ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign

Data-Driven Morphological Exploration and Shape Optimization for Turbulent Pipe Systems

Kazuya Urata
Department of Mechanical Engineering,
The University of Osaka,
Osaka 565-0871, Japan
e-mail: urata@syd.mech.eng.osaka-u.ac.jp

Ryo Tsumoto
Department of Mechanical Engineering,
The University of Osaka,
Osaka 565-0871, Japan
e-mail: tsumoto@syd.mech.eng.osaka-u.ac.jp

Kentaro Yaji[1]
Department of Mechanical Engineering,
The University of Osaka,
Osaka 565-0871, Japan
e-mail: yaji@mech.eng.osaka-u.ac.jp

Kikuo Fujita
Department of Mechanical Engineering,
The University of Osaka,
Osaka 565-0871, Japan
e-mail: fujita@mech.eng.osaka-u.ac.jp

Optimal arrangements of turbulent pipe systems strongly depend on branch patterns, and turbulence fields typically cause involved multimodality in the solution space. These features hinder gradient-based structural optimization frameworks from finding promising solutions for turbulent pipe systems. In this article, we propose a multi-stage framework that integrates data-driven morphological exploration and evolutionary shape optimization to address the challenges posed by the complexity of turbulent pipe systems. Our framework begins with data-driven morphological exploration, aiming to find promising morphologies. It results in the shapes for selecting a reasonable number of candidates for the next shape refinement stage. Herein, we employ data-driven topology design, a gradient-free, and multi-objective optimization methodology incorporating a deep generative model and the concept of evolutionary algorithms to generate promising arrangements. Subsequently, a deep clustering strategy extracts representative shapes. The final stage involves refining these shapes through shape optimization using a genetic algorithm. Applying the framework to a two-dimensional turbulent pipe system with a minimax objective shows its effectiveness in delivering high-performance solutions for the turbulent flow optimization problem with branching. [DOI: 10.1115/1.4068984]

Keywords: data-driven topology design, turbulence problem, deep clustering, shape optimization, data-driven design, design optimization, generative design, multi-objective optimization, structural optimization, topology optimization

1 Introduction

Pipe systems, whose representative example is flow distributors in air conditioners, are the equipment that divide the inflow into several paths and merge them into the outflow. Considering an optimization problem for designing such pipe systems, the solution space becomes complex because the branch pattern depends on a combination of multiple levels, the number of branches, and the number of paths after branching. Each branch pattern has its inherent optimal coordinates of branch points, and the optimum shape is determined based on each branch pattern and its branch points. Therefore, the solution space of an optimization problem for a pipe system is affected by a complexity derived from combinations of branching. Moreover, turbulence fields are the typical state in most fluids from the engineering viewpoint, so it is essential to consider turbulent flow fields when designing fluid devices practically. On the other hand, turbulence fields also typically cause multimodality, where multiple local optima exist in the solution space of their optimization problem. Hence, an optimization problem for turbulent pipe systems is strongly affected by involved multimodality derived from combinations of branching and turbulent flow fields. Conventional gradient-based structural optimization frameworks face challenges when applied to a design problem under turbulence depending on the degree of design freedom [1].

Gradient-free optimization, such as evolutionary algorithms (EAs), could effectively design shapes under turbulent flow fields. Many studies have shown that evolutionary shape optimization could be applied to turbulent design problems [2–5]. The solution space of a shape optimization problem is, however, limited since only boundary moving is possible during the optimization process in general. Optimization results via this framework strongly depend on the initial guess. It is impractical to apply this framework to all of the enormous variety of branch patterns and select a superior optimized design among them.

Topology optimization [6] has the highest degree of design freedom in structural optimization as it is capable of searching not only for arrangements in terms of material distribution but also for shapes. As an attractive feature, the issue of how to arrange initial guesses in topology optimization is not critical compared with shape optimization. In exchange for its high degree of design freedom, topology optimization might be impractical in finding promising solutions due to the multimodality, where multiple local optima exist in the solution space of the optimization

This paper is a revised version of Paper No. IDETC2024-143383 published in the ASME IDETC Design Automation Conference, Aug. 25–28, 2024.
[1]Corresponding author.
Contributed by Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received November 22, 2024; final manuscript received June 10, 2025; published online July 21, 2025. Assoc. Editor: Shikui Chen.

problem [7]. It tends to be strongly affected by the issue when dealing with complex problems such as turbulent pipe systems. This is because topology optimization is typically based on a gradient-based approach to realize its high degree of design freedom. Although there has been a part of progressive research on gradient-based topology optimization for turbulent problems [8–10], their scope of application is limited to relatively simple turbulence fields due to the complexity of the problems.

In view of the features of shape and topology optimization methods, it could be identified that a two-stage approach is promising: (1) topology optimization under tractable analysis models is used for generating an initial guess, and (2) the topology-optimized candidate is finalized by shape optimization. Several works have proved such a two-stage approach could be an effective method to design structures with high performance in several areas such as structural problems [11,12], fluid problems [13], acoustics [14], and micro electro mechanical systems [15]. Since applicable problems are limited due to the multimodality in a solution space of a topology optimization problem, it should be noted that the applicability of the two-stage approach is also limited to relatively simple problems. Gradient-free topology optimization is promising as the first stage to remove the gap over the second stage. Still, it typically needs to lose its advantage, namely, a high degree of design freedom, due to the curse of dimensionality [16]. Introducing a gradient-free topology optimization method that has the potential to avoid the curse of dimensionality is an effective strategy to fill the gap over the second stage in the two-stage approach.

Deep generative models [17], which can reduce the dimensionality of the design space, have been attracting significant attention. Several frameworks integrating a topology optimization method and a deep generative model on engineering design problems have been proposed to address the issues of the curse of dimensionality [18,19]. Guo et al. [20] have proposed pioneering work that introduces a variational autoencoder (VAE) [21], one of the representative deep generative models, in topology optimization framework to generate initial guess of topology optimization effectively. Oh et al. [22] have employed a generative adversarial network (GAN) [23] to generate promising material distributions from topologically optimized structures.

Based on the idea of a data-driven approach and EAs, Yamasaki et al. [24] proposed data-driven topology design (DDTD), which is a gradient-free and multi-objective topology optimization method. The concept of DDTD is that a dataset composed of design candidates, including various topologies, is updated during the optimization process [22]. In this method, a deep generative model, VAE [21], is used to correspond to the role of crossover, which is an operation to generate a new solution by two parent solutions, in the process of EAs. Deep generative models can extract features from high-dimensional data into a low-dimensional space and generate data that inherit their features by sampling in the aforementioned space. In other words, a high degree of design freedom can be handled with a small number of variables. Consequently, this framework has been successfully applied to a feasible methodology for the optimization problems that are hard to solve with gradient-based topology optimization, e.g., turbulent problems [25], minimax problems [26,27], and latent heat storage problems [28]. The two-stage approach discussed above is a natural extension for DDTD that can deal with involved problems for conventional topology optimization in the first stage.

In this article, we propose a multi-stage optimal design framework composed of DDTD and evolutionary shape optimization and apply it to a design problem for a turbulent pipe system. The main idea of the proposed framework is as follows. First, we generate various arrangements through DDTD under a bi-objective problem, namely, the pressure drop and uniformity of the pipe system. After getting promising arrangements in the sense of Pareto optima, we divide those arrangements into several clusters using deep clustering—variational deep embedding (VaDE) [29]—and acquire representative shapes from the clusters. These procedures are interpreted as *data-driven morphological exploration*, which enables exploring topologies as well as shapes by a data-driven approach with cyclopedic data and aims at selecting a reasonable number of candidates for the next shape refinement stage. In the finalizing stage, we refine the representative shapes from the previous step using evolutionary shape optimization. In this study, we use the Bezier curve to represent the shapes of the pipe system and optimize the control points by a real-coded genetic algorithm [30]. We demonstrate the efficacy of the proposed framework through a design problem for a two-dimensional pipe system under turbulence.

2 Framework

In this section, we describe the overview of the proposed framework shown in Fig. 1. This framework is built from two major compartments: data-driven morphological exploration and shape refinement. The detailed procedures of those two compartments are described in Sec. 4.

