalization</td>
</tr>
</tbody>
</table>

The flowcharts of EGO and STO are shown in Fig. 6. For fair comparison, a uniform test bed is built by using the GPML toolbox. To train a shared design space, the neural network architecture of GAN is shown in Table 1. The input of the generator is set as random variables which follow the uniform distribution, i.e., $Unif(0, 1)$. The UIUC dataset which contains more than 1500 airfoils of various applications is used as the training set. The Tensorflow 2.0 is used to train the GAN model. To train the GAN model, a mixed programming strategy is used to integrate the GPML-based optimization algorithms and GAN. When starting the optimization search, the initial training samples of of the target objective are generated by the Latin hypercube sampling (Stein, 1987). Meanwhile, the same number of low-fidelity samples are drawn from a completed source task to build multi-fidelity GP. In particular, these low-fidelity samples are selected as those with best objective function values from the source dataset. The optimizations with EGO, STO and GTO are stopped when the total number of function calls reaching to the sample budget.

Design Optimization of a Low-speed Airfoil

By following the settings in (Chen et al., 2019), the design objective of the low-speed airfoil is set to be the lift to drag ratio, $L/D$, with $Re = 1.8 \times 10^6$, Mach number $Ma_{\infty} = 0.01$, and attach angle $AoA = 0$ deg. The related mathematical
Expression is formulated as:

$$\min f(z) = \min \frac{L}{D}(z)$$

with $$Ma_\infty = 0.01$$, $$Re = 1.8 \times 10^6$$, $$AoA = 0 \text{ deg}$$  \( (12) \)

In the meantime, the optimal solutions of a subsonic airfoil working with $$Ma_\infty = 0.45$$ is used as the source task for the transfer optimization in this case. The Reynolds number $$Re$$ and the attack angle $$AoA$$ of the completed source task are assumed to be the same as the target problem. Additionally, XFOIL \((Drela, 1989)\) is used to compute $$\frac{L}{D}$$.

Figure 7 shows the optimization results of the low-speed airfoil, where the shaded areas in Figs. 7(a) and 7(b) show the variations of the optimization results over 10 runs. The boxplot in Figs. 7(c) exhibits the medians and the distributions of the final optimal solutions. Compared to the conventional approach without knowledge transfer (denoted by EGO), the sequential transfer optimization (denoted by STO) shows better convergence rate at the early stage. Nevertheless, EGO attains even better solutions at the end of the optimization cycle.

The reason behind can be explained as follows. By leveraging knowledge from the source samples, the multi-fidelity GP can have better accuracy than the single-fidelity GP that built with a small number of target samples alone. In other words, the source samples helps to warm up the optimization search, and hence STO progresses even faster at the early stage. However, the neighborhood of the real optimal solution of the low-speed airfoil and that of the subsonic airfoil cannot be overlapped. Hence, as the iteration goes on, the accuracy of single-fidelity GP built with much more target samples can become better than that of multi-fidelity GP \((Guo et al., 2018)\). And then, “negative transfer” \((Wang et al., 2020)\) happens, i.e., the information gained from the source task misleads the algorithm to query samples in areas other than the vicinity of the real optimal solution. Hence, STO achieves worse optimal solutions than that of EGO at the end.

Differently, both the single- and multi-fidelity GP are used in our proposed GTO. Instead of only taking the current best solution attained with multi-fidelity GP \(i.e., \ f^{(t)}_{\text{best}} (\hat{x}_{\text{MFGP}}) \) into account, the EI acquisition function in our proposed GTO selects the next “promising” sample candidate to query by considering $$f^{(t)}_{\text{best}} (\hat{x}_{\text{MFGP}})$$ and $$f^{(t)}_{\text{best}} (\hat{x}_{\text{SFGP}})$$ simultaneously, as shown in Eq. \((10)\). This way, at later stage of the optimization process, the misleading of multi-fidelity GP can be corrected by the single-fidelity GP of better accuracy, and thereby the “negative transfer” issue can be addressed. Therefore, our proposed GTO achieves the best solutions with even faster convergences rate, as shown in Fig. 7(a). Moreover, the single- and multi-fidelity GP based loops can be conducted in parallel in GTO. Hence, when the wall clock time \(i.e., \) the number of optimization cycles other than the total number of sample cost) is concerned, the advantage of our proposed GTO is more significant, as shown in Fig. 7(b).

**Design Optimization of a Transonic Airfoil**

Similar to the design of low-speed airfoil, $$\frac{L}{D}$$ is also used as the objective function for the shape optimization of a transonic airfoil. The working conditions are set as follows, \(i.e., \) $$Re = 6.5 \times 10^6$$, Mach number $$Ma_\infty = 0.734$$, and attach
Figure 8 Optimization of a Transonic airfoil with source data coming from subsonic airfoils, with different Reynolds number

angle \( \text{AoA} = 0 \text{ deg} \). Correspondingly, the mathematical formulation for the transonic airfoil optimization is expressed as:

\[
\min f(z) = \min \frac{L}{D}(z) \\
\text{with } Ma_\infty = 0.734, \ Re = 6.5 \times 10^6, \ AoA = 0 \text{ deg}.
\] (13)

Meanwhile, a source task with different Mach number and Reynolds number is used for the transfer optimization in this case, with \( Ma_\infty = 0.45, \ Re = 3.5 \times 10^6, \ AoA = 0 \text{ deg} \). XFOIL is used to compute \( L/D \).

Figure 8 shows the optimization results of the transonic airfoil. Note that the flow conditions of the transonic and subsonic airfoils are more close to each other, when comparing to that between the low-speed and subsonic airfoils. And accordingly, the data samples of the subsonic and transonic airfoils can be better correlated. Hence, instead of “negative transfer” as encountered for the low-speed airfoil optimization, the sample knowledge from the source task can be fully leveraged to accelerate the optimization process of the transonic airfoil. Therefore, STO achieves better results with even faster convergence rate than the conventional approach without knowledge transfer (denoted by EGO). And further, by combining the strategies of STO with multitasking MFO in the searching propose, our proposed GTO achieves the best results within budget. With the above, the effectiveness of our proposed GTO has been demonstrated.

CONCLUSIONS

In this paper, we propose a generative transfer optimization (GTO) framework, which leverages the unique advantage offered by generative model based optimization (GMO) and transfer optimization (TO), respectively, to boost the performance of aerodynamic shape optimization (ASO). In particular, notice that the latent space of GMO can learn a shared design space for a wide range of similar design tasks, we propose to leverage useful knowledge from the completed ASO problems, by building a multi-fidelity Gaussian process (GP) over this latent space. Further, considering the fact that “negative transfer” may happen at later optimization stage when using multi-fidelity GP, a single-fidelity GP is also used for the sample search in GTO. More importantly, the optimization process that uses the single- and multi-fidelity GP, respectively, are made to serve as a helper task of the other, which exchanges optimal solutions in each optimization cycle. And hence, our proposed GTO can make full use the knowledge gained from the related tasks to attain the optimal aerodynamic designs most efficiently. Through tests on airfoil designs with different working conditions, the effectiveness of our proposed GTO has been well demonstrated.

ACKNOWLEDGMENTS

The authors would like to express appreciation for the financial support from the National Natural Science Foundation of China (Grant No. 51676149), National Science and Technology Major Project (2019-II-0008-0028).

References


