

# Design Optimization for Bellow Soft Pneumatic Actuators in Shape-Matching\*

Yao Yao<sup>1</sup>, Yuwen Chen<sup>2</sup>, Liang He<sup>1</sup> and Perla Maiolino<sup>1</sup>

**Abstract**—Design optimization of soft actuators is essential for task-oriented applications. Models derived from analytical solutions, the Finite Element Method (FEM), or empirical characterized datasets are widely used to estimate the response of the actuators during actuation, acting as the backbone for design optimization. Faced with the trade-off between speed and accuracy, substantial challenges occur when moving from simulation to optimization due to the compliant, high degree of freedom, and high-dimensional design space of the soft-bodied robot. FEM becomes increasingly computationally expensive with increased design complexity during optimization iterations, while the data-driven modeling approach (e.g., Artificial Neural Network) consumes significant resources prior to optimization. To address the challenge of highly nonlinear and non-convex design optimization in soft robots using the black box modeling, this paper compares of Bayesian optimization (BO) algorithm and genetic algorithm (GA) with FEM and Artificial Neural Network (ANN) models. The shape-matching of a multi-legged robot (a starfish) is demonstrated as an example of a task-oriented design scenario that presents design optimization challenges of the design space scalability. The experimental results show that the bi-level BO outperforms BO with FEM by achieving 2.8 to 9.8 times smaller objective values within a certain time for low-dimensional design problems; GA with the ANN model can achieve lower objective values 3 to 18 times faster in high-dimensional design problems than bi-level BO with FEM in low-dimensional design problems.

## I. INTRODUCTION

The field of soft robotics is rapidly developing rapidly, extending the paradigm of traditional rigid robotics because (1) lightweight and low-cost soft materials provide safe interaction, compliance, and mobility during tasks; (2) design flexibility leads to different deformations, which allows morphological adaptability; (3) the commonly used pressurized air driving method reduces the complexity of control [1]. Pinski et al. decomposed the design architecture of soft robots into subspaces [2], mapping high-level tasks to robot morphology to detailed design optimization problems. For example, fish-like underwater robots are optimized to maximize swimming frequency and velocity [3], while finger-based gripping robots are optimized in their workspace and force for safe grasping [4]. Optimizing the geometry of a complex soft body to perform a specific action (task) through actuation is crucial for the design of soft robots.

\*This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/V000748/1;

<sup>1</sup>Yao Yao, Liang He and Perla Maiolino are with Oxford Robotics Institute, University of Oxford, Oxford, OX1 2JD, United Kingdom yao.yao/liang.he/perla.maiolino@eng.ox.ac.uk

<sup>2</sup>Yuwen Chen is with the Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, United Kingdom yuwen.chen@eng.ox.ac.uk

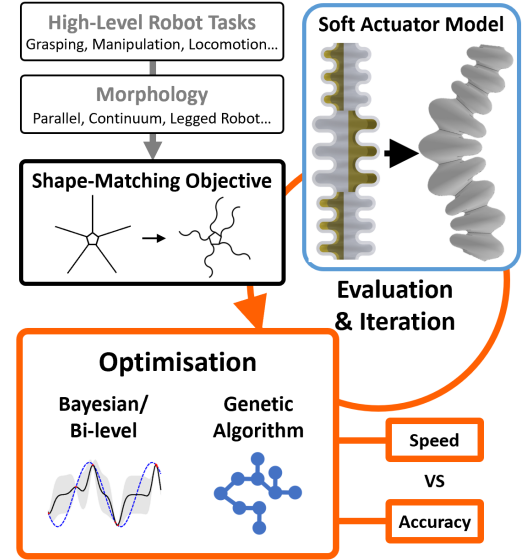


Fig. 1. Schematic diagram of the soft robot design process, from high-level robot tasks to specific morphology to detailed design objectives such as shape matching. The soft robot design is iteratively optimized using optimization algorithms (Bayesian optimization and Genetic Algorithm are considered in this paper), with models that predict the configuration of soft robots. The ultimate challenge of the design optimization of the soft bodied robot is the trade-off between speed and accuracy.

Design optimization builds on a reliable model to predict the configuration and response of the soft robot, bridging the gap between the design space and the objectives of the task. Analytical models are commonly used for fast evaluation in the optimization process [5]. However, theoretical models for continuum robots lack generality and are not suitable for design problems of complex and diverse applications [2]. In contrast, the Finite Element Method (FEM) is generally adaptability and easy to implement [6]. It discretizes the structure into small elements and solves the physics problem in an element-wise manner, accurately predicting the performance of soft robots with different features [7][8][9]. Nevertheless, FEM can be computationally expensive and have convergence issues due to the large deformation and nonlinear material properties of soft robots, posing efficiency challenges in design optimizations[10].