A two-stage approach, which utilizes topology optimization for generating an initial guess and shape optimization for shape refinement on the topology-optimized candidate, is promising. Despite the prospectivity of this approach, the applicability of the first

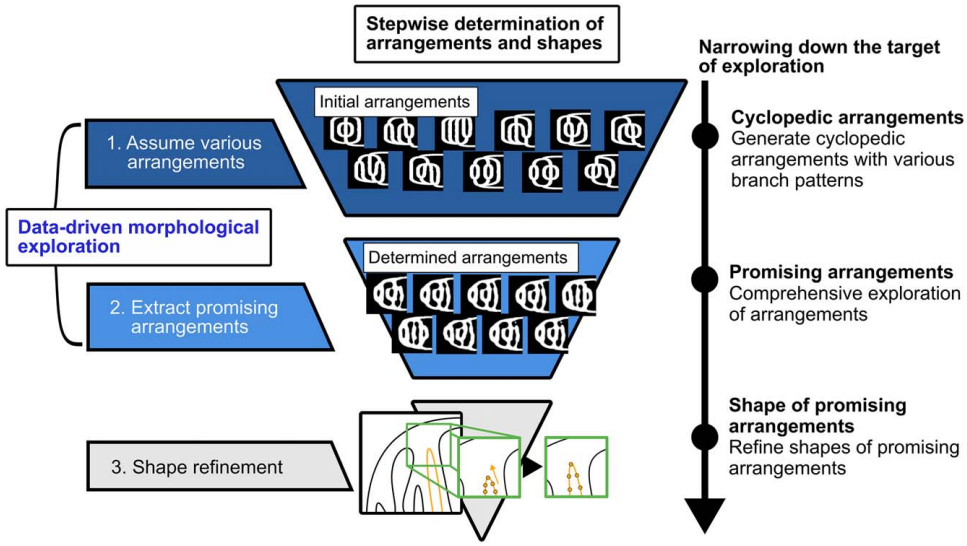

Fig. 1 Overview of proposed framework

stage, topology optimization, is limited. Regarding turbulent pipe systems, each branch pattern is expected to have its optimal branch points and shape. Moreover, turbulence fields cause a multimodality in the solution space of an optimization problem. Therefore, a structural optimization problem for turbulent pipe systems is complex. This complexity arose from a huge number of combinations derived from branching and turbulence narrows the range of applications of conventional topology optimization methods. For this reason, we propose a multi-stage optimal design framework to tackle problems with such complexities, e.g., an optimization problem for turbulent pipe systems.

First, we assume various arrangements, including a wide range of branch patterns, as initial solutions, unlike a conventional topology optimization method using such as a homogenization method and density method, which generally derives an optimized design from uniform material distribution. Then, promising arrangements are squeezed through a comprehensive exploration of arrangements by the scheme combining an evolutional strategy and generation of new data with features of the previous ones based on a deep generative model. After squeezing promising arrangements, we determine representative shapes that can show high performance from them. These procedures are interpreted as data-driven morphological exploration, which aims to explore topologies as well as shapes and determine several promising candidates for the next stage, shape refinement. Finally, we refine the boundaries of representative ones.

This framework enables dealing with complex design problems for which conventional structural optimization methods cannot find promising solutions by limiting the scope of exploration for the solution space strategically and incrementally.

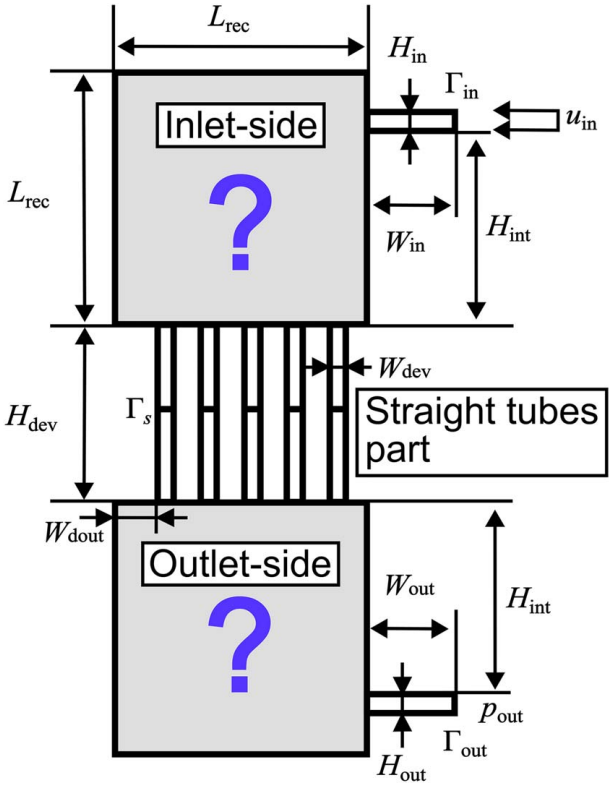

Fig. 2 Analysis domain

3 Problem Settings

This study deals with a two-dimensional design problem under turbulent flow fields with branching. The dimensions and boundary conditions are shown in Fig. 2. Design domains of this problem are assumed to be square areas between the inlet and five paths, and five paths and the outlet.

3.1 Governing Equations. First, we describe the governing equation and boundary conditions. We assume steady and incompressible turbulent flow fields in this problem. The flow fields are simulated by solving the Reynolds-averaged Navier–Stokes (RANS) equation with k–ε model. We applied the equation of continuity, RANS equation, and transport equations of turbulence kinetic energy and dissipation rate. The governing equations are written as follows:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla \cdot \left[-p\boldsymbol{I} + (\mu + \mu_T)(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)\right] \tag{2}$$

$$\rho(\boldsymbol{u} \cdot \nabla)k = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_k}\right)\nabla k\right] + P_k - \rho\varepsilon \tag{3}$$

$$\rho(\boldsymbol{u} \cdot \nabla)\varepsilon = \nabla \cdot \left[\left(\mu + \frac{\mu_T}{\sigma_\varepsilon}\right)\nabla \varepsilon\right] + C_{1\varepsilon}\frac{\varepsilon}{k}P_k - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \tag{4}$$

$$\mu_T = \rho C_\mu \frac{k^2}{\varepsilon} \tag{5}$$

$$P_k = \mu_T\left[\nabla \boldsymbol{u} : \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)\right] \tag{6}$$

where $\boldsymbol{u}(\boldsymbol{x})$ and $p(\boldsymbol{x})$ are time-averaged flow velocity vector and time-averaged pressure at $\boldsymbol{x}$ as position in analysis domains. ρ, μ, μ_T, and P_k are, respectively, the density, the viscosity coefficient, the coefficient of eddy viscosity, and the generation term of turbulence kinetic energy. σ_k, σ_ε, $C_{1\varepsilon}$, $C_{2\varepsilon}$, and C_μ are model constants. Uniform flowrate as u_{in} and p_{out} is applied at the inlet and outlet boundaries, and the wall function is applied at other boundaries as boundary conditions.

3.2 Formulation of Optimization Problem. Considering the pressure drop between the inlet and outlet, and the difference in the flowrate of five paths as performance indicators, objective functions of this problem are formulated as follows:

$$J_1 = p_{in} - p_{out} = \int_{\Gamma_{in}} p ds - \int_{\Gamma_{out}} p ds \tag{7}$$

$$J_2 = \max_e \left(\int_{\Gamma_s} \boldsymbol{n} \cdot \boldsymbol{u} ds\right) - \min_e \left(\int_{\Gamma_s} \boldsymbol{n} \cdot \boldsymbol{u} ds\right) \tag{8}$$

where Γ_{in}, Γ_{out}, Γ_s, and $\boldsymbol{n}$ are inlet boundary, outlet boundary, and boundaries in five paths, and a downward unit normal vector of Γ_s. The objective function J_1 corresponds to the pressure drop through the whole pipe system and is formulated as the boundary integral difference between the inlet static pressure p_{in} and the outlet static pressure p_{out}. The objective function J_2 corresponds to the difference in flowrate among five paths in the straight tubes part and is formulated as the difference between the maximum and minimum flow velocities of five paths. It should be noted that the proposed method can directly solve the minimax problem, which cannot be directly solved without any relaxation techniques such as the p-norm function [31] in the design problem for conventional topology optimization because of its indifferentiability.

4 Procedures

In this section, we divide the proposed framework into two steps and describe each of them in detail.

