Journal of Computational Physics 462 (2022) 111263

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Multi-condition multi-objective optimization using deep reinforcement learning

Sejin Kim [1], Innyoung Kim [1], Donghyun You [*]

Department of Mechanical Engineering, Pohang University of Science and Technology, 77 Cheongam-Ro, Nam-Gu, Pohang, Gyeongbuk 37673, South Korea

ARTICLE INFO

Article history:
Received 19 January 2022
Received in revised form 26 April 2022
Accepted 27 April 2022
Available online 4 May 2022

Keywords:
Multi-condition multi-objective optimization
Deep reinforcement learning
Shape optimization
Pareto front
Aerodynamic shape design

ABSTRACT

A novel multi-condition multi-objective optimization method that can find Pareto front over a defined condition space is developed using deep reinforcement learning. Unlike the conventional methods which perform optimization at a single condition, the present method learns correlations between conditions and optimal solutions. The exclusive capability of the developed method is examined in solutions of a modified Kursawe benchmark problem and an airfoil shape optimization problem. The solutions include nonlinear characteristics which are difficult to be resolved using conventional optimization methods. Pareto front with high resolution over a condition space is successfully determined in both problems. Compared with multiple operations of a single-condition optimization method for multiple conditions, the present multi-condition optimization method shows a greatly accelerated search of Pareto front by reducing the required number of function evaluations. An analysis of aerodynamic performance of optimally designed airfoils confirms that multi-condition optimization is indispensable to avoid significant degradation of target performance for varying flow conditions.

1. Introduction

Optimization is a long-standing unanimous research topic in all engineering fields [1–4]. For example, countless studies for shape optimization have been conducted with the objective of performance improvement of fluid machines [5–11]. Fluid machines often have a wide range of operating conditions and multiple objectives [12–18]. For instance, wind turbines should be designed in consideration of various wind conditions and multiple objectives such as enhancement of power generation and mitigation of blade loads and acoustic noise. [19–21]. Aircraft encounter various flight conditions such as takeoff, cruise, and landing, and the aerodynamic requirements vary depending on the conditions. [22]. In such cases, there are considerable demands for an optimization method that can handle multiple objectives over a range of operating conditions.

Optimization for multiple objectives is possible through conventional multi-objective (MO) optimization methods which find a set of optimal solutions, i.e., Pareto front [23–26]. However, to the best of our knowledge, there is no optimization method that can find optimal solutions over a defined condition space. To find solutions within a condition space, multiple operations of optimization should be performed for several discrete conditions in the space. Secanell *et al.* [22] performed

[*] Corresponding author.
E-mail address: dhyou@postech.ac.kr (D. You).
[1] Authors contributed equally.

https://doi.org/10.1016/j.jcp.2022.111263
0021-9991/© 2022 Elsevier Inc. All rights reserved.

multiple operations of optimization at 7 discrete flight conditions to design a morphing airfoil. Lyu and Martins [27] performed optimization separately at 407 different conditions to design a morphing wing, which required 407 × 6 hours on 64 processors.

However, if optimization is performed at some chosen conditions, there can be significant degradation of target performance at other conditions. It becomes severe in problems with nonlinear characteristics such as shape optimization in fluid mechanics. In addition, performing optimization at sufficiently many conditions is inefficient in terms of computational cost and time. To overcome the limitations, a multi-condition (MC) optimization method that can efficiently find optimal solutions over a defined condition space is highly necessary.

Recently, studies combining artificial intelligence with optimization are being actively conducted [28,29]. Deep reinforcement learning (DRL) is emerging as a new engine for shape optimization [30]. Viquerat *et al.* [31] showed the capability of DRL by successfully performing airfoil shape optimization at a single flow condition. Qin *et al.* [32] conducted MO optimization for the shape of a cascade blade at a single flow condition using DRL. Li *et al.* [33] conducted airfoil shape optimization and showed that, even at untrained flow conditions, DRL can extract improved shapes. The previous studies demonstrate that DRL has good capability for MO optimization, and also has potential for being employed for multi-condition multi-objective (MCMO) optimization.

The suitability of DRL for MCMO optimization can be found in the nature of DRL. The basic concept of DRL is to find an optimal action for a given state. Thus, by setting the state considering a condition space and multiple objectives, the corresponding optimal solution can be obtained as the action. A DRL-based MCMO optimization method can be efficient in handling various conditions as it learns correlations between conditions and optimal solutions. This is in contrast to the conventional methods in which optimization is performed at each condition independently.

In the present study, a DRL-based MCMO optimization method that can efficiently find Pareto front over a defined condition space is developed for the first time in the literature. Then, two MCMO optimization problems are solved by the developed method. The first problem is a benchmark problem to validate the method. As the benchmark problem, the Kursawe test problem [34], which is a representative MO optimization problem, is newly extended to an MCMO optimization problem. The second problem is airfoil shape optimization. It is a representative shape optimization problem involving fluid mechanics, which has been actively studied due to its broad applicability to many engineering fields [31,35,36]. Finally, further analysis is conducted in each problem to elucidate the exclusive capability of the proposed method.

2. Background

2.1. Multi-objective optimization

2.1.1. Problem description

An MO optimization problem with the number of k objective functions can be defined as follows:

$$\begin{aligned}&\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = \min_{\boldsymbol{x}}(f_1(\boldsymbol{x}), \dots, f_k(\boldsymbol{x}))\\ &\qquad\qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega},\end{aligned} \tag{1}$$

where $\boldsymbol{\Omega}$ is the decision space which is defined as the set $\{\boldsymbol{x}|\ g_m(\boldsymbol{x}) \leq 0, m = 1, 2, \dots, n\}$. $\boldsymbol{x}$ is a decision vector, $g_m(\boldsymbol{x})$ is a constraint function that determines the feasible region of $\boldsymbol{x}$, and n is the number of the constraints of $\boldsymbol{x}$. $\boldsymbol{f}$ consists of k real-valued objective functions.

For this problem, in general, no single solution can optimize objectives simultaneously as the objectives can conflict with each other. Instead, a set of optimal trade-off solutions among different objectives exists by the concepts of Pareto dominance and Pareto optimality which are defined as follows:

- Pareto dominance: $\boldsymbol{x}$ is said to Pareto dominate $\boldsymbol{y}$, denoted by $\boldsymbol{x} \preceq \boldsymbol{y}$, if and only if $\forall i \in \{1, 2, \dots, k\}$, $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$ for at least one index $j \in \{1, 2, \dots, k\}$.
- Pareto optimality: a solution $\boldsymbol{x}^* \in \boldsymbol{\Omega}$ is said to be Pareto optimal if and only if $\nexists \boldsymbol{x} \in \boldsymbol{\Omega}$ such that $\boldsymbol{x} \preceq \boldsymbol{x}^*$.

The goal of an MO optimization problem is to find a Pareto optimal set and the corresponding Pareto front which is defined as $\{\boldsymbol{f}(\boldsymbol{x})|\ \boldsymbol{x} \in \text{Pareto optimal set}\}$.

2.1.2. Weighted Chebyshev method

The weighted Chebyshev method is one of decomposition-based methods for solving an MO optimization problem. It scalarizes an MO optimization problem into multiple single-objective (SO) optimization problems by introducing a weight vector $\boldsymbol{w}$ and the Chebyshev scalarizing function $f_{Chebyshev}$. The weight vector $\boldsymbol{w}$ determines weights among objectives and $f_{Chebyshev}$ is the scalarized objective of each SO optimization problem. Then, the original MO optimization problem can be solved by performing a number of scalarized SO optimization processes with different $\boldsymbol{w}$. The scalarized SO optimization problem can be written as follows:

$$\min_{\boldsymbol{x}} f_{Chebyshev} = \min_{\boldsymbol{x}}(\max_{i=1,\dots,k} w_i|f_i(\boldsymbol{x}) - f_i^*|), \tag{2}$$

where $|\boldsymbol{w}| = 1$ and $w_i \geq 0$. f_i^* is an utopia value which is defined as $f_i^* = \inf\{f_i(\boldsymbol{x})\} - \tau_i$, where τ_i is a small positive value. The detailed description of the weighted Chebyshev method in the context of Pareto optimality is presented in Appendix A.

Unlike other decomposition-based methods, the weighted Chebyshev method guarantees that all Pareto optimal solutions can be obtained for both convex and nonconvex problems [37]. Because of the advantage, it has been widely used in the literature [38,39] and was also successfully coupled with DRL [40]. One of the difficulties in adopting the weighted Chebyshev method for MO optimization is that the utopia point $\boldsymbol{f}^*$ should be known before optimization, which requires SO optimization for each objective in advance. In the present study, the difficulty is overcome by adaptively updating the utopia point in a single optimization process, which will be further discussed in Section 4.1.1.

2.2. *Deep-reinforcement-learning-based optimization*

2.2.1. *Deep reinforcement learning*

DRL is a process of learning a policy to determine an optimal action from a given state [41]. At each discrete step n, it determines an action according to its current policy $\boldsymbol{a}_n = \pi(\boldsymbol{s}_n)$. Then, through the execution of the action, a reward r_n and the next state $\boldsymbol{s}_{n+1}$ are determined. As the step progresses, data is accumulated and learning proceeds. The goal of learning is to find an optimal policy that maximizes a value function $Q^\pi(\boldsymbol{s}_n, \boldsymbol{a}_n)$ [42] which is defined as the expected sum of an immediate reward r_n and discounted future rewards as follows:

$$Q^\pi(\boldsymbol{s}_n, \boldsymbol{a}_n) = \mathbb{E}(r_n + \gamma r_{n+1} + \gamma^2 r_{n+2} + \ldots), \tag{3}$$

where γ is a discount factor that determines the weight between short-term and long-term future rewards. This process is repeated until the terminal state. When the terminal state is reached, one episode ends.

2.2.2. *Single-step deep-reinforcement-learning-based optimization*

Single-step DRL-based optimization is recently introduced by Viquerat *et al.* [31] where one learning episode consists of a single step; if an action is determined from a given state, a reward is given accordingly and the episode ends without the next state. Since the future rewards in Eq. (3) do not exist, the discount factor γ does not have to be defined and learning proceeds to maximize only the immediate reward. As a result, the optimal action that maximizes the reward itself can be directly determined. Therefore, if the reward is set as the objective function to be optimized, the optimal solution that maximizes the objective function can be directly obtained. By virtue of the characteristics, single-step DRL is preferred for solving optimization problems [43].

3. Multi-condition multi-objective optimization problem

An MCMO optimization problem is extended from an MO optimization problem to include not only the decision vector $\boldsymbol{x}$ but also the condition vector $\boldsymbol{c}$. The problem with the number of k objective functions is defined as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{c}) = \min_{\boldsymbol{x}} (f_1(\boldsymbol{x}, \boldsymbol{c}), \ldots, f_k(\boldsymbol{x}, \boldsymbol{c})) \\ &\qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega}, \boldsymbol{c} \in \boldsymbol{\Phi}, \end{aligned} \tag{4}$$

where $\boldsymbol{\Omega}$ is the decision space which is defined as the set $\{\boldsymbol{x}|\ g_m(\boldsymbol{x}) \leq 0,\ m = 1, 2, \ldots, n\}$. $\boldsymbol{x}$ is a decision vector, $g_m(\boldsymbol{x})$ is a constraint function that determines the feasible region of $\boldsymbol{x}$, and n is the number of the constraints of $\boldsymbol{x}$. $\boldsymbol{\Phi}$ is the condition space which is defined as the set $\{\boldsymbol{c}|\ h_p(\boldsymbol{c}) \leq 0,\ p = 1, 2, \ldots, q\}$. $\boldsymbol{c}$ is a condition vector, $h_p(\boldsymbol{c})$ is a constraint function that determines the feasible region of $\boldsymbol{c}$, and q is the number of the constraints of $\boldsymbol{c}$. $\boldsymbol{f}$ consists of k real-valued objective functions.