Data-driven methods using neural networks to learn and generate models from reliable datasets have been proposed for efficient design optimization of soft robots [2]. Supervised Artificial Neural Networks (ANNs) are capable of accurately predicting the performance of parameterized soft actuators by learning labeled training data [11], but this approach requires large-scale data collection in simu-

lations/experiments to generate an accurate model, moving most of the resource cost ahead of the optimization.

Both FEM and ANN model lack analytical sensitivity, treated as black boxes in the optimization process. In addition, the large deformation, material nonlinearity, and complex boundary conditions of soft robots present a highly nonlinear and non-convex design space, making gradient-based optimization algorithms prone to falling into local minima and highly dependent on the selection of initial points. In this situation, global optimization approaches, such as model-based and heuristic search algorithms, have been adopted as alternative approaches [12]. Model-based global optimization algorithms create models to learn the objective function's behavior and determine the next evaluation point, keeping the iteration number small [13]. Heuristic search algorithms use strategies inspired by biological behaviors to evaluate and update a group of possible candidates for better search in the space [14].

Despite the wide use of these global optimization algorithms in soft robot design problems, selecting an optimization algorithm is still challenging due to the diversity of available algorithms and the lack of guidance on their suitability for different design spaces and simulation methods [5]. Moreover, the trade-off between computational resources and performance in optimization algorithms selection affects the usability of the methods with adaptable tasks that update with environmental changes.

Therefore, this paper investigates optimization algorithms targeting FEM and ANN models for the shape-matching design optimization of bellow soft pneumatic actuators (bellow-SPAs), see Fig. 1. We specialize in bellow soft pneumatic actuators (SPAs) because their modular design enables various deformation behaviors, which have been used for wide applications in soft robotics [15]. Shape-matching is chosen as a case study for the optimization objective because many tasks require actuators to achieve specific shapes/geometries through actuation [16], [17]. The widely implemented scenario can be simplified to match the desired shapes at critical points along the trajectory. After formulating the design optimization problem, Bayesian optimization (BO) and genetic algorithm (GA) are selected as representative algorithms among model-based and heuristic search algorithms, combined with FEM and ANN models, respectively. Their performance is evaluated through the shape-matching of a starfish design (a simple multi-legged soft robot design) with up to five legs, each consisting of multiple segments of constant curvature. The dimension/number of design variables is varied to compare the converge performance of the objective in terms of iteration numbers of different optimization algorithms, providing a selection strategy for shape-matching design optimization problems.

The rest of this article is organized in Section (II) the design of bellow SPAs and models to predict their performance, (III) the formulated design optimization problem and algorithms, (IV) the results of comparing algorithms for designing starfish-inspired soft robots, (V) the discussion and

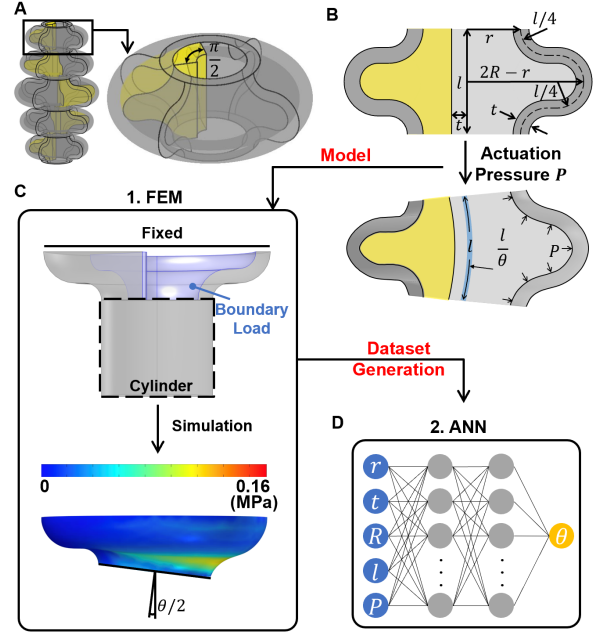


Fig. 2. The design and models of bellow SPAs. (A) A bellow SPA consists of different modules. (B) The parameterized module and its kinematics model after actuation. (C) The FEM model and stress distribution of half module. (D) The ANN model trained from FEM dataset.

conclusion.

## II. DESIGN AND MODELS

The design of the target bellow SPAs is shown in Fig. 2. The actuator consists of parameterized individual modules (Fig. 2A and B). By tuning the inner radius  $r$ , wall thickness  $t$ , average radius  $R$  and module length  $l$ , and the orientation of the filling (a quarter fan-shaped filling shown in yellow), the actuator can achieve different bending radius and