4.1 Data-Driven Morphological Exploration. First, we explore arrangements with superior performance using DDTD, which is a methodology that enables gradient-free and multi-objective optimization. In this method, candidate arrangements are updated iteratively on the basis of EA strategy and deep generative model. The schematic flowchart of this method is shown in Fig. 3, and each operation is described in the following. In this article, candidate arrangements are represented as grayscale bitmap images, and the top and bottom half of images stand for the inlet-side and outlet-side parts of the design domain, respectively as indicated in the top right part of Fig. 3. Every pixel in the image has a continuous value from 0 to 1, and flow paths are constructed by taking the contour on an arbitrary value.

4.1.1 Generation of Initial Data. Various arrangements, including a wide range of structures, are necessary to explore comprehensive solution space. In this study, we prepare various arrangements according to Fig. 4 and the following procedure:

(1) Determine the branch pattern corresponding to the inlet-side and outlet-side parts of the design domain from Fig. 5.
(2) Put the points at the end points of five paths and, inlet or outlet.
(3) Put the point at the first branch point. x coordinate of that point is randomly determined between x_{lower} and x_{upper}. y coordinate is set to y_1.
(4) Put the points at the second branch points. x coordinates of those points are set to the center of branch destinations. y coordinates are set to y_2.
(5) Connect those points above with the Bezier curves. The width of curves is set to w_{path}.
(6) Prepare another inverted data and combine them.

We suppose a branch pattern for five paths in one or two levels. Considering branching into two paths in the first level, four types of pattern are assumed: one of these paths branches into two paths and other branches into three, one branches three and other branches two, one remains one path and other branches four paths, and one branches four and one remains one path, as showed as patterns 1, 2, 6, and 7 in Fig. 5, respectively. Taking the case of three, four, and five branches in the first stage into account in the same way, a total of 15 different patterns can be assumed. Regarding the center of branch destinations in step 4, the term "the center of branch destinations" indicates the center of paths of the straight tubes part to be connected by the second level of the branch. Considering branching into two paths, and those paths are connected to the leftmost path and second leftmost path of the straight tubes part at the second level, the center of the branch destination is determined as the center of the leftmost path and second leftmost path of the straight tubes part.

4.1.2 Evaluation. The governing equations (Sec. 3.1) are solved using the finite element method and two objective functions (Sec. 3.2) are evaluated.

In the evaluation step in data-driven morphological exploration, the performance of the candidate solutions is evaluated by approximating the contour plots of the candidate solutions represented as bitmap images with a cubic spline curve and using it as the geometry. Moreover, to avoid small curvature in bitmap images, a smoothing technique is used [32].

4.1.3 Population Forming. According to objective values acquired in the previous step, superior candidates are selected according to elite strategy. Elite strategy in EAs is a rule that superior candidates are preserved for the next generation. This rule enables the prevention of a situation where superior candidates are extinguished and the obtaining of an optimized design with a relatively small number of iterations. In this study, we adopt the non-dominated sorting genetic algorithm II (NSGA-II) [33] as an operator for population forming. This algorithm affords candidates selection based on ranks of candidate solutions according to Pareto dominance relation, which is a superiority relation of each solution in the solution space.

4.1.4 Generative Model. A generative model is used to generate the next candidates from solutions formed by the process above. We adopt a VAE in this step. In order to generate various data that

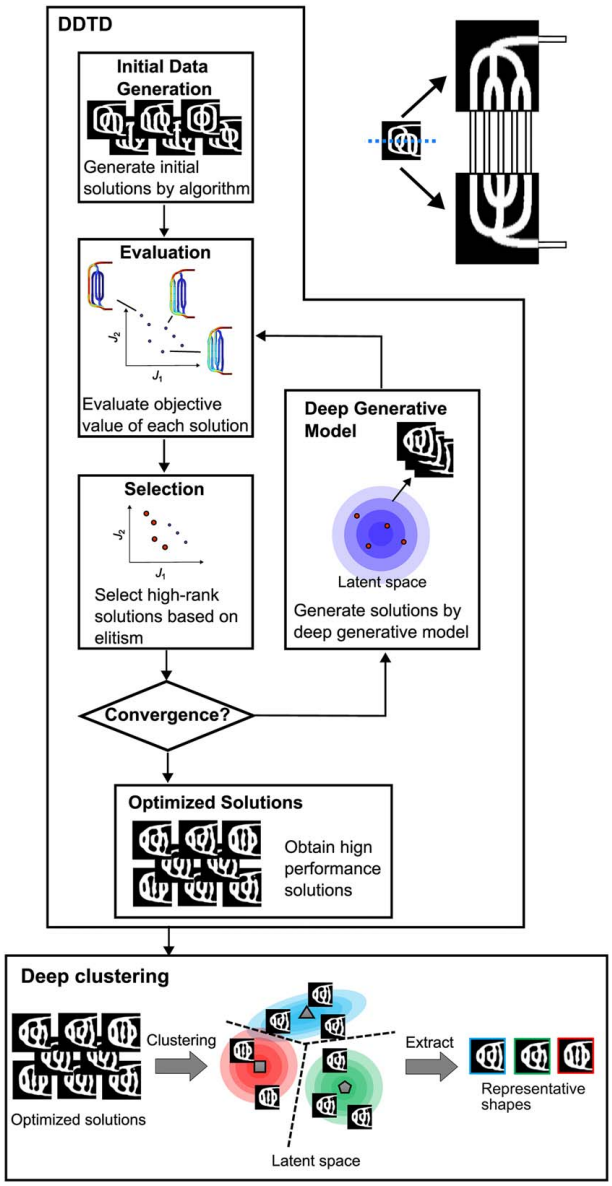

Fig. 3 Process of data-driven morphological exploration

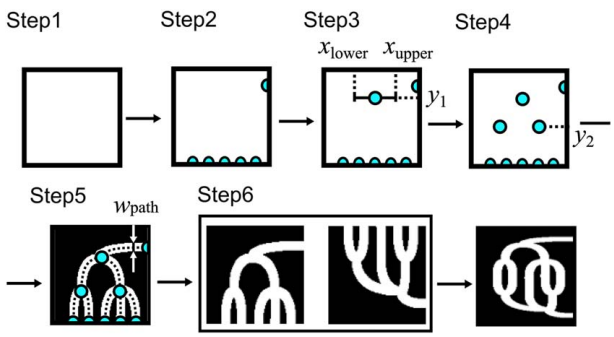

Fig. 4 Procedure for generation of initial data

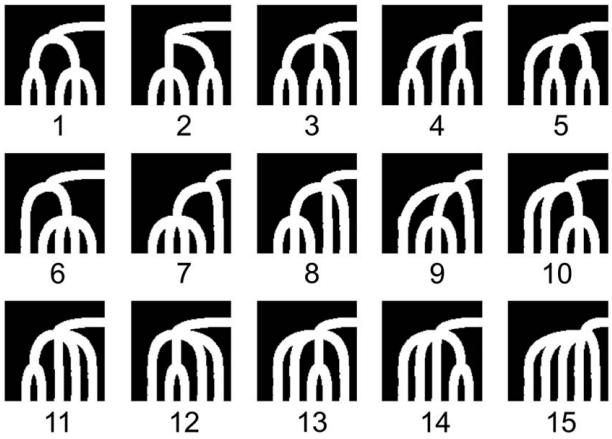

Fig. 5 Assumed branch patterns corresponding to the inlet-side and outlet-side parts of design domain

inherit features of candidates in the previous population by the generative model to proceed with the comprehensive exploration, the generative model needs to be stable and therefore the network of the model is desirable to be simple. GAN [23], one of the representative generative models could cause instability such as mode collapse, a phenomenon that causes low diversity of output data by a generator, in exchange for the capability of generating sharp images. Compared to GAN, VAE is less susceptible to such instability since the network architecture of VAE is relatively simple [34]; therefore, VAE is suitable for this method.

VAE is composed of two neural networks: an encoder and a decoder. It can be expected that data that inherit features from inputs will be generated since they can extract the information that represents the structures of inputs and reduce it to a low-dimensional manifold called the latent space. Here, we describe the architecture of VAE. Figure 6 shows the schematic diagram of VAE. The input layer has neurons corresponding to the number of input dimensions. This input layer is fully connected to the hidden layer. Then, this layer is also fully connected to the two layers comparable to the mean value vector $\boldsymbol{\mu}$ and the variance value vector $\boldsymbol{\sigma}$. The latent variable $\boldsymbol{z}$ is defined as follows:

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \boldsymbol{\varepsilon} \tag{9}$$

where $\circ$ is the element-wise product and $\boldsymbol{\varepsilon}$ is a random vector in accordance with the standard normal distribution. The layer of the latent variable $\boldsymbol{z}$ is fully connected to another hidden layer as well.