The concepts of Pareto dominance and Pareto optimality are extended to take the condition vector $\boldsymbol{c}$ into account as follows:

- Pareto dominance: $\boldsymbol{x}$ is said to Pareto dominate $\boldsymbol{y}$ for the condition vector $\boldsymbol{c} \in \boldsymbol{\Phi}$, denoted by $\boldsymbol{x} \preceq_{\boldsymbol{c}} \boldsymbol{y}$, if and only if $\forall i \in \{1, 2, \ldots, k\}$, $f_i(\boldsymbol{x}, \boldsymbol{c}) \leq f_i(\boldsymbol{y}, \boldsymbol{c})$ and $f_j(\boldsymbol{x}, \boldsymbol{c}) < f_j(\boldsymbol{y}, \boldsymbol{c})$ for at least one index $j \in \{1, 2, \ldots, k\}$.
- Pareto optimality: a solution $\boldsymbol{x}_{\boldsymbol{c}}^* \in \boldsymbol{\Omega}$ is said to be Pareto optimal for the condition vector $\boldsymbol{c} \in \boldsymbol{\Phi}$, if and only if $\nexists \boldsymbol{x} \in \boldsymbol{\Omega}$ such that $\boldsymbol{x} \preceq_{\boldsymbol{c}} \boldsymbol{x}_{\boldsymbol{c}}^*$.

As in the MO optimization problem, solving an MCMO optimization problem is to find a Pareto optimal set and the corresponding Pareto front which is defined as $\{\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{c}}, \boldsymbol{c})|\ \boldsymbol{x}_{\boldsymbol{c}} \in \text{Pareto optimal set}\}$. If $n(\boldsymbol{\Phi}) = 1$, the MCMO optimization problem is reduced to an MO optimization problem.

4. Multi-condition multi-objective optimization method

4.1. Deep-reinforcement-learning algorithm

4.1.1. State, action, and reward

In MCMO optimization, an optimal solution exists per condition and weight vector which determines weights among objectives. Therefore, the state of DRL is defined as follows:

$$\boldsymbol{s} = [\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{f}^*], \tag{5}$$

where $\boldsymbol{c}$ is a condition vector, $\boldsymbol{w}$ is a weight vector, and $\boldsymbol{f}^*$ is a utopia point vector at that condition. In the present study, $\boldsymbol{f}^*$ is adaptively updated during optimization to overcome the difficulty that $\boldsymbol{f}^*$ should be known in advance as described in Section 2.1.2. It is updated to a slightly lower value than the minimum value of each objective function. Since the Chebyshev scalarizing function $f_{Chebyshev}$ varies depending on $\boldsymbol{f}^*$ as in Eq. (2), the changing utopia information is included in the state for stable learning.

The action of DRL determines $\boldsymbol{x}$ which is an element vector in the decision space according to its policy as follows:

$$\boldsymbol{a} = [\boldsymbol{x}]. \tag{6}$$

In addition, all the variables in the state and the action are normalized to an absolute magnitude around 1 for scaling.

Finally, the reward of DRL is a quantitative evaluation of an action, which is defined as follows:

$$r = -f_{Chebyshev} = -\max_{i=1,\ldots,k} w_i |f_i(\boldsymbol{x}) - f_i^*|, \tag{7}$$

where $f_i(\boldsymbol{x})$ is the value of the objective function obtained by executing an action. Note that the minus sign is added because the aim of optimization is to find $\boldsymbol{x}$ minimizing the Chebyshev scalarizing function.

4.1.2. Data reproduction method

In the present study, a data reproduction method is applied to enlarge the amount and the diversity of data by exploiting the nature of the Chebyshev scalarizing function $f_{Chebyshev}$. As $\boldsymbol{f}$ is independent of $\boldsymbol{w}$, it is possible to reproduce an original data of $(\boldsymbol{s}, \boldsymbol{a}, r)$ for arbitrary $\boldsymbol{w}$ once $\boldsymbol{f}$ is evaluated. This method accelerates learning by ensuring the amount and diversity of data and, thus, is essential for optimization problems where the function evaluation is costly. In the present study, at each episode, 100 data are reproduced from a single original data by selecting $\boldsymbol{w}$ in a uniform distribution between 0 and 1.

4.1.3. Learning procedure

Algorithm 1: Single-step DRL for MCMO optimization.

```
initialize actor network $\pi_\phi$ and critic network $Q_\theta$ with random parameters $\phi$, $\theta$;
initialize episode, utopia $\boldsymbol{f}^*$, and replay buffer $\mathcal{B}$;
repeat
    episode ← episode+1;
    select condition and weight: $\boldsymbol{c} \sim \mathrm{U}(0,1)$, $\boldsymbol{w} \sim \mathrm{U}(0,1)$ $s.t.$ $|\boldsymbol{w}| = 1$;
    calculate utopia $\boldsymbol{f}^*(\boldsymbol{c})$;
    receive state: $\boldsymbol{s} \leftarrow [\boldsymbol{c}, \boldsymbol{w}, \boldsymbol{f}^*]$;
    select action with exploration noise: $\boldsymbol{a} \leftarrow \mathrm{clip}(\pi_\phi(\boldsymbol{s}) + \boldsymbol{\epsilon}, -1, 1)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$;
    execute action and evaluate objective function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{c})$;
    $\forall i$, if $f_i(\boldsymbol{x}, \boldsymbol{c}) < f_i^*(\boldsymbol{c})$, update utopia and state;
    calculate reward and reproduce data;
    store data of $(\boldsymbol{s}, \boldsymbol{a}, r)$ in $\mathcal{B}$;
    for $l = 1$ to $l_{max}$ do
        sample mini-batch of $N_b$ data from $\mathcal{B}$;
        update $\theta$ with the loss $N_b^{-1} \sum (r - Q_\theta(\boldsymbol{s}, \boldsymbol{a}))^2$;
        if $l$ mod $l_d$ then
            update $\phi$ by the deterministic policy gradient $N_b^{-1} \sum \nabla_{\boldsymbol{a}} Q_\theta(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\pi_\phi(\boldsymbol{s})} \nabla_\phi \pi_\phi(\boldsymbol{s})$;
        end
    end
until convergence;
```

The learning procedure of the present method is summarized in Algorithm 1. For DRL, the actor-critic algorithm [44] is used. In the algorithm, two types of neural networks are employed. One is an actor network which determines an action in a continuous action space. The other is a critic network which predicts the value function depending on the state and the action. As learning progresses, the critic network predicts the value function more and more accurately, and therefore, the probability that the actor network selects an optimal action increases.

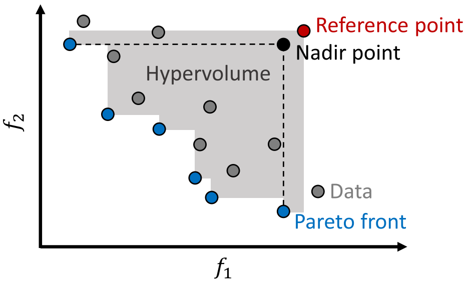


Fig. 1. Schematic illustration of the hypervolume (HV) in the plane of two objectives f_1 and f_2.

Both networks are set as fully connected networks with four hidden layers of 512, 256, 256, and 128 neurons, and the Leaky ReLU activation function [45] is used for the hidden layers in both networks. Although two hidden layers have been generally used in the past DRL studies [46,47], the present study employs four hidden layers to predict optimal solutions of problems with nonlinear characteristics accurately. At the output layer of the actor network, the Tanh activation function is added so that the action values range from -1 to 1. The Adam optimizer [48] with a learning rate of 10^{-4} which has been commonly used in actor-critic algorithms [46,47] is used for updating the network parameters. The actor network is updated every two learning iterations ($l_d = 2$) for stable learning [47]. The mini-batch size N_b is set to 100, and the learning amount per episode l_{max} is set to 100 which is the same as the number of reproduced data from one original data. The standard deviation of the exploration noise σ is set to 1 in the initial 1000 warm-up episodes and $0.05(\cos(2\pi/1000 \times \text{episode}) + 1)$ afterward. During the warm-up episodes, actions are determined almost randomly with large noise to gather a variety of data. Afterward, a cosine function is used to avoid local minima and to find optimal solutions accurately by periodically manipulating the magnitude of the noise.

4.2. Selection of Pareto front

Pareto dominance in an MCMO optimization problem is defined at each condition vector $\boldsymbol{c} \in \boldsymbol{\Phi}$. However, as $\boldsymbol{c}$ in the obtained data are scattered over $\boldsymbol{\Phi}$, there is no exactly the same $\boldsymbol{c}$ where the dominance can be judged. Therefore, a concept of decomposition of the condition space is introduced to derive approximate solutions for an MCMO optimization problem. $\boldsymbol{\Phi}$ is decomposed into N subspaces as follows:

$$\begin{aligned} &\boldsymbol{\Phi}_1 \cup \boldsymbol{\Phi}_2 \cup \ldots \cup \boldsymbol{\Phi}_N = \boldsymbol{\Phi}, \\ &\boldsymbol{\Phi}_i \cap \boldsymbol{\Phi}_j = \varnothing \text{ where } i \neq j. \end{aligned} \tag{8}$$

In each subspace $\boldsymbol{\Phi}_i$, Pareto front is selected from the data by assuming $\boldsymbol{c}$ is the same.

Note that the decomposition has no effect on the optimization process and can be modified during or after the optimization process. Therefore, N can be freely adjusted according to the desired quality. For example, the denser the decomposition, the higher the resolution of the selected Pareto front, but the number of episodes required for convergence increases. In the present study, $\boldsymbol{\Phi}$ is uniformly decomposed into 100 subspaces for selecting Pareto front.

4.3. Convergence judgment

To judge the convergence of an optimization process, the hypervolume indicator (HV) [49] is adopted. It refers to the volume in the objective space between Pareto front and a fixed reference point as shown in Fig. 1. Due to its monotonic characteristic, the larger the HV, the more accurate the Pareto front. Therefore, it is one of the most frequently used indicators for convergence and capability assessment of MO optimization methods. A general guideline for determining the reference point is to use a slightly worse point than the nadir point consisting of the worst objective values over the Pareto front [50].

In MCMO optimization, as described in Section 4.2, the condition space $\boldsymbol{\Phi}$ is decomposed into subspaces and Pareto front is selected respectively in each decomposed subspace. Likewise, the HV is defined in each decomposed subspace. Therefore, in this study, the convergence of an optimization process is judged by the average HV over all the decomposed subspaces (HV$_{avg}$).

5. Results and discussion

In this section, the proposed DRL-based MCMO optimization method is applied to two problems, and the results are analyzed. The first problem is a new benchmark problem for MCMO optimization which is derived by modifying the Kursawe test function. The second problem is airfoil shape optimization which is a representative shape optimization problem involving fluid mechanics.