This architecture above is trained with the same dataset for inputs and outputs to construct the latent space using the following loss function, $L_{\mathrm{VAE}} : eqqL_{\mathrm{recon}} + L_{\mathrm{KL}}$, where L_{recon} is the reconstruction loss measured by the mean-squared error and L_{KL} is the

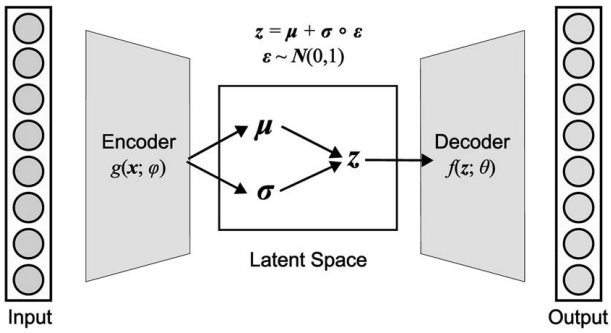

Fig. 6 Architecture of VAE

Kullback–Leibler(KL) divergence. L_{KL} is figured as follows:

$$L_{\mathrm{KL}} = -\frac{\varrho}{2}\sum_{i=1}^{N_{\mathrm{lt}}}\left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right) \tag{10}$$

where ϱ is the weight parameter controlling the impact of the KL divergence to regularize the latent space to the standard normal distribution, N_{lt} is the number of dimensions of the latent space, and μ_i and σ_i are respectively the ith components of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

It is expected that the important features of training data are extracted into the latent space because the dimensionality is reduced from the input and output layers into the low-dimensional latent space. Unlike autoencoders [35], a probabilistic sampling with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ enables continuous interpolation of input data in the latent space of VAE. Based on these characteristics, we can obtain material distributions from sampling in the latent space. VAE assumes a standard normal distribution as the prior distribution in the latent space, we generate material distributions from the sampling vectors comprised be uniformly distributed random numbers in $[-4, 4]$, which encompasses 99.7% of the data within $\pm 4\sigma$, for each latent variable. Therefore, data that are diverse and inherit the important features of training data are generated.

4.1.5 Extraction of Representative Shapes. Representative shapes of Pareto solutions produced by DDTD are extracted by applying a VaDE [29]. VaDE is an unsupervised generative and clustering model. The functionality of VaDE is explained as a further extension of the VAE. The architecture of VaDE is represented in the same manner as VAE shown in Fig. 6, and the latent variable $\boldsymbol{z}$ is also defined as Eq. (9).

Unlike VAE, VaDE assumes not a single Gaussian, but a mixture-of-Gaussian as the prior distribution in the latent space. Thus, a learned VaDE model can cluster samples based on the posterior of each Gaussian. Furthermore, since a VaDE is a generative model, representative data of each cluster can be generated by decoding the means of a mixture-of-Gaussian. Besides, VaDE can be applied to an extraction of the representative shapes from Pareto solutions represented by high-dimensional design variables due to the nature of deep learning.

4.2 Shape Refinement. Finally, representative shapes are finalized by evolutionary shape optimization. In this study, we apply a real-coded genetic algorithm, one of EAs since the number of design variables in this method is fewer than that in data-driven topology design by far due to the difference in the manner of structural representation. This method is gradient-free and multi-objective as well as DDTD.

4.2.1 Approximation by the Bezier Curve. Explicit geometry expressions of shapes are essential to perform shape optimization. However, representative shapes are represented as grayscale bitmap images and thus have no explicit geometry expressions of boundaries. According to the above, we replace representative shapes with the Bezier curves as follows:

(1) Prepare the contour plot of a boundary from a representative shape and choose a target curve from the contour.
(2) Prepare nth order Bezier curve $\boldsymbol{P}_n$. Here, coordinates of $\boldsymbol{p}_0$ and $\boldsymbol{p}_n$ accord with end points of original shapes.

$$\boldsymbol{P}_n = \sum_{i=0}^{n} {}_nC_i B_i \boldsymbol{p}_i \quad (i = 0, \ldots, n) \tag{11}$$

where ${}_nC_i$ is the binomial coefficient, ${}_nC_iB_i$ is the Bernstein polynomial, and $\boldsymbol{p}_i$ is a vector representing coordinates of ith control points of the Bezier curve.
(3) Perform fitting with coordinates of the midpoint and parameters of the Bezier curve as design variables. This is performed by minimizing the loss function f_{loss}: squared sum of distances between points on the contour plot and the

Bezier curve

$$f_{\text{loss}} = \sum_{j=1}^{m} \left\| \left(\boldsymbol{q}_{\text{target}_j} - \boldsymbol{q}_{t_j} \right) \right\|^2 \tag{12}$$

where $\boldsymbol{q}_{\text{target}_j}$ is jth vector representing points on the contour plot, $\boldsymbol{q}_{t_j}$ is jth vector representing points on the Bezier curve corresponding to parameter $\boldsymbol{t}$, and m is the number of points on those shapes.

(4) If the value of loss function f_{loss} is below the threshold value $C_{\text{threshold}}$, this curve is adopted as approximated curve. If the value is over the threshold value, prepare $(n+1)$th order Bezier curve and back to step 2. The threshold value $C_{\text{threshold}}$ is formulated as follows:

$$C_{\text{threshold}} = \frac{H_{\text{rec}}}{C_{\text{sq}}} \left(1 - \frac{1}{n-1} \right) \tag{13}$$

where C_{sq} is the constant corresponding to the size of the design domain and n is the number of order of the Bezier curve.

We obtain the approximated shape of the representative one by performing the above procedure for all curves of the representative shape.

An example of the approximation is shown in Fig. 7. The most left point represents a control point of the most left curve and the remaining points represent control points of the most right curve, respectively.

4.2.2 Initial Seeding. To proceed with the exploration based on a genetic algorithm, diverse shapes are necessary, but features of each cluster should be kept through the process because shapes with those features tend to show higher performance than others. Accordingly, we generate initial shapes as follows:

$$x_{\text{ori},k} - c_{\text{seed}} \leq x_{\text{gen},k} \leq x_{\text{ori},k} + c_{\text{seed}} \tag{14}$$

where $x_{\text{gen},k}$ is kth design variable generated in this step, $x_{\text{ori},k}$ is kth design variable obtained in Sec. 4.2.1, and c_{seed} is constant respectively.

4.2.3 Evaluation and Population Forming. As for this operation, see the section on evaluation and population forming for DDTD since this is the same as the one performed in the DDTD above (Secs. 4.1.2 and 4.1.3).

4.2.4 Selection. As opposed to DDTD, which generates new candidates from the latent space with features of the current population, new candidates are created by a crossover of two solutions from the current population in this method. In this study, we select them with the crowded tournament selection [33]. First, we choose a subset of solutions randomly from the current population according to the number of tournament size. Then, we obtain two superior solutions from them. The standard of superiority is as follows:

(1) Having a lower rank derived from NSGA-II,
(2) Having a higher crowding distance if they have the same rank.

Fig. 7 Example of approximation by the Bezier curve; left: representative shape, center: contour plot, and right: approximated curve with control points

4.2.5 Crossover. The crossover operation is performed with two solutions from the previous step. We adopt BLX-α [36], which is used in a real-coded genetic algorithm to generate new solutions. Each design variables of new solutions are randomly selected from the interval as follows:

$$[x^l_{\min} - \alpha d^l, x^l_{\max} + \alpha d^l] \tag{15}$$

where

$$x^l_{\min} = \min(x^l_{\text{parent},1}, x^l_{\text{parent},2}) \tag{16}$$

$$x^l_{\max} = \max(x^l_{\text{parent},1}, x^l_{\text{parent},2}) \tag{17}$$

$$d^l = |x^l_{\text{parent},1} - x^l_{\text{parent},2}| \tag{18}$$

α, $x^l_{\text{parent},1}$, and $x^l_{\text{parent},2}$ are constant, lth design variables selected in the previous step.

5 Results and Discussion

We now introduce numerical examples of a two-dimensional turbulent design problem with branching and demonstrate the effectiveness of the proposed framework. The simulation of flow fields using the finite element method and the evaluation of two objective functions employs COMSOL MULTIPHYSICS (version 6.0).