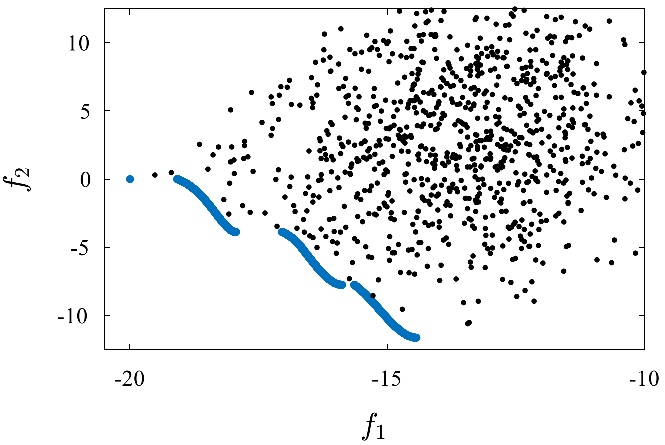


Fig. 2. Exact Pareto front (blue lines) and random data (black dots) of the original Kursawe test problem.

5.1. Modified Kursawe test problem

5.1.1. Problem setup

The original Kursawe test problem [34], which is a representative MO optimization problem, is defined as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = \min_{\boldsymbol{x}}(f_1(\boldsymbol{x}), f_2(\boldsymbol{x})) \\ &\qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega}, \end{aligned} \tag{9}$$

where

$$\begin{aligned} f_1(\boldsymbol{x}) &= \sum_{i=1}^{2}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})), \\ f_2(\boldsymbol{x}) &= \sum_{i=1}^{3}(|x_i|^{0.8} + 5\sin(x_i^3)), \\ \boldsymbol{\Omega} &= \{(x_1, x_2, x_3)| -5 \le x_i \le 5,\ i = 1, 2, 3\}. \end{aligned}$$

Fig. 2 shows the Pareto front and random data of the Kursawe test problem. It has discontinuous and nonconvex Pareto front, and the objective functions f_1 and f_2 vary greatly depending on the decision vector $\boldsymbol{x}$. Due to these characteristics, the Kursawe test problem has been actively adopted to evaluate the capability of MO optimization methods [51–54].

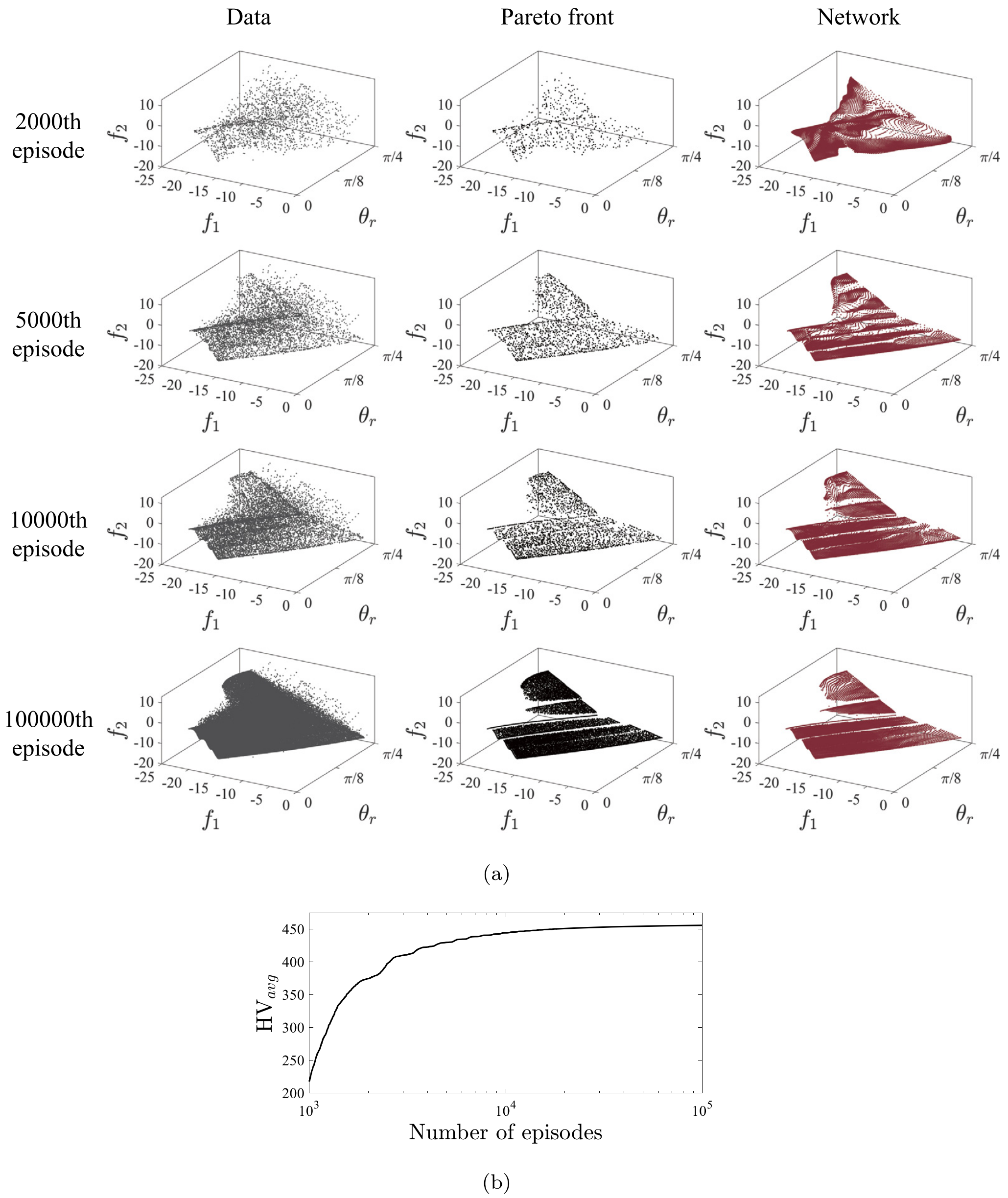
In the present study, the original Kursawe test problem is extended for MCMO optimization as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{c}) = \min_{\boldsymbol{x}}(f_1(\boldsymbol{x}, \boldsymbol{c}), f_2(\boldsymbol{x}, \boldsymbol{c})) \\ &\qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega}, \boldsymbol{c} \in \boldsymbol{\Phi}, \end{aligned} \tag{10}$$

where

$$\begin{aligned} f_1(\boldsymbol{x}, \boldsymbol{c}) &= g_1(\boldsymbol{x})\cos\theta_r - g_2(\boldsymbol{x})\sin\theta_r, \\ f_2(\boldsymbol{x}, \boldsymbol{c}) &= g_1(\boldsymbol{x})\sin\theta_r + g_2(\boldsymbol{x})\cos\theta_r, \\ g_1(\boldsymbol{x}) &= \sum_{i=1}^{2}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})), \\ g_2(\boldsymbol{x}) &= \sum_{i=1}^{3}(|x_i|^{0.8} + 5\sin(x_i^3)), \\ \boldsymbol{\Omega} &= \{(x_1, x_2, x_3)| -5 \le x_i \le 5,\ i = 1, 2, 3\}, \\ \boldsymbol{\Phi} &= \{(\theta_r)|\ 0 \le \theta_r \le \pi/4\}. \end{aligned}$$

For the modification, θ_r is introduced for rotational transformation. If θ_r is set to 0, it reduces to the original Kursawe test problem. Extending the problem through rotational transformation has two advantages. Firstly, characteristics of the original



(a)

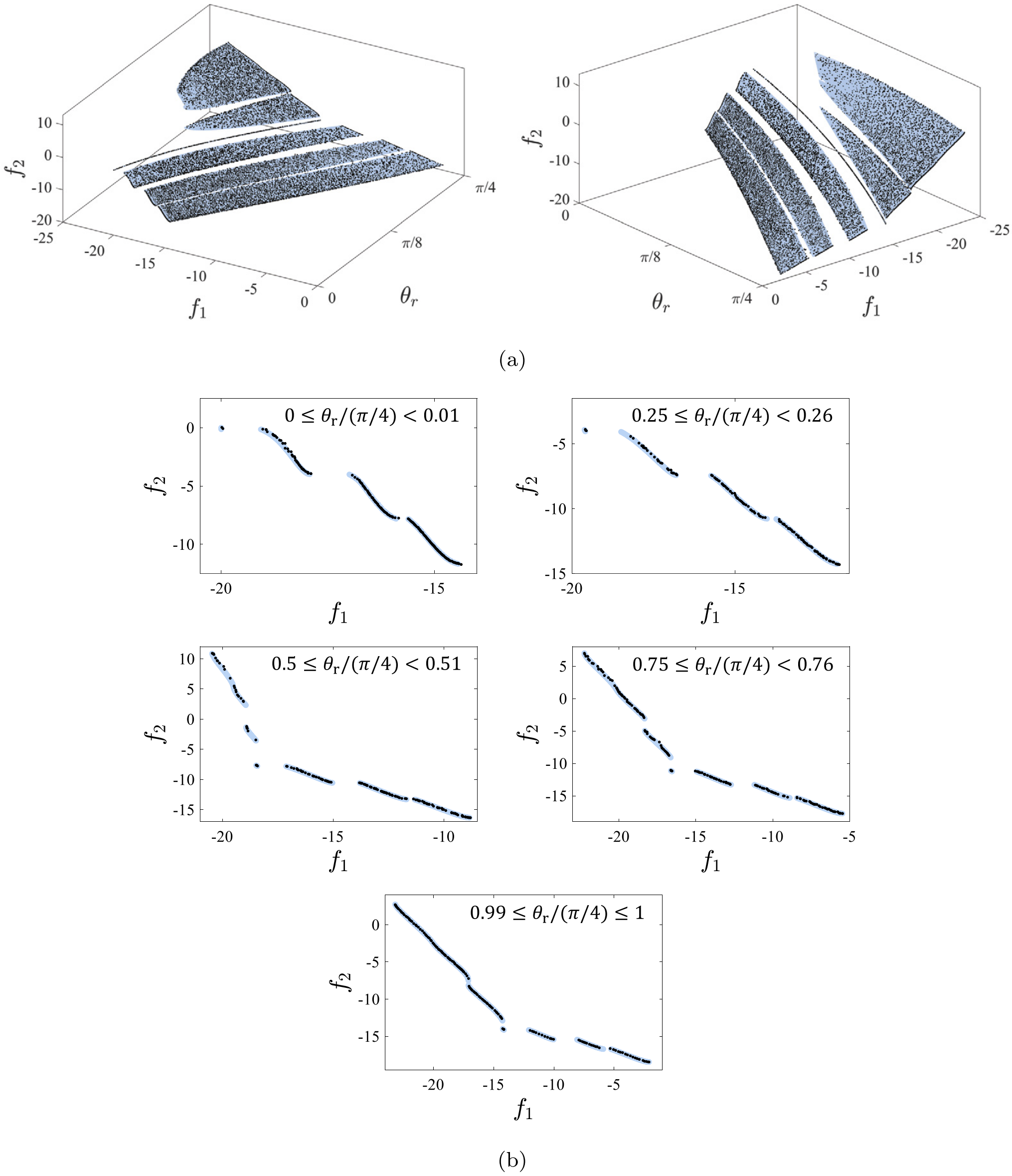
(b)

Fig. 3. Optimization process for the modified Kursawe test problem. (a) Results during the optimization process. One data point is added for every episode. Pareto front is selected from the data as described in Section 4.2. The network shows values of objective functions corresponding to actions determined by the network. (b) Average HV (HV_{avg}) as a function of episode with a reference point (f_1^{ref}, f_2^{ref}) of (-2, 13).

problem are preserved. The discontinuity and nonconvexity of the original Pareto front remain unchanged. Secondly, exact solutions can be readily obtained, which is crucial in designing a benchmark problem. The boundary shape of the feasible region in the objective space remains unchanged from the original Kursawe test problem by the rotational transformation. Therefore, the exact Pareto front corresponding to θ_r can be obtained by judging dominance from the rotated boundary.