5.1 Design Settings. The problem that we address is shown in Fig. 2, where all the constants are dimensionless values. The values of those constants and boundary conditions are shown in Tables 1 and 2. Moreover, model constants in the governing equation are determined according to the standard k–ε model [37] as given in Table 3.

5.2 Results for Data-Driven Morphological Exploration

5.2.1 Generation of Initial Data. This step aims to generate multiple arrangements with various structures to explore a large solution space of this optimization problem through DDTD. As the dimensions of arrangements, we set x_{lower}, x_{upper}, y_1, y_2, and w_{path} are 45, 105, 80, 40, and 10, respectively according to the image size of 120×120 in Sec. 4.1.1.

Table 1 Parameter settings for the geometry of design domain and non-design domain

Symbol	Value
W_{in}	4
H_{in}	1
L_{rec}	12
H_{int}	9
W_{dev}	1
W_{dout}	2
H_{dev}	8
W_{out}	4
H_{out}	1

Table 2 Parameters of boundary conditions

Symbol	Value
u_{in}	1
p_{out}	0
Re	5×10^3

Table 3 Parameters of turbulence model

Symbol	Value
C_μ	0.09
σ_k	1
σ_ε	1.3
$C_{\varepsilon 1}$	1.44
$C_{\varepsilon 2}$	1.92

Figure 8 shows generated arrangements of the inlet-part and outlet-part of the design domain without the straight tubes part using the procedure shown in Fig. 4. In order to verify the diversity of them, we focus on the types of branch patterns. 225 patterns are assumed for the design domain in total since 15 patterns shown in Fig. 5, are assumed in the inlet-side and outlet-side parts of it respectively. The result shows that 178 branch patterns are generated, i.e., 80% of the total. This result confirmed that multiple arrangements with diversity are generated.

5.2.2 Improvement of Pareto Solutions. Initial arrangements from the previous step are the input to DDTD to produce the Pareto solutions. The convergence criterion is set to the maximum iteration number of 300. 512 elite solutions survive every iteration through the optimization process. The number of dimensions of the input layer and layer of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ in VAE is set to 14,400 and 8, respectively.

Figure 9 shows the values of objective functions on elite solutions at the initial, iteration of 100, 200, and optimized results. This result confirmed that the optimized solutions completely dominate the initial solutions. DDTD can work for an optimization problem for a turbulence pipe system with a minimax function.

Figure 10 shows arrangements of the inlet-part and outlet-part without the straight tubes part in the final results. We can classify them into three types of branch patterns based on the visual inspection and those patterns are seen in initial arrangements. This result emphasizes that the optimized arrangements generated by DDTD are greatly affected by the diversity of initial arrangements. Considering this result and input arrangements shown in Fig. 8, it can be expected that DDTD achieves the exploration for a large solution space of this optimization problem successfully due to the diversity of the input.

Figure 11 shows the comparison of velocity and pressure distribution between optimized and initial arrangements (hereinafter, those arrangements are called "Arrangement 1" to "Arrangement 6" from left to right). Two objective values of every arrangement are shown in Table 4. In this figure, six arrangements: three of them from optimized ones and the same number of them that have corresponding branch patterns from initial ones are selected. Arrangement 1 and 2 have the highest flow uniformity and the lowest pressure drop, respectively. Arrangement 3 has a relatively low pressure drop compared to other optimized ones.

The key factor to achieve high flow uniformity is the wall shape between the first and second paths from the right. The velocity distribution of Arrangement 3, 4, and 6 indicate that the flow from the inlet tends to rush into paths at the center or the left side, and it is

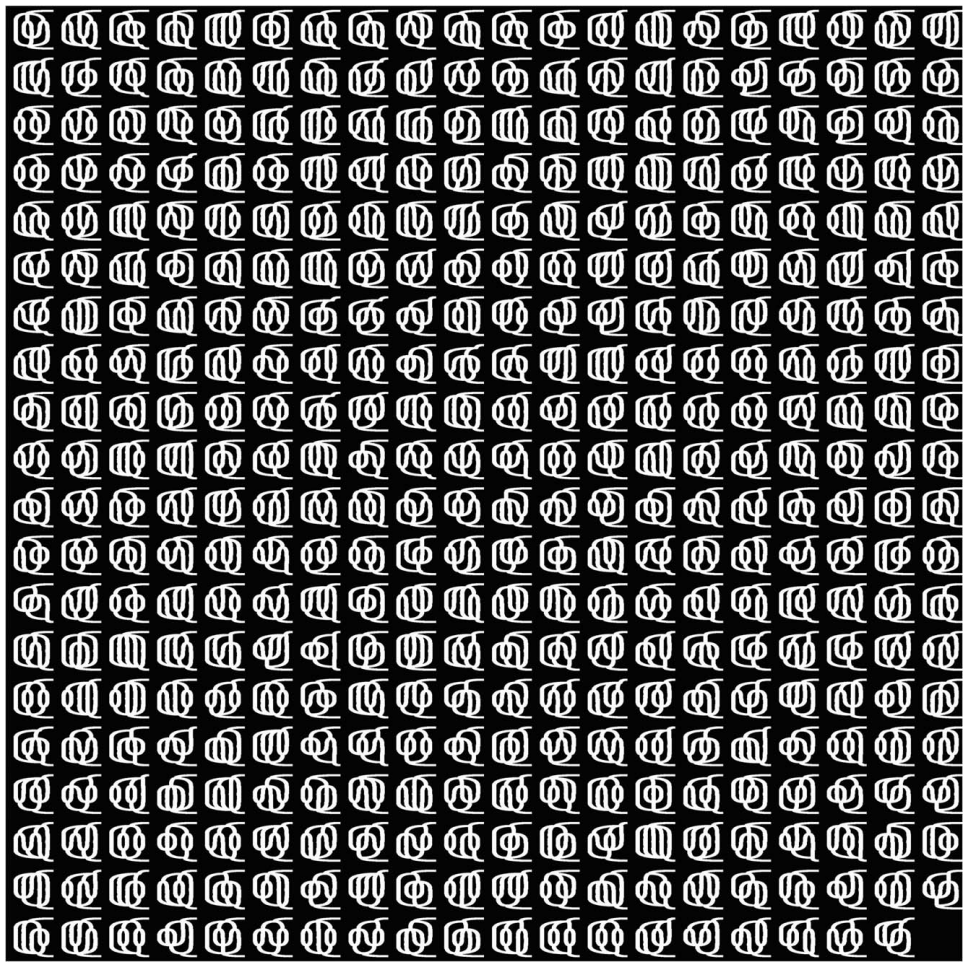

Fig. 8 Generated initial arrangements of inlet-side and outlet-side parts of design domain without straight tubes part

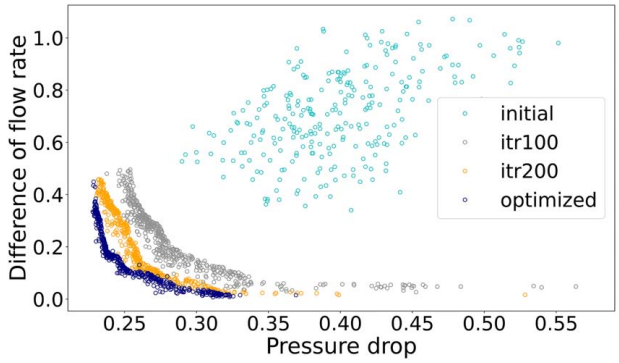

Fig. 9 Improvement of Pareto front by DDTD

difficult to direct it to paths on the right side. On the other hand, only Arrangement 1 intercepts inflow and directs it into the rightmost path by such a wall shape to achieve high flow uniformity.

From the perspective of the pressure drop, simplicity of the flow is the key factor for high performance. The flow in Arrangement 2 and 3 is uncomplicated in comparison with the sinuous flow in Arrangement 4, 5, and 6 caused by undulating paths. Sinuous flow causes an increase in the pressure drop since that in a bent flow is evaluated with its curvature. Besides, flexibility on the thickness of paths is one of the major reasons for decreasing pressure drop. The major factor of pressure drop in a straight flow path is the diameter of the paths. Low pressure drop is achieved by broadening paths in the process.

5.2.3 Results for Extraction of Representative Shapes. A VaDE model is trained by 512 arrangements shown in Fig. 10 to extract representative shapes. The learned VaDE compresses each solution represented as a 64×64 grayscale bitmap image into the five-dimensional latent space and simultaneously clusters them into five groups. This study sets the latent dimension to five based on the reconstruction loss of the learned VaDE model and the number of clusters to five experimentally.

Figure 12 shows representative shapes of five clusters generated by decoding the center of each cluster in the latent space. Intriguingly, representative shapes from Cluster 1, 2, and 5 are classified as the other cluster but they look like the same branch pattern visually. This result indicates that promising arrangements derived from the DDTD are clustered without relying on the visual inspection and representative shapes of each cluster are extracted.

5.3 Results for Shape Refinement. Finally, we perform shape optimization using a real-coded genetic algorithm to refine five representative shapes extracted in the previous step. We set m in Eq. (12) and C_{sq} in Eq. (13) to 9 and H_{rec}. The convergence criterion is set to the maximum iteration number of 150. 256 elite solutions survive every iteration through this optimization process. The α for crossover, BLX-α, is set to 0.2. The tournament size is set to 5.

Figure 13(*a*) shows the comparison of objective values of optimized solutions between DDTD and shape optimization. This result confirmed that the optimized solutions by the shape refinement overtake those by DDTD in terms of objective values. Additionally, to elucidate the effectiveness of data-driven morphological exploration, we apply shape optimization to the input of DDTD shown in Fig. 8

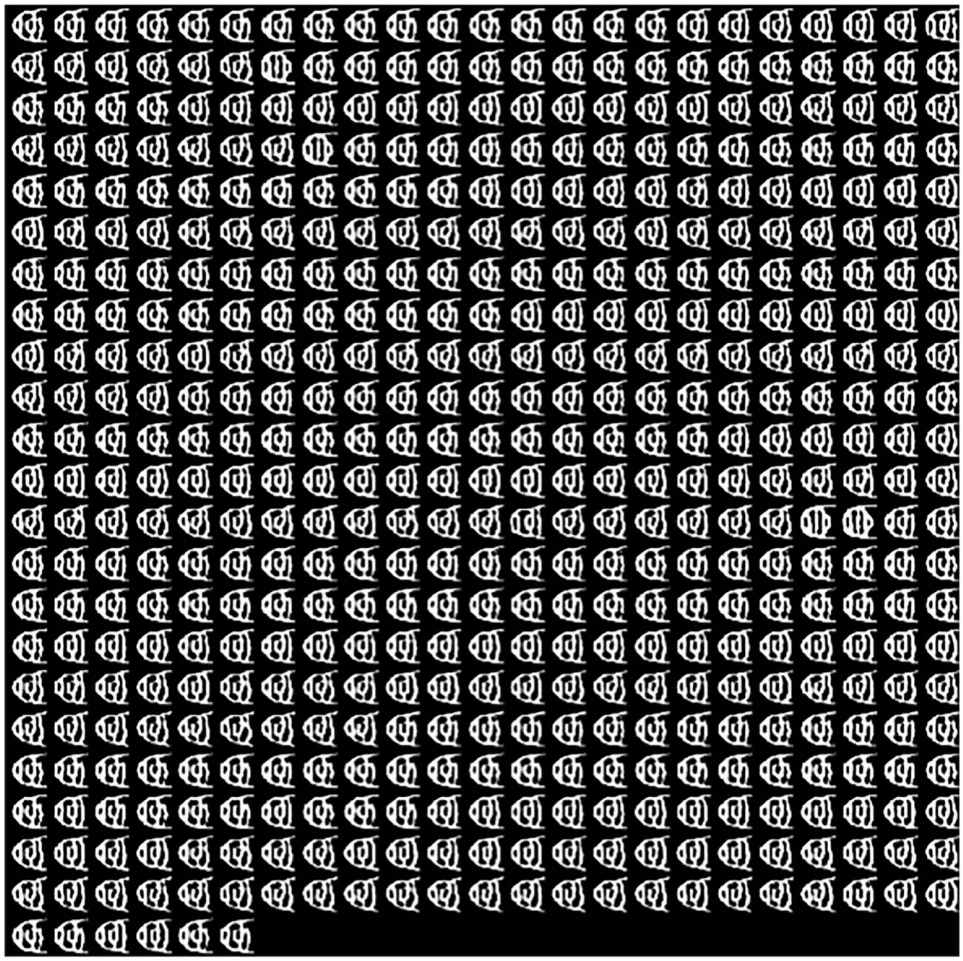

Fig. 10 Optimized arrangements of inlet-side and outlet-side parts of design domain without straight tubes part by DDTD

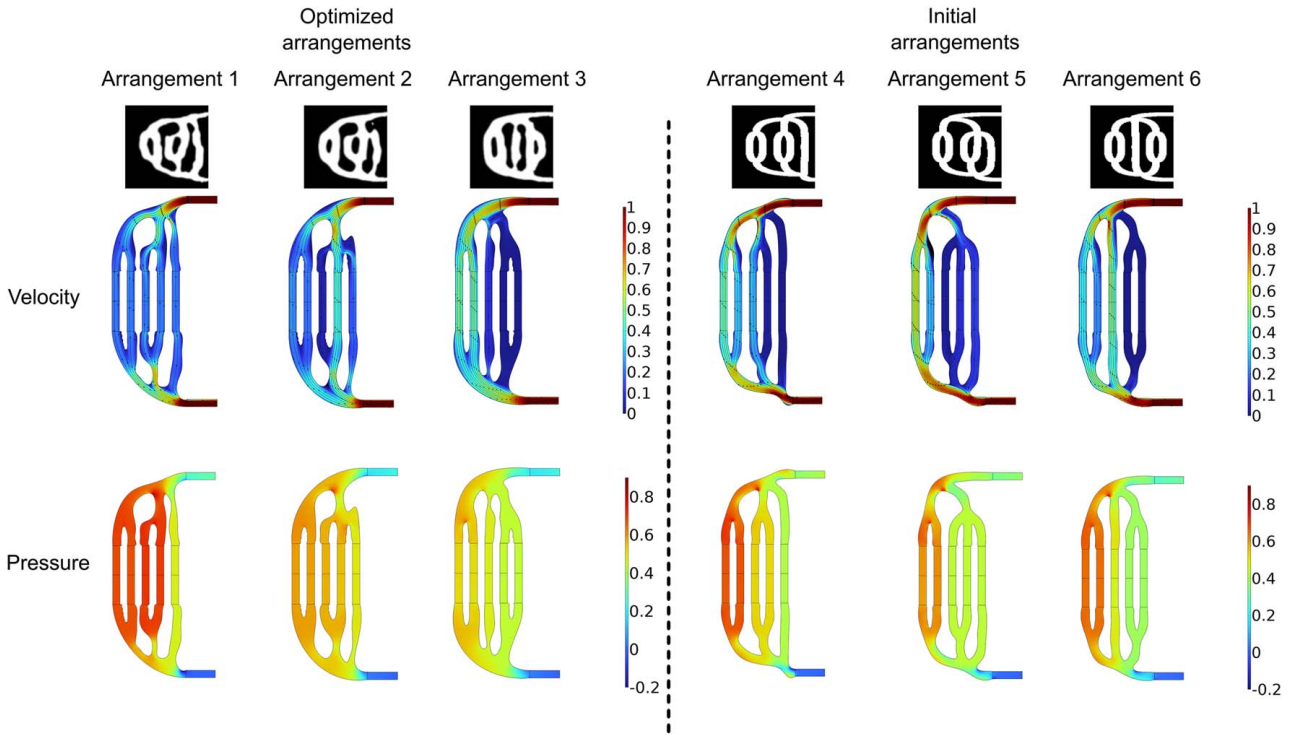

Fig. 11 Comparison of velocity and pressure distribution between optimized and initial arrangements for DDTD

Table 4 Objective values for optimized and initial arrangements for DDTD

	Arrangement 1	Arrangement 2	Arrangement 3	Arrangement 4	Arrangement 5	Arrangement 6
J_1	0.3115	0.2279	0.2295	0.4137	0.3795	0.3187
J_2	0.007386	0.3349	0.4297	0.4488	0.5824	0.4671