5.1.2. Optimization results

The modified Kursawe test problem is solved as described in Algorithm 1. As shown in Fig. 3a, in the early episodes, data are widely scattered as the network is not developed enough. However, as optimization proceeds, data are accumulated, and the network learns a policy to find optimal solutions within the condition space. As the result, better solutions are obtained for newly given conditions and objectives, thereby increasing the resolution of the Pareto front. Finally, the Pareto front and the network converge at the 100000th episode, which can be also seen in terms of HV_{avg} as shown in Fig. 3b.



(a)

(b)

Fig. 4. Optimization results of the modified Kursawe test problem. (a) Pareto front from two different angles in the domain of two objectives f_1 and f_2 and the condition θ_r. (b) Pareto front in five decomposed condition subspaces. Black dots in (a) and (b) show Pareto front obtained at the 100000th episode. Blue surfaces in (a) and blue lines in (b) show exact Pareto front. Note that the obtained Pareto front shows good agreement with the exact Pareto front.

Fig. 4 shows the optimization results at the converged episode. Through the optimization, high-resolution Pareto front of 15268 solutions over the entire condition space is obtained. As shown in Fig. 4a, it shows good agreement with the exact Pareto front including highly nonlinear parts near $\theta_r/(\pi/4) = 0.4$ where the shape of the Pareto front drastically changes along θ_r. It shows the exclusive ability of the proposed MCMO optimization method. If several representative conditions are predetermined and optimization is performed at each condition, it is difficult to capture such nonlinear parts. Fig. 4b shows the optimization results in five decomposed condition subspaces. Note that since the condition space $\mathbf{\Phi}$ is equally decomposed into 100 subspaces in the present study, each figure in Fig. 4b shows one decomposed condition subspace. Note also that in those decomposed subspaces, the solutions obtained using the present method match well with the exact Pareto front.

5.1.3. Effectiveness of multi-condition optimization

In this section, a computational experiment based on the modified Kursawe test problem is set to analyze how effective MC optimization is compared to single-condition (SC) optimization. The experiment is designed to compare the number of function evaluations to reach the same quality of optimization. Since SC optimization cannot be performed over a condition

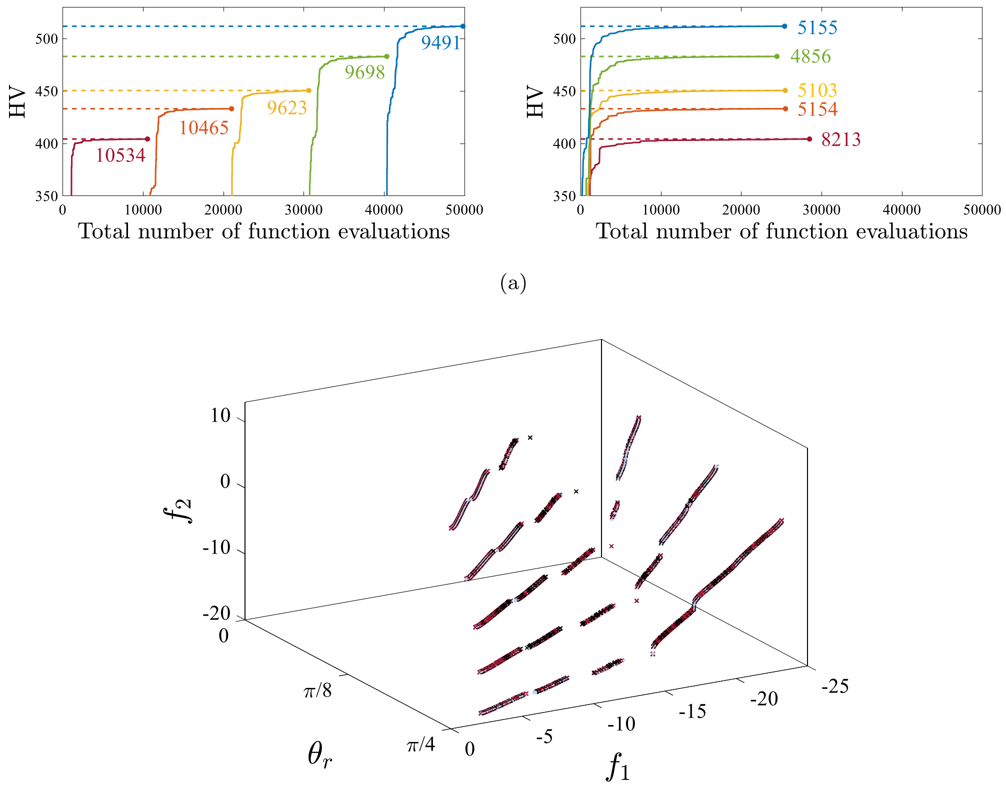


Fig. 5. Optimization results of the computational experiment with five selected conditions ($N_c = 5$). (a) HVs as a function of the total number of function evaluations with a reference point (f_1^{ref}, f_2^{ref}) of (-2, 13) for the five selected conditions. ——, $\theta_r/(\pi/4) = 0/4$; ——, $\theta_r/(\pi/4) = 1/4$; ——, $\theta_r/(\pi/4) = 2/4$; ——, $\theta_r/(\pi/4) = 3/4$; ——, $\theta_r/(\pi/4) = 4/4$. The left figure shows HV from single-condition (SC) optimization and the right figure shows HV from multi-condition (MC) optimization. Dashed lines in both figures indicate $\mathbf{HV}^{ref}$. If the HV at a specific condition satisfies $\mathbf{HV}^{ref}$, optimization at the condition terminates. Numbers near the lines are the numbers of function evaluations at each condition. (b) Pareto front obtained from two cases. Black crosses show the Pareto front obtained from SC optimization and red crosses show the Pareto front obtained from MC optimization. Blue lines show the exact Pareto front at the five selected conditions. Note that all the symbols overlap.

space, equally distributed N_c conditions are selected in the condition space $\mathbf{\Phi}$ for the comparison. To measure the quality of optimization, a reference HV (HV_i^{ref}) is set at each selected condition.

Two cases are compared by the total number of function evaluations required to reach the same $\mathbf{HV}^{ref}$ at all selected conditions. The first case is multiple operations of SC optimization, which are performed independently at each selected condition. The SC optimization method can be easily derived by fixing the condition in the present method. The second case is a single operation of MC optimization, which is modified to be conducted at the selected conditions. When comparing the two cases, the average number of total function evaluations of ten runs is used since the optimization process is stochastic due to exploration of DRL.

Fig. 5 shows an example of the results when $N_c = 5$. As shown in Fig. 5a, SC optimization is performed independently while MC optimization is performed simultaneously. Note that about 10000 function evaluations are required for the convergence of SC optimization at each condition, which is comparable to other studies dealing with the Kursawe test problem [51–54]. Note also that the number of function evaluations at each condition is reduced in the MC optimization, resulting in a significant reduction of the total number of function evaluations. Total 49811 function evaluations are required in the SC optimization while total 28481 function evaluations are required to reach the same $\mathbf{HV}^{ref}$ in the MC optimization. This reduction is attributed to the fact that the MC optimization learns the correlations between the conditions and the optimal solutions. By utilizing the correlations, it can effectively find Pareto front with a small number of function evaluations. Fig. 5b shows that the obtained Pareto front shows good agreement with exact Pareto front in both cases.

Fig. 6 shows results obtained from a parameter study with N_c. In the SC optimization, the number of function evaluations increases linearly with N_c. This is because optimization is performed at each condition independently. However, in the MC optimization, the increment gradually decreases, so that the difference between the two cases increases with N_c. Especially, when $N_c = 17$, the number of function evaluations of the MC optimization is only 33% of that of the SC optimization.

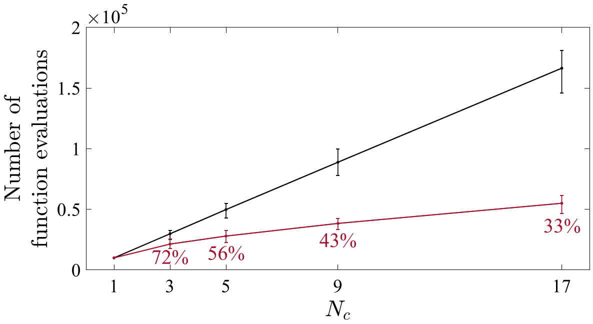


Fig. 6. The numbers of function evaluations of SC optimization and MC optimization, respectively, required to reach the same $\mathbf{HV}^{ref}$ as a function of N_c. The black line indicates the average of ten runs of SC optimization and the red line indicates the average of ten runs of MC optimization. Error bars show the minimum and maximum values over ten runs. Each number below the red line is the percentage of the number of function evaluations in MC optimization compared to that in SC optimization.

5.2. Airfoil shape optimization

5.2.1. Problem setup

In many aeronautical applications, the flow condition and the aerodynamic requirement of an airfoil can vary depending on the operation situation. In this section, an MCMO problem for airfoil shape optimization is considered. The lift coefficient C_L and the lift-to-drag ratio C_L/C_D are set as objectives of optimization to be maximized. These objectives are crucial factors in characterizing the aerodynamic performance of an airfoil [55–58]. The chord-based Reynolds number Re_c is considered as a condition variable of optimization.

In this study, an airfoil shape is parameterized using the Kármán-Trefftz transformation [59]. A Kármán-Trefftz airfoil is generated from transformation of a circle in the ζ-plane to the physical z-plane. The circle in the ζ-plane is centered on (μ_x, μ_y) and passes $(1, 0)$. A complex variable $\zeta = \xi + i\eta$ on the circle is transformed to a complex variable $z = x + iy$ to generate an airfoil as follows:

$$\begin{aligned} z &= n\frac{(\zeta+1)^n + (\zeta-1)^n}{(\zeta+1)^n - (\zeta-1)^n}, \\ n &= 2 - \frac{\beta}{\pi}, \end{aligned} \tag{11}$$

where β is a trailing-edge angle of the generated airfoil. μ_x is a factor that determines the thickness of the airfoil. The smaller the absolute value, the thinner airfoil is generated. μ_y determines the camber of the airfoil. $\mu_y = 0$ indicates a symmetric airfoil, and the larger the value, the more cambered airfoil is generated. Since the Kármán-Trefftz transformation can generate various and realistic airfoils, it has been utilized in many studies [60,61].

Along with the shape of an airfoil, the angle of attack α is an important factor that greatly influences the aerodynamic characteristics. Thus, when designing an airfoil, α relative to the flow direction should be optimized to achieve optimal performance [58]. In the present study, in addition to μ_x, μ_y, and β which determine a Kármán-Trefftz airfoil, α is included as a design variable of optimization.

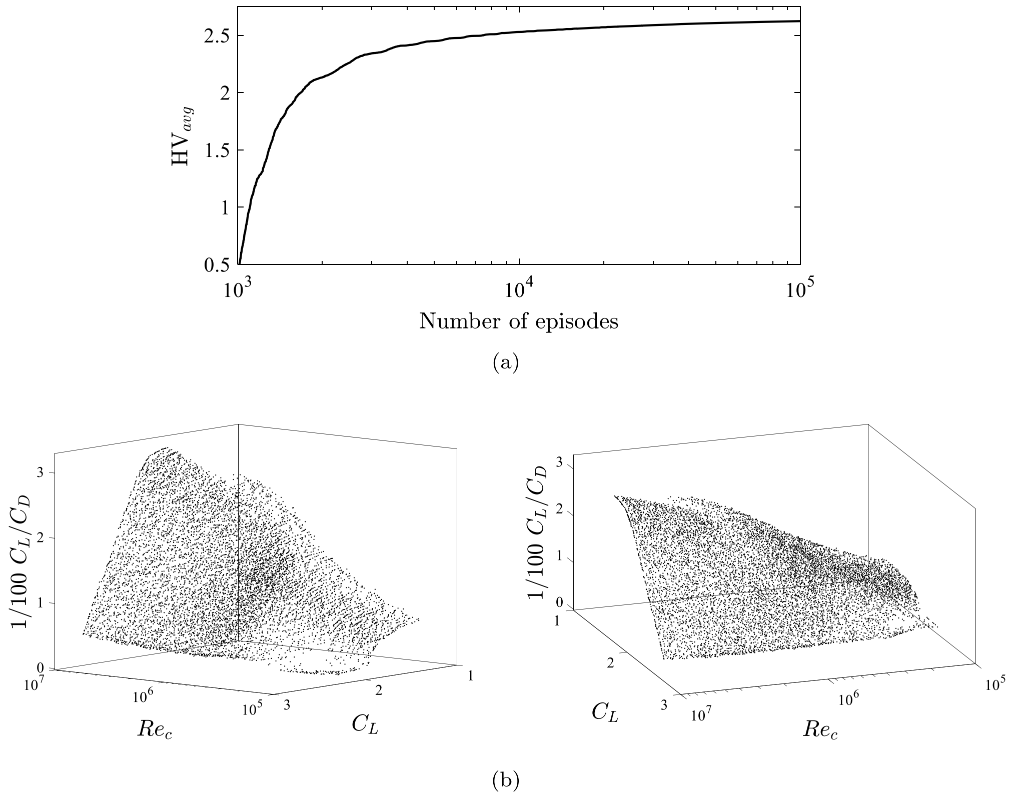
The MCMO problem for airfoil shape optimization is defined as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{c}) = \min_{\boldsymbol{x}} (f_1(\boldsymbol{x}, \boldsymbol{c}), f_2(\boldsymbol{x}, \boldsymbol{c})) \\ &\qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega}, \boldsymbol{c} \in \boldsymbol{\Phi}, \end{aligned} \tag{12}$$

where

$$\begin{aligned} f_1(\boldsymbol{x}, \boldsymbol{c}) &= -1/100\, C_L/C_D, \\ f_2(\boldsymbol{x}, \boldsymbol{c}) &= -C_L, \\ \boldsymbol{\Omega} = \{(\mu_x, \mu_y, \beta, \alpha)| &-0.4 \le \mu_x \le -0.05, \\ 0 \le \mu_y \le 0.4,\ 1^\circ &\le \beta \le 30^\circ,\ 0^\circ \le \alpha \le 30^\circ\}, \\ \boldsymbol{\Phi} &= \{(Re_c)|\ 10^5 \le Re_c \le 10^7\}. \end{aligned}$$

1/100 is multiplied to C_L/C_D to match the scale between $f_1(\boldsymbol{x}, \boldsymbol{c})$ and $f_2(\boldsymbol{x}, \boldsymbol{c})$. To evaluate C_L/C_D and C_L, XFOIL which is an analysis tool for airfoils [62] is adopted in the present study. It is widely used for calculating airfoil performance due to its low computational cost [63–65]. The ranges of the design variables μ_x, μ_y, β, and α are determined to generate various airfoil shapes while excluding unrealistic shapes such as excessively thin and cambered airfoils. The range of Re_c is determined to cover wide applications from wind turbines to general aviation [66].



(a)

(b)

Fig. 7. Results from airfoil shape optimization. (a) Average HV (HV$_{avg}$) as a function of episode with a reference point (f_1^{ref}, f_2^{ref}) of $(0, -1)$. (b) Pareto front at the 100000th episode from two different angles in the domain of two objectives 1/100 C_L/C_D and C_L and the condition Re_c.

5.2.2. Optimization results

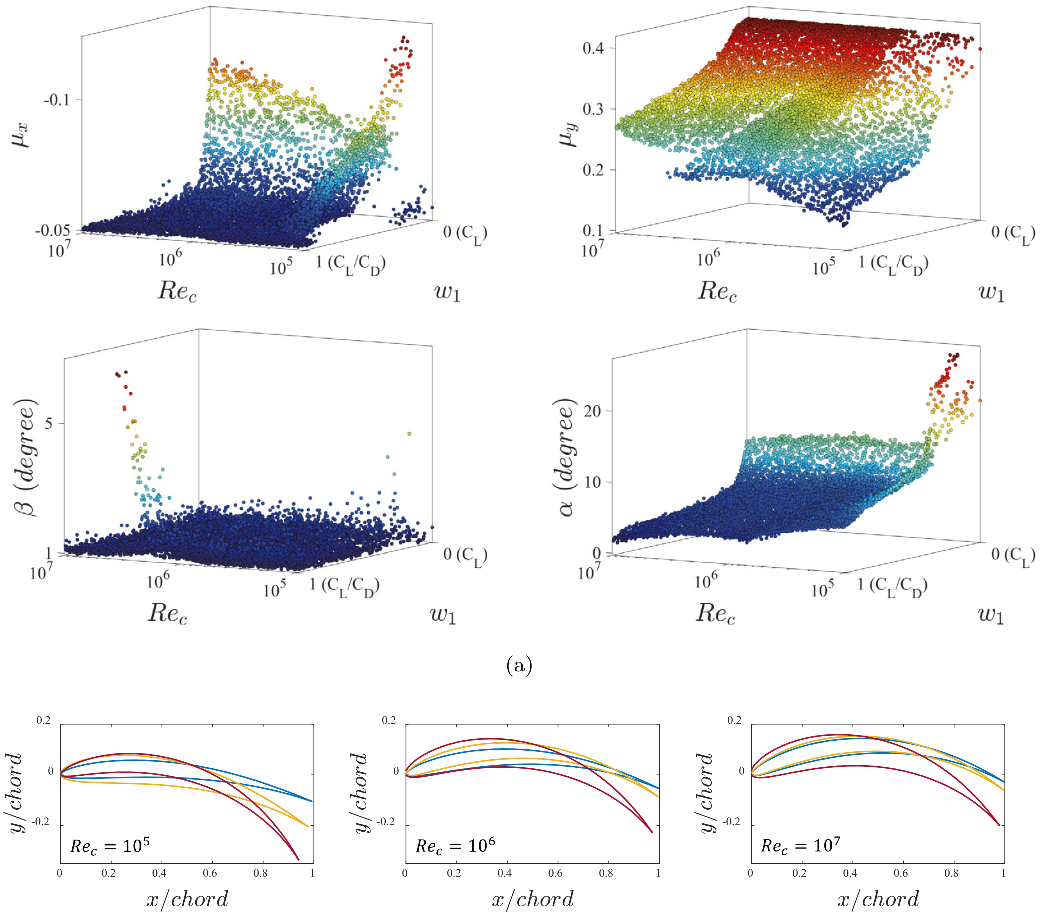
The airfoil shape optimization is conducted as described in Algorithm 1. As shown in Fig. 7a, HV$_{avg}$ is shown to converge at the 100000th episode. Fig. 7b shows Pareto front at the converged episode. Overall, Pareto front of sufficient resolution is successfully obtained within the Reynolds number space. As shown in Fig. 7b, the maximum C_L/C_D increases with Re_c while the maximum C_L remains relatively constant. In particular, along the line where C_L/C_D is maximized, C_L does not change significantly, which refers that C_D decreases with Re_c. In addition, two distinct features are observed in the Pareto front. When C_L is maximized, nonconvex parts are observed near $Re_c = 10^5$. Also, when C_L/C_D is maximized, nonlinear parts are observed near $Re_c = 2 \times 10^6$. These parts will be further discussed through analysis of the optimal airfoil shapes.

As shown in Fig. 8a, various values of optimal design parameters are obtained depending on Re_c and w_1. Also, nonlinear features are observed where the parameters change dramatically with respect to Re_c and w_1. These are particularly noticeable for μ_x and α at around $Re_c = 10^5$ and $w_1 = 0$ and for μ_y and β at around $Re_c = 2 \times 10^6$ and $w_1 = 1$. These parts correspond to the aforementioned nonconvex and nonlinear parts observed in the Pareto front, respectively. Except for the parts, several noticeable trends are observed. As the weight of C_L/C_D increases, thinner and less cambered airfoils with low α are generated. On the other hand, As the weight of C_L increases, thicker and more cambered airfoils with high α are generated. The trailing-edge angle β shows relatively small variation from the minimum value.

Fig. 8b shows the optimal airfoil shapes at different Re_c and w_1. As w_1 gets closer to 0, airfoils with higher camber and higher α are generated to maximize the lift at all Re_c. However, values of α do not increase to the maximum, which is due to consideration of stall caused at excessively high α. On the contrary, as w_1 gets closer to 1, the opposite tendency is observed due to the drag. When the drag is considered, thinner airfoils are generated except for $Re_c = 10^5$ where the nonlinearity in the Pareto front exists.

5.2.3. Analysis of aerodynamic performance of optimal airfoils

In this section, based on the optimization results in Section 5.2.2, the importance of MC optimization is demonstrated. For this goal, it is analyzed whether optimal shapes determined at some representative conditions can provide sufficient performance over the entire condition space. For the analysis, a typical optimization problem encountered in many aviation applications is considered. C_L/C_D is often maximized while maintaining a certain level of C_L to sustain the weight of an aircraft [58,67,68]. In the present analysis, the optimal airfoil shapes that maximize C_L/C_D with a constraint $C_L \geq 2$ are selected in the obtained Pareto front as shown in Fig. 9a.



(a)

(b)

Fig. 8. Optimal design parameters and optimal airfoil shapes. (a) Optimal design parameters as a function of Re_c and w_1. w_1 is the weight of C_L/C_D. $w_1 = 1$ corresponds to maximization of C_L/C_D while $w_1 = 0$ corresponds to maximization of C_L. (b) Optimal airfoil shapes at different Re_c and w_1. —, $w_1 = 0$; —, $w_1 = 0.5$; —, $w_1 = 1$. The airfoils are depicted according to α assuming that the flow direction is horizontal from left to right.

Black lines in Fig. 9b indicate the aerodynamic performance of the selected optimal shapes. Among them, two shapes corresponding to $Re_c = 2 \times 10^5$ and $Re_c = 2 \times 10^6$ are analyzed. Red lines show aerodynamic performance of the shape at $Re_c = 2 \times 10^5$ calculated over the Reynolds number space. Compared to black lines, both C_L and C_L/C_D decrease notably except for $Re_c = 2 \times 10^5$. In particular, there is a significant reduction in C_L at Re_c slightly lower than 2×10^5. In a similar way, blue lines represent aerodynamic performance of the shape at $Re_c = 2 \times 10^6$. C_L/C_D substantially decreases at Re_c slightly higher than 2×10^6.

Fig. 9c shows optimal airfoil shapes at different Re_c. The optimal shape at $Re_c = 2 \times 10^5$ quite differs from the optimal shape at $Re_c = 10^5$, which results in a huge difference in C_L. However, the optimal shape at $Re_c = 2 \times 10^6$ is very similar to the optimal shape at $Re_c = 3 \times 10^6$, although the difference in C_L/C_D is large. Similarly, except for the optimal shape at $Re_c = 10^5$, there is no noticeable difference among the other optimal shapes while there exist significant performance differences. These results are attributed to the nonlinear characteristics of the present problem involving fluid mechanics. The optimal shape can drastically vary according to the condition, and even a slight variation in the shape can make a huge performance difference.