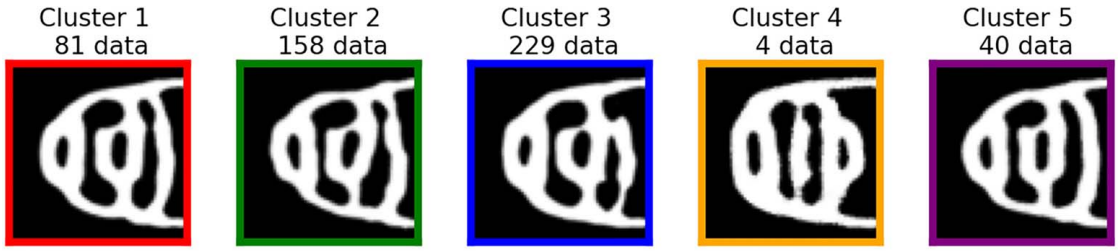

Fig. 12 Representative shapes by VaDE

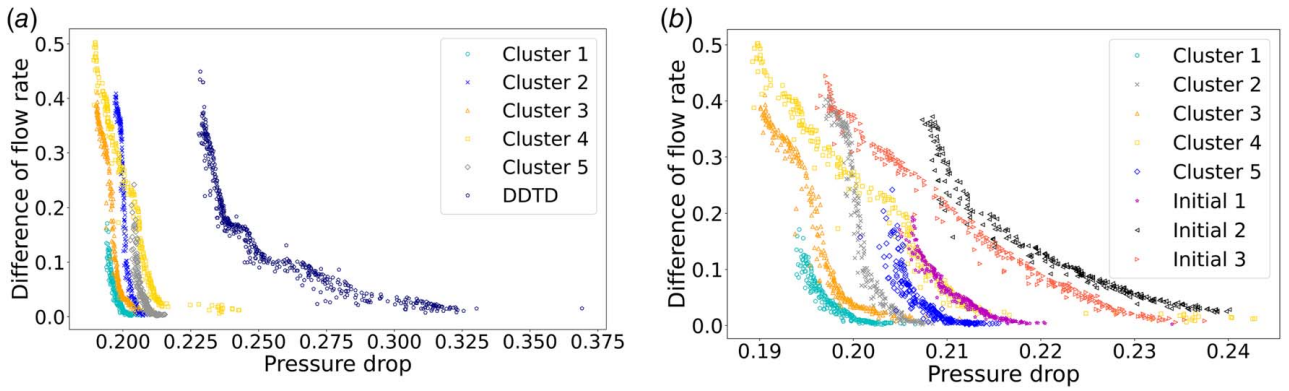

Fig. 13 Comparison of Pareto fronts by DDTD and shape optimization: (*a*) optimized solutions by DDTD and shape optimization and (*b*) shape optimization on optimized and initial arrangements for DDTD

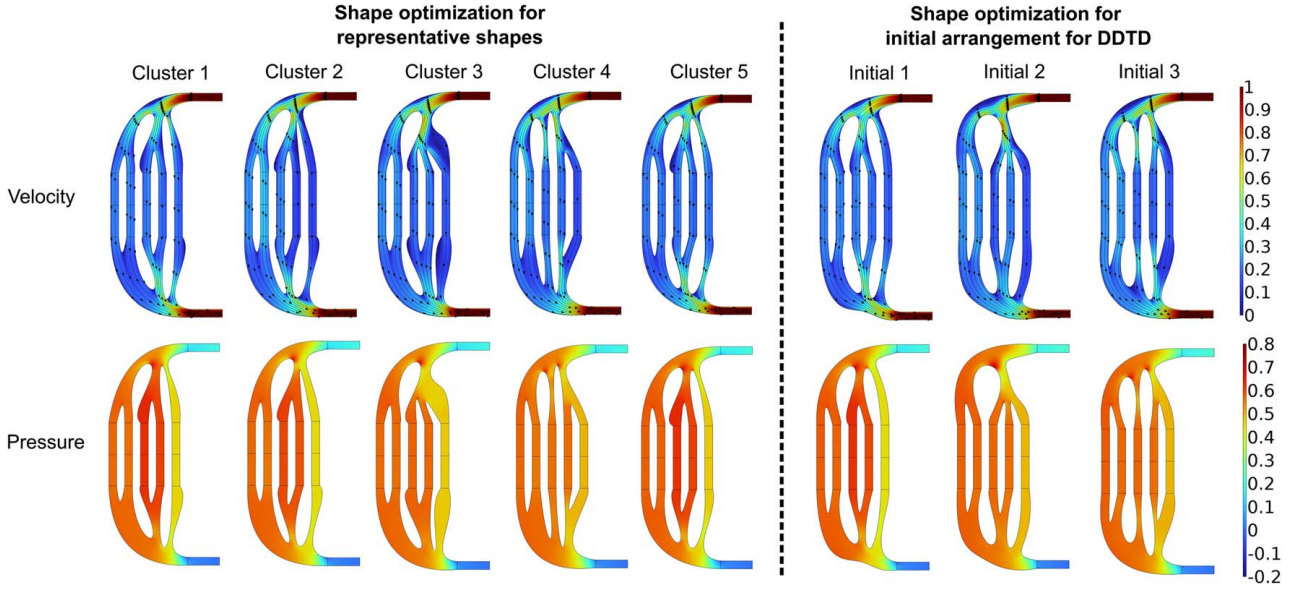

Fig. 14 Velocity and pressure distribution of optimized shapes

as well and compare to the final results of shape optimization for representative ones shown in Fig. 13(*a*). We choose three arrangements, which have the same branch pattern as three patterns seen in optimized arrangements visually, from the input of DDTD.

Figure 13(*b*) shows the comparison results of shape optimization for generated shapes via data-driven morphological exploration and initial arrangements on DDTD. Compared to the results of shape optimization for initial arrangements, shapes with superior performance are incidental to data-driven morphological exploration in the proposed framework.

The velocity and pressure distribution of optimized shapes by shape optimization are shown in Fig. 14. In this figure, a rank-one solution from each Pareto front shown in Fig. 13(*b*) is listed. Compared to the shapes of the optimized solution by DDTD, the wavy shapes of walls are restrained by using the Bezier curves to represent geometries. The wall shapes between the inlet and the most left path, and the most right path and the outlet of the result for initial arrangements of DDTD are undulating in comparison to the result for representative shapes. This is because the Bezier curves of initial arrangements have more control points on the curve and a higher degrees-of-freedom than representative shapes. On the other hand, this result indicates that the high degrees-of-freedom does not necessarily affect the high performance. With respect to the objective values, a similar trend is seen; flow uniformity of the optimized shapes of Cluster 1 and 5 tends to be high, and the pressure drop of Cluster 3 and 4 is relatively low. This indicates that the factor for each objective function of DDTD, the wall shape between the first and second paths from the right for flow uniformity, and simplicity of the flow for pressure drop, is key for the result of shape optimization as well. To achieve high flow uniformity, more control points on the wall shape between the first and second paths from the right in the inlet-side design domain need to be placed. In spite of that shapes classified as Cluster 2 look like representative shapes of Cluster 1 and 5, there are more solutions with relatively low flow uniformity in the Pareto front of Cluster 2, than Cluster 1 and 5. This is caused by the low accuracy of approximation by the Bezier curves. The wall shape between the first and second paths from the right in the inlet-side design domain of the representative shape of Cluster 2 is narrower than the shape of the representative shapes of Cluster 1 and 5, so the approximation process shown in Sec. 4.2.1 does not accurately represent this wall shape of the representative shape of Cluster 2. Considering the above, the appropriate allocation of control points on the Bezier curves is achieved in the representative shapes of Cluster 1, 3, 4, and 5 through the process of DDTD.

These results proved that the proposed multi-stage framework is effective for designing a high-performance turbulent pipe system.

6 Conclusion

In this article, we propose a multi-stage framework for designing turbulent shapes with branching, which is impractical in finding promising solutions with conventional gradient-based topology optimization due to the complexity of the combination of branches and turbulence. This framework consists of data-driven morphological exploration by DDTD and VaDE, and shape optimization based on a real-coded genetic algorithm and enables dealing with complex design problems rationally incorporating the stepwise determination of arrangements and shapes. In order to verify the effectiveness of this framework, we demonstrate its application to a two-dimensional turbulent pipe system with a minimax objective. We found that objective values of optimized solutions by DDTD completely dominate those of the initial ones and optimized solutions. Then, those solutions are classified into several clusters without relying on the visual inspection and representative solutions of each cluster are generated by VaDE. Finally, we perform shape optimization based on a real-coded genetic algorithm for generated shapes via data-driven morphological exploration and initial arrangements for DDTD and compare the objective values of optimized solutions. The result shows that optimized solutions of shape refinement for generated shapes via data-driven morphological exploration are superior to those for initial arrangements.