Through the analysis, it is shown that an optimal shape obtained at a specific condition may not be valid at the nearby conditions, which can be more severe in problems that have nonlinear characteristics. Therefore, it is essential to consider the entire condition space through the MC optimization method to avoid significant degradation of target performance.

5.3. Airfoil shape optimization with multiple condition variables

In the problems we have dealt with, the condition vector $\boldsymbol{c}$ consists of one condition variable. However, the proposed method can handle multiple condition variables. In airfoil shape optimization, angle of attack α is often considered as a

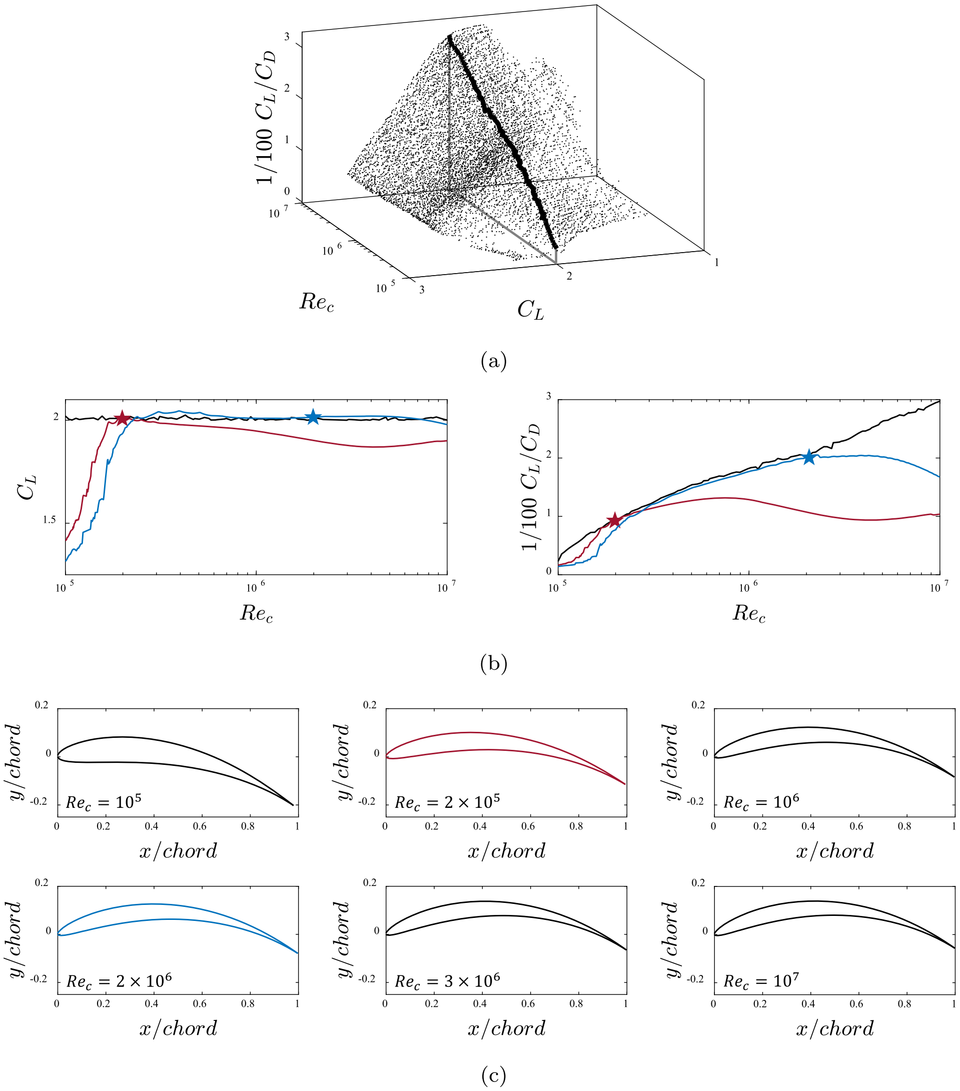


Fig. 9. Performance analysis of optimal airfoil shapes. (a) Finding optimal shapes that maximize C_L/C_D with a constraint $C_L \geq 2$ in Pareto front. The black line corresponds to the optimal shapes. (b) C_L and C_L/C_D of optimal shapes as a function of Re_c. Black lines correspond to the optimal shapes obtained in (a). Red and blue lines correspond to each optimal airfoil at $Re_c = 2 \times 10^5$ and $Re_c = 2 \times 10^6$, respectively, depicted as stars. (c) Optimal airfoil shapes at various Re_c. Note that red and blue airfoils correspond to the airfoils selected in (b). The airfoils are depicted according to α assuming that the flow direction is horizontal from left to right.

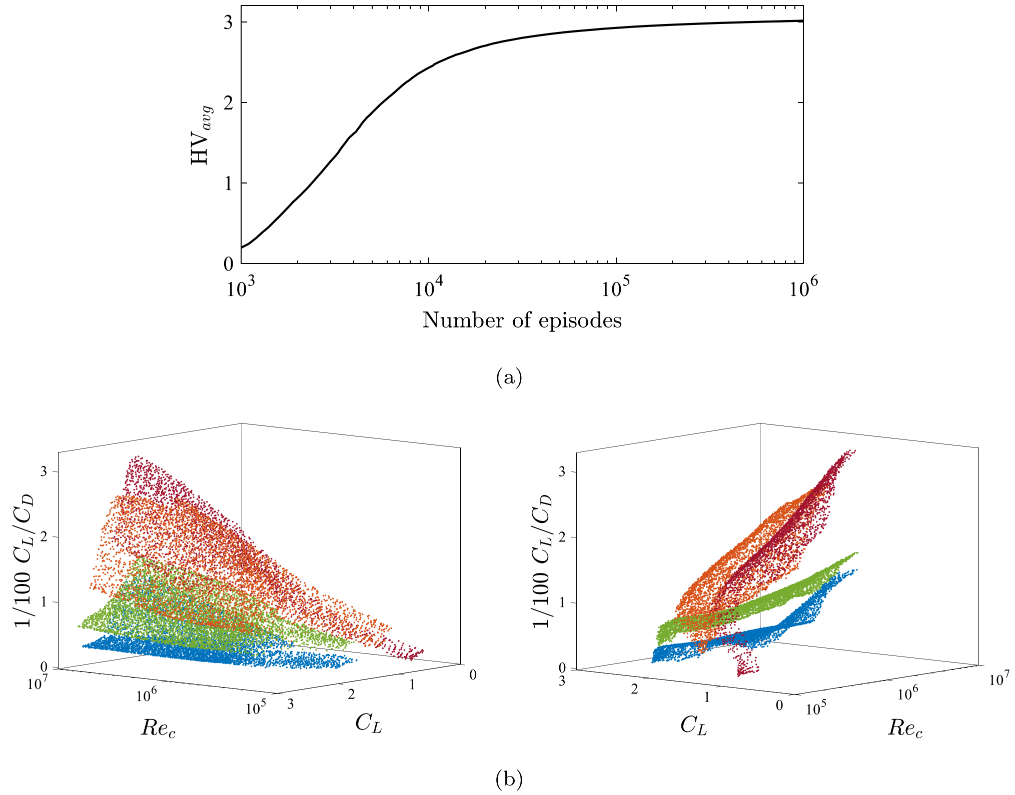
given condition along with Re_c [69,70]. In this section, an airfoil shape optimization problem, where both Re_c and α are considered as condition variables, is solved using the proposed method.

The optimization problem is defined as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{c}) = \min_{\boldsymbol{x}}(f_1(\boldsymbol{x}, \boldsymbol{c}), f_2(\boldsymbol{x}, \boldsymbol{c})) \\ &\qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega}, \boldsymbol{c} \in \boldsymbol{\Phi}, \end{aligned} \tag{13}$$

where

$$\begin{aligned} f_1(\boldsymbol{x}, \boldsymbol{c}) &= -1/100\, C_L/C_D, \\ f_2(\boldsymbol{x}, \boldsymbol{c}) &= -C_L, \\ \boldsymbol{\Omega} = \{(\mu_x, \mu_y, \beta) | &-0.4 \leq \mu_x \leq -0.05,\ 0 \leq \mu_y \leq 0.4,\ 1^\circ \leq \beta \leq 30^\circ\}, \\ \boldsymbol{\Phi} = \{(Re_c, \alpha) | &\ 10^5 \leq Re_c \leq 10^7,\ 0^\circ \leq \alpha \leq 15^\circ\}. \end{aligned}$$



(a)

(b)

Fig. 10. Airfoil shape optimization with multiple condition variables. (a) Average HV (HV$_{avg}$) as a function of episode with a reference point (f_1^{ref}, f_2^{ref}) of $(0, 0)$. (b) Pareto front at the 1000000th episode from two different angles in the domain of two objectives $1/100\ C_L/C_D$ and C_L and the condition variable Re_c at four α subspaces. ●, $0° \le \alpha < 0.15°$; ●, $4.95° \le \alpha < 5.1°$; ●, $9.9° \le \alpha < 10.05°$; ●, $14.85° \le \alpha \le 15°$.

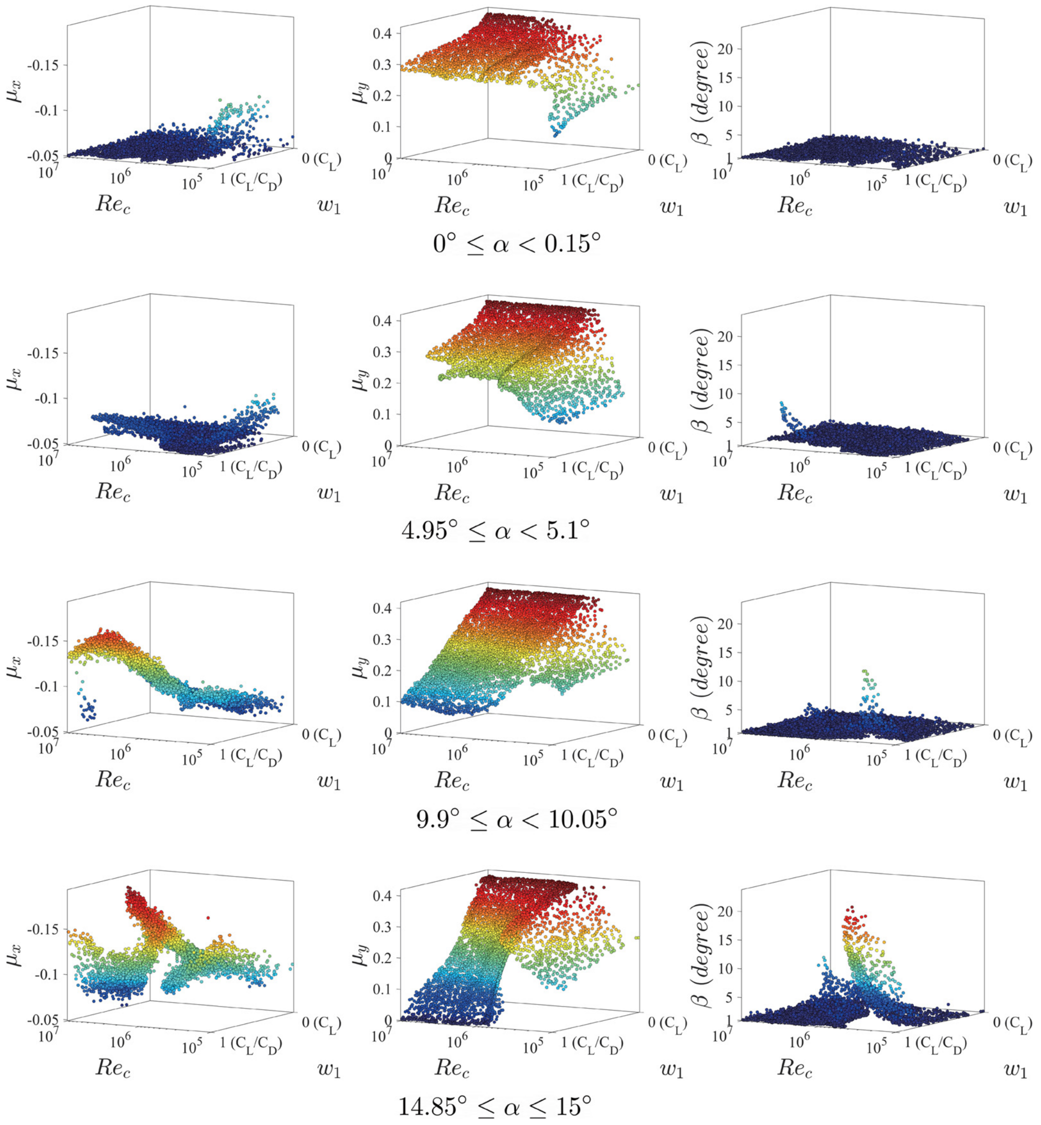
The optimization is conducted as described in Algorithm 1, and the results are shown in Fig. 10. To select Pareto front, condition space $\mathbf{\Phi}$ is uniformly decomposed into 100 subspaces for each condition variable, resulting in 10000 subspaces. As shown in Fig. 10a, HV$_{avg}$ is shown to converge at the 1000000th episode. Fig. 10b shows Pareto front at the converged episode at four α subspaces. The shape of Pareto front varies considerably with α.

Fig. 11 shows optimal design parameters at four α subspaces. Depending on α, various values of optimal design parameters are obtained. The distributions of design parameters differ with the results of Section 5.2.2 in which α is not considered as a condition variable. Overall, the absolute value of μ_x and β increase as α increases. In particular, the maximum value of β significantly increases up to 20°, which means much thicker airfoils are generated. In the case of μ_y, as α increases, the minimum value decreases to 0, resulting in symmetric airfoils.

6. Concluding remarks

For the first time in the literature, an MCMO optimization method based on DRL has been developed to find Pareto front over a defined condition space. The main idea is based on that DRL can learn a policy for finding optimal solutions according to varying conditions and objectives. The method has been applied to two MCMO optimization problems. Firstly, as a benchmark problem, the Kursawe test problem has been newly modified to an MCMO optimization problem. Secondly, an airfoil shape optimization problem has been dealt with as a practical engineering application. The present MCMO optimization method shows outstanding ability in finding high-resolution Pareto front within the entire condition space including nonlinear and nonconvex parts.

Two additional analyses have been conducted to show its exclusive capability. Firstly, a computational experiment based on the modified Kursawe test problem has been carried out to identify the effectiveness of MC optimization. Compared with multiple operations of SC optimization for multiple conditions, the number of function evaluations required to find Pareto front is significantly reduced. This efficient optimization is enabled by learning correlations between conditions and optimal solutions. Secondly, the necessity for MC optimization has been confirmed through an analysis of aerodynamic performance of airfoils with optimally designed shapes. It has been found that an optimal solution at a specific condition may not be valid at the nearby conditions, resulting in significant deterioration of target performance. Thus, it is essential to cover the entire condition space for problems with nonlinear characteristics such as airfoil shape optimization, which is possible through the proposed MC optimization method.



$0^\circ \leq \alpha < 0.15^\circ$

$4.95^\circ \leq \alpha < 5.1^\circ$

$9.9^\circ \leq \alpha < 10.05^\circ$

$14.85^\circ \leq \alpha \leq 15^\circ$

Fig. 11. Optimal design parameters as a function of Re_c and w_1 at four α subspaces.

The proposed method shows outstanding capability in optimization problems where conditions and objectives are not fixed. A representative example is shape optimization in fluid mechanics. The proposed method is not limited to shape optimization and can be applied to general MCMO optimization problems.

In addition, the proposed MCMO optimization method is expected to be useful for a reduced order modeling framework in two ways. The first way is to use a reduced order model to perform the function evaluations required during the optimization process. In situations where generating high-fidelity data is expensive, a reduced order model can be helpful. It is also possible to initially utilize the data generated by a reduced order model and fine-tune the network with additional details through transfer learning. The second way is to develop a reduced order model using the proposed MCMO optimization method. In the proposed method, the critic network learns to model the reward for a given state and action. The trained critic network is expected to perform the role of a reduced order model.

CRediT authorship contribution statement

Sejin Kim: Conceptualization, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Innyoung Kim:** Conceptualization, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Donghyun You:** Conceptualization, Funding acquisition, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work was supported by the National Research Foundation of Korea (NRF) under the Grant Number NRF-2021R1A2C2092146 and the Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TB1703-51.

Appendix A. Weighted Chebyshev method and Pareto optimality

The weighted Chebyshev method is based on the weighted metrics method [37] which is defined as follows:

$$\min_{\boldsymbol{x}} L_p(\boldsymbol{x}) = \min_{\boldsymbol{x}} \Big(\sum_i (w_i |f_i(\boldsymbol{x}) - f_i^*|^p)^{1/p}\Big), \tag{A.1}$$

where $|\boldsymbol{w}| = 1$ and $w_i \geq 0$. f_i^* is a utopia value which is defined as $f_i^* = \inf\{f_i(\boldsymbol{x})\} - \tau_i$, where τ_i is a small positive value. L_p metrics indicate the distance between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}^*$, which differs depending on p $(0 < p \leq \infty)$. When $p = \infty$, the metric L_∞ becomes the Chebyshev scalarizing function $f_{Chebyshev}$, and the weighted metrics method is called the weighted Chebyshev method.

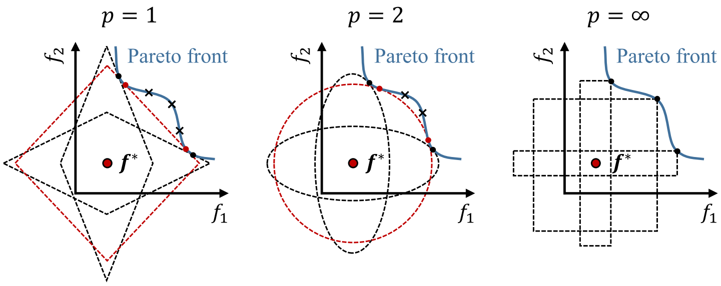


Fig. A.12. Schematic illustrations of the weighted metrics method with different p in the plane of two objectives f_1 and f_2 where $\boldsymbol{f}^*$ is an utopia point. ●, obtainable points; ●, marginal obtainable points; ×, unobtainable points.

Fig. A.12 shows schematics of optimization processes using the weighted metrics method with different p. Blue lines show Pareto front that satisfies Pareto optimality. When $p = 1$, a rhombus centered on $\boldsymbol{f}^*$ is formed, where its ratio of the width to the height is determined for given $\boldsymbol{w}$. The objective of the optimization is to minimize L_1 that is proportional to the rhombus size, as illustrated by the black rhombuses in Fig. A.12. Note that all Pareto front cannot be obtained since it is restricted to the rhombus shape. The red rhombus shows a marginal situation, and the black crosses show unobtainable points. When $p = 2$, an ellipse is formed and the optimization is performed in a similar manner. Although the obtainable part increases compared with $p = 1$, unobtainable points still exist. When $p = \infty$, a rectangle is formed and there is no marginal situation. Thus, the weighted Chebyshev method guarantees that all Pareto front can be obtained while it is not guaranteed for $0 < p < \infty$.

References

[1] E.K. Chong, S.H. Zak, An Introduction to Optimization, John Wiley & Sons, 2004.

[2] J. Semmler, L. Pflug, M. Stingl, G. Leugering, Shape Optimization in Electromagnetic Applications, Springer International Publishing, 2015, pp. 251–269.

[3] D. Taylor, J.-H. Dirks, Shape optimization in exoskeletons and endoskeletons: a biomechanics analysis, J. R. Soc. Interface 9 (77) (2012) 3480–3489, https://doi.org/10.1098/rsif.2012.0567.

[4] J. Park, A. Sutradhar, J.J. Shah, G.H. Paulino, Design of complex bone internal structure using topology optimization with perimeter control, Comput. Biol. Med. 94 (2018) 74–84, https://doi.org/10.1016/j.compbiomed.2018.01.001.

[5] B. Mohammadi, O. Pironneau, Shape optimization in fluid mechanics, Annu. Rev. Fluid Mech. 36 (1) (2004) 255–279, https://doi.org/10.1146/annurev.fluid.36.050802.121926.

[6] G. Droandi, G. Gibertini, Aerodynamic blade design with multi-objective optimization for a tiltrotor aircraft, Aircr. Eng. Aerosp. Technol., Int. J. 87 (1) (2015) 19–29, https://doi.org/10.1108/AEAT-01-2013-0005.

[7] D. Peri, M. Rossetti, E.F. Campana, Design optimization of ship hulls via CFD techniques, J. Ship Res. 45 (02) (2001) 140–149, https://doi.org/10.5957/jsr.2001.45.2.140.

[8] S. Percival, D. Hendrix, F. Noblesse, Hydrodynamic optimization of ship hull forms, Appl. Ocean Res. 23 (6) (2001) 337–355, https://doi.org/10.1016/S0141-1187(02)00002-0.

[9] S.-H. Yun, Y.-C. Ku, J.-H. Rho, D.-H. Lee, Application of Function Based Design Method to Automobile Aerodynamic Shape Optimization, Multidisciplinary Analysis Optimization Conferences, American Institute of Aeronautics and Astronautics, 2008.

[10] W. Xudong, W.Z. Shen, W.J. Zhu, J.N. Sørensen, C. Jin, Shape optimization of wind turbine blades, Wind Energy 12 (8) (2009) 781–803, https://doi.org/10.1002/we.335.

[11] D. Bertetta, S. Brizzolara, S. Gaggero, M. Viviani, L. Savio, CPP propeller cavitation and noise optimization at different pitches with panel code and validation by cavitation tunnel measurements, Ocean Eng. 53 (2012) 177–195, https://doi.org/10.1016/j.oceaneng.2012.06.026.

[12] S. Vasista, O. Mierheim, M. Kintscher, Morphing Structures, Applications of, Springer, Berlin Heidelberg, 2019, pp. 1–13.

[13] C.G. Diaconu, P.M. Weaver, F. Mattioni, Concepts for morphing airfoil sections using bi-stable laminated composite structures, Thin-Walled Struct. 46 (6) (2008) 689–701, https://doi.org/10.1016/j.tws.2007.11.002.

[14] S. Barbarino, O. Bilgen, R.M. Ajaj, M.I. Friswell, D.J. Inman, A review of morphing aircraft, J. Intell. Mater. Syst. Struct. 22 (9) (2011) 823–877, https://doi.org/10.1177/1045389X11414084.