We demonstrate that our proposed framework is favorable for generating promising solutions for a two-dimensional turbulent pipe system. On the other hand, additional considerations, such as three-dimensional problem settings, are necessary to address practical problems. By considering the possibility of further combinations of branching and the computational cost with a three-dimensional problem setting, our proposed method is expected to show significant potential for application in three-dimensional turbulent pipe systems.

Acknowledgment

K.Y. was supported by JSPS KAKENHI Grant No. 23H03799, and K.F. was supported by JSPS KAKENHI Grant No. 23K28370.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Thévenin, D., and Janiga, G., 2008, *Optimization and Computational Fluid Dynamics*, Springer-Verlag, Heidelberg, Germany.

[2] Mäkinen, R., Neittaanmäki, P., Périaux, J., and Toivanen, J., 1998, "A Genetic Algorithm for Multiobjective Design Optimization in Aerodynamics and Electromagnetics," Proceedings of the ECCOMAS 98 Conference, Athens, Greece, Sept. 7–11.

[3] Shahrokhi, A., and Jahangirian, A., 2007, "Airfoil Shape Parameterization for Optimum Navier-Stokes Design With Genetic Algorithm," Aerosp. Sci. Technol., **11**(6), pp. 443–450.

[4] Lee, Y.-T., Ahuja, V., Hosangadi, A., and Ebert, M., 2010, "Shape Optimization of a Multi-element Foil Using an Evolutionary Algorithm," ASME J. Fluids Eng., **132**(5), p. 051401.

[5] Lee, J., and Lee, K.-S., 2013, "Correlations and Shape Optimization in a Channel With Aligned Dimples and Protrusions," Int. J. Heat Mass Transfer, **64**, pp. 444–451.

[6] Bendsøe, M. P., and Kikuchi, N., 1988, "Generating Optimal Topologies in Structural Design Using a Homogenization Method," Comput. Methods Appl. Mech. Eng., **71**(2), pp. 197–224.

[7] Sigmund, O., and Petersson, J., 1998, "Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing With Checkerboards, Mesh-dependencies and Local Minima," Struct. Optim., **16**(1), pp. 68–75.

[8] Yoon, G. H., 2016, "Topology Optimization for Turbulent Flow With Spalart-Allmaras Model," Comput. Methods Appl. Mech. Eng., **303**, pp. 288–311.

[9] Dilgen, C. B., Dilgen, S. B., Fuhrman, D. R., Sigmund, O, and Lazarov, B. S., 2018, "Topology Optimization of Turbulent Flows," Comput. Methods Appl. Mech. Eng., **331**, pp. 363–393.

[10] Yoon, G. H., 2020, "Topology Optimization Method With Finite Elements Based on the k-ε Turbulence Model," Comput. Methods Appl. Mech. Eng., **361**, p. 112784.

[11] Chirehdast, M., Gea, H.-C., Kikuchi, N., and Papalambros, P. Y., 1994, "Structural Configuration Examples of an Integrated Optimal Design Process," ASME J. Mech. Des., **116**(4), pp. 997–1004.

[12] Nguyen, T. T., Bærentzen, J. A., Sigmund, O., and Aage, N., 2020, "Efficient Hybrid Topology and Shape Optimization Combining Implicit and Explicit Design Representations," Struct. Multidiscipl. Optim., **62**(3), pp. 1061–1069.

[13] Koch, J. R. L., Papoutsis-Kiachagias, E. M., and Giannakoglou, K. C., 2017, "Transition From Adjoint Level Set Topology to Shape Optimization for 2D Fluid Mechanics," Comput. Fluids, **150**, pp. 123–138.

[14] Jiang, F., Zhao, W., Chen, L., Zheng, C., and Chen, H., 2021, "Combined Shape and Topology Optimization for Sound Barrier by Using the Isogeometric Boundary Element Method," Eng. Anal. Bound. Elements, **124**, pp. 124–136.

[15] Jang, G.-W., Kim, K. J., and Kim, Y. Y., 2008, "Integrated Topology and Shape Optimization Software for Compliant MEMS Mechanism Design," Adv. Eng. Softw., **39**(1), pp. 1–14.

[16] Sigmund, O., 2011, "On the Usefulness of Non-Gradient Approaches in Topology Optimization," Struct. Multidiscipl. Optim., **43**(5), pp. 589–596.

[17] Foster, D., 2019, *Generative Deep Learning – Teaching Machines to Paint, Write, Compose and Play*, O'Reilly Media, Sebastopol, CA.

[18] Regenwetter, L., Nobari, A. H., and Ahmed, F., 2022, "Deep Generative Models in Engineering Design: A Review," ASME J. Mech. Des., **144**(7), p. 071704.

[19] Lee, D., Chen, W., Wang, L., Chin, Y.-C., and Chen, W., 2024, "Data-Driven Design for Metamaterials and Multiscale Systems: A Review," Adv. Mater., **36**(8), p. 2305254.

[20] Guo, T., Lohan, D. J., Cang, R., Ren, M. Y., and Allison, J. T., "An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer," 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, FL, Jan. 8–12, p. 0804.

[21] Kingma, D. P., and Welling, M., 2013, "Auto-Encoding Variational Bayes," arXiv Preprint.

[22] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019, "Deep Generative Design: Integration of Topology Optimization and Generative Models," ASME J. Mech. Des., **141**(11), p. 111405.

[23] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Networks," arXiv Preprint.

[24] Yamasaki, S., Yaji, K., and Fujita, K., 2021, "Data-Driven Topology Design Using a Deep Generative Model," Struct. Multidiscipl. Optim., **64**(1), pp. 1401–1420.

[25] Yaji, K., Yamasaki, S., and Fujita, K., 2022, "Data-Driven Multifidelity Topology Design Using a Deep Generative Model: Application to Forced Convection Heat Transfer Problems," Comput. Methods Appl. Mech. Eng., **388**(2), p. 114284.

[26] Kato, M., Kii, T., Yaji, K., and Fujita, K., 2023, "Tackling an Exact Maximum Stress Minimization Problem With Gradient-Free Topology Optimization Incorporating a Deep Generative Model," Vol. 3B: 49th Design Automation Conference, p. V03BT03A008.

[27] Kii, T., Yaji, K., Fujita, K., Sha, Z., and Seepersad, C. C., 2024, "Latent Crossover for Data-Driven Multifidelity Topology Design," ASME J. Mech. Des., **146**(5), p. 051713.

[28] Luo, J.-W., Yaji, K., Chen, L., and Tao, W.-Q., 2025, "Data-Driven Multi-fidelity Topology Design of Fin Structures for Latent Heat Thermal Energy Storage," Appl. Energy, **377**, p. 124596.

[29] Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H., 2017, "Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering," Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), Melbourne, Australia, Aug. 19–25, pp. 1965–1972.

[30] Herrera, F., Lozano, M., and Verdegay, J. L., 1998, "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis," Artif. Intel. Rev., **12**(4), pp. 265–319.

[31] Duysinx, P., and Sigmund, O., 1998, "New Developments in Handling Stress Constraints in Optimal Material Distribution," Vol. 3, pp. 1501–1509.

[32] de Boor, C., 1978, *A Practical Guide to Spline*, Vol. 27, Springer-Verlag, New York, NY.

[33] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evolut. Comput., **6**(2), pp. 182–197.

[34] Atienza, R., 2018, *Advanced Deep Learning With Keras: Apply Deep Learning Techniques, Autoencoders, GANs, Variational Autoencoders, Deep Reinforcement Learning, Policy Gradients, and More*, Expert Insight, Packt Publishing, Birmingham, UK.

[35] Hinton, G. E., and Salakhutdinov, R. R., 2006, "Reducing the Dimensionality of Data With Neural Networks," Science, **313**(5786), pp. 504–507.

[36] Eshelman, L. J., and Schaffer, J. D., 1993, "Real-Coded Genetic Algorithms and Interval-Schemata," *Foundations of Genetic Algorithms, Vol. 2 of Foundations of Genetic Algorithms*, L. D. Whitley, ed., Morgan Kaufmann Publishers, San Mateo, CA, pp. 187–202.

[37] Launder, B. E., and Spalding, D. B., 1974, "The Numerical Computation of Turbulent Flows," Comput. Methods Appl. Mech. Eng., **3**(2), pp. 269–289.