[15] R.M. Ajaj, C.S. Beaverstock, M.I. Friswell, Morphing aircraft: the need for a new design philosophy, Aerosp. Sci. Technol. 49 (2016) 154–166, https://doi.org/10.1016/j.ast.2015.11.039.

[16] N. Garg, G.K.W. Kenway, Z. Lyu, J.R.R.A. Martins, Y.L. Young, High-fidelity hydrodynamic shape optimization of a 3-D hydrofoil, J. Ship Res. 59 (04) (2015) 209–226, https://doi.org/10.5957/jsr.2015.59.4.209.

[17] M. Sacher, M. Durand, É. Berrini, F. Hauville, R. Duvigneau, O. Le Maître, J.-A. Astolfi, Flexible hydrofoil optimization for the 35th America's Cup with constrained EGO method, Ocean Eng. 157 (2018) 62–72, https://doi.org/10.1016/j.oceaneng.2018.03.047.

[18] F. Chen, L. Liu, X. Lan, Q. Li, J. Leng, Y. Liu, The study on the morphing composite propeller for marine vehicle. Part I: design and numerical analysis, Compos. Struct. 168 (2017) 746–757, https://doi.org/10.1016/j.compstruct.2017.02.072.

[19] X. Lachenal, S. Daynes, P.M. Weaver, Review of morphing concepts and materials for wind turbine blade applications, Wind Energy 16 (2) (2013) 283–307, https://doi.org/10.1002/we.531.

[20] X. Shen, J.-G. Chen, X.-C. Zhu, P.-Y. Liu, Z.-H. Du, Multi-objective optimization of wind turbine blades using lifting surface method, Energy 90 (2015) 1111–1121, https://doi.org/10.1016/j.energy.2015.06.062.

[21] Y. Li, K. Wei, W. Yang, Q. Wang, Improving wind turbine blade based on multi-objective particle swarm optimization, Renew. Energy 161 (2020) 525–542, https://doi.org/10.1016/j.renene.2020.07.067.

[22] M. Secanell, A. Suleman, P. Gamboa, Design of a morphing airfoil using aerodynamic shape optimization, AIAA J. 44 (7) (2006) 1550–1562, https://doi.org/10.2514/1.18109.

[23] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248, https://doi.org/10.1162/evco.1994.2.3.221.

[24] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, https://doi.org/10.1109/4235.996017.

[25] C. Coello Coello, M. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02, vol. 2, 2002, pp. 1051–1056 (Cat. No. 02TH8600).

[26] K. Miettinen, M.M. Mäkelä, On scalarizing functions in multiobjective optimization, OR Spektrum 24 (2) (2002) 193–213, https://doi.org/10.1007/s00291-001-0092-9.

[27] Z. Lyu, J.R.R.A. Martins, Aerodynamic shape optimization of an adaptive morphing trailing-edge wing, J. Aircr. 52 (6) (2015) 1951–1970, https://doi.org/10.2514/1.C033116.

[28] X. Yan, J. Zhu, M. Kuang, X. Wang, Aerodynamic shape optimization using a novel optimizer based on machine learning techniques, Aerosp. Sci. Technol. 86 (2019) 826–835, https://doi.org/10.1016/j.ast.2019.02.003.

[29] J. Li, M. Zhang, J.R.R.A. Martins, C. Shu, Efficient aerodynamic shape optimization with deep-learning-based geometric filtering, AIAA J. 58 (10) (2020) 4243–4259, https://doi.org/10.2514/1.J059254.

[30] J. Rabault, F. Ren, W. Zhang, H. Tang, H. Xu, Deep reinforcement learning in fluid mechanics: a promising method for both active flow control and shape optimization, J. Hydrodyn. 32 (2) (2020) 234–246, https://doi.org/10.1007/s42241-020-0028-y.

[31] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, E. Hachem, Direct shape optimization through deep reinforcement learning, J. Comput. Phys. 428 (2021) 110080, https://doi.org/10.1016/j.jcp.2020.110080.

[32] S. Qin, S. Wang, L. Wang, C. Wang, G. Sun, Y. Zhong, Multi-objective optimization of cascade blade profile based on reinforcement learning, Appl. Sci. 11 (1) (2021) 106, https://doi.org/10.3390/app11010106.

[33] R. Li, Y. Zhang, H. Chen, Learning the aerodynamic design of supercritical airfoils through deep reinforcement learning, AIAA J. 59 (10) (2021) 3988–4001, https://doi.org/10.2514/1.J060189.

[34] F. Kursawe, A variant of evolution strategies for vector optimization, in: Parallel Problem Solving from Nature, Springer, Berlin Heidelberg, 1991, pp. 193–197.

[35] G. Chen, K.J. Fidkowski, Discretization error control for constrained aerodynamic shape optimization, J. Comput. Phys. 387 (2019) 163–185, https://doi.org/10.1016/j.jcp.2019.02.038.

[36] E. Gillebaart, R. De Breuker, Low-fidelity 2D isogeometric aeroelastic analysis and optimization method with application to a morphing airfoil, Comput. Methods Appl. Mech. Eng. 305 (2016) 512–536, https://doi.org/10.1016/j.cma.2016.03.014.

[37] K. Miettinen, Nonlinear Multiobjective Optimization, vol. 12, Springer Science & Business Media, 2012.

[38] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, https://doi.org/10.1109/TEVC.2007.892759.

[39] Y.-Y. Tan, Y.-C. Jiao, H. Li, X.-K. Wang, MOEA/D+ uniform design: a new version of MOEA/D for optimization problems with many objectives, Comput. Oper. Res. 40 (6) (2013) 1648–1660, https://doi.org/10.1016/j.cor.2012.01.001.

[40] K. Van Moffaert, M.M. Drugan, A. Nowé, Scalarized multi-objective reinforcement learning: novel design techniques, in: 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), pp. 191–199, https://doi.org/10.1109/ADPRL.2013.6615007.

[41] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[42] R. Bellman, Dynamic programming, Science 153 (3731) (1966) 34–37, https://doi.org/10.1126/science.153.3731.34.

[43] J. Viquerat, P. Meliga, E. Hachem, A review on deep reinforcement learning for fluid mechanics: an update, arXiv:2107.12206, 2021.

[44] V.R. Konda, J.N. Tsitsiklis, Actor-critic algorithms, in: Advances in Neural Information Processing Systems, 2000, pp. 1008–1014.

[45] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the 30th International Conference on Machine Learning, vol. 30, Citeseer, 2013, p. 3.

[46] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv:1509.02971, 2019.

[47] S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: Proceedings of the 35th International Conference on Machine Learning, vol. 80, PMLR, 2018, pp. 1587–1596.

[48] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, arXiv:1412.6980, 2017.

[49] E. Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, vol. 63, Citeseer, 1999.

[50] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: theoretical foundations and practical implications, Theor. Comput. Sci. 425 (2012) 75–103, https://doi.org/10.1016/j.tcs.2011.03.012.

[51] W.J. Lim, A.B. Jambek, S.C. Neoh, Kursawe and ZDT functions optimization using hybrid micro genetic algorithm (HMGA), Soft Comput. 19 (12) (2015) 3571–3580, https://doi.org/10.1007/s00500-015-1767-5.

[52] C.J. Tan, C.P. Lim, Y.-N. Cheah, A modified micro genetic algorithm for undertaking multi-objective optimization problems, J. Intell. Fuzzy Syst. 24 (2013) 483–495, https://doi.org/10.3233/IFS-2012-0568.

[53] M.-F. Leung, S.-C. Ng, C.-C. Cheung, A.K. Lui, A new strategy for finding good local guides in MOPSO, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1990–1997.

[54] Y. Naranjani, C. Hernández, F.-R. Xiong, O. Schütze, J.-Q. Sun, A hybrid method of evolutionary algorithm and simple cell mapping for multi-objective optimization problems, Int. J. Dyn. Control 5 (3) (2017) 570–582, https://doi.org/10.1007/s40435-016-0250-1.

[55] R. Mukesh, K. Lingadurai, U. Selvakumar, Airfoil shape optimization using non-traditional optimization technique and its validation, J. King Saud Univ., Eng. Sci. 26 (2) (2014) 191–197, https://doi.org/10.1016/j.jksues.2013.04.003.

[56] A.F.P. Ribeiro, A.M. Awruch, H.M. Gomes, An airfoil optimization technique for wind turbines, Appl. Math. Model. 36 (10) (2012) 4898–4907, https://doi.org/10.1016/j.apm.2011.12.026.

[57] S. Zhang, H. Li, W. Jia, D. Xi, Multi-objective optimization design for airfoils with high lift-to-drag ratio based on geometric feature control, IOP Conf. Ser. Earth Environ. Sci. 227 (2019) 032014, https://doi.org/10.1088/1755-1315/227/3/032014.

[58] L. Huyse, S.L. Padula, R.M. Lewis, W. Li, Probabilistic approach to free-form airfoil shape optimization under uncertainty, AIAA J. 40 (9) (2002) 1764–1772, https://doi.org/10.2514/2.1881.

[59] L.M. Milne-Thomson, Theoretical Aerodynamics, Courier Corporation, 1973.

[60] P. Puorger, D. Dessi, F. Mastroddi, Preliminary design of an amphibious aircraft by the multidisciplinary design optimization approach, in: 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 2007, p. 1924.

[61] M. Berci, V.V. Toropov, R.W. Hewson, P.H. Gaskell, Multidisciplinary multifidelity optimisation of a flexible wing aerofoil with reference to a small UAV, Struct. Multidiscip. Optim. 50 (4) (2014) 683–699, https://doi.org/10.1007/s00158-014-1066-2.

[62] M. Drela, XFOIL: an analysis and design system for low Reynolds number airfoils, in: Low Reynolds Number Aerodynamics, Springer, Berlin Heidelberg, 1989, pp. 1–12.

[63] T.H. Hansen, Airfoil optimization for wind turbine application, Wind Energy 21 (7) (2018) 502–514, https://doi.org/10.1002/we.2174.

[64] S. Zhang, H. Li, A.A. Abbasi, Design methodology using characteristic parameters control for low Reynolds number airfoils, Aerosp. Sci. Technol. 86 (2019) 143–152, https://doi.org/10.1016/j.ast.2019.01.003.

[65] K.R. Ram, S.P. Lal, M.R. Ahmed, Design and optimization of airfoils and a 20 kW wind turbine using multi-objective genetic algorithm and HARP_Opt code, Renew. Energy 144 (2019) 56–67, https://doi.org/10.1016/j.renene.2018.08.040.

[66] P.B.S. Lissaman, Low-Reynolds-number airfoils, Annu. Rev. Fluid Mech. 15 (1) (1983) 223–239, https://doi.org/10.1146/annurev.fl.15.010183.001255.

[67] H.P. Buckley, B.Y. Zhou, D.W. Zingg, Airfoil optimization using practical aerodynamic design requirements, J. Aircr. 47 (5) (2010) 1707–1719, https://doi.org/10.2514/1.C000256.

[68] M. Nemec, D.W. Zingg, T.H. Pulliam, Multipoint and multi-objective aerodynamic shape optimization, AIAA J. 42 (6) (2004) 1057–1065, https://doi.org/10.2514/1.10415.

[69] H. Lim, H. Kim, Multi-objective airfoil shape optimization using an adaptive hybrid evolutionary algorithm, Aerosp. Sci. Technol. 87 (2019) 141–153, https://doi.org/10.1016/j.ast.2019.02.016.

[70] P. Della Vecchia, E. Daniele, E. D'Amato, An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm, Aerosp. Sci. Technol. 32 (1) (2014) 103–110, https://doi.org/10.1016/j.ast.2013.11.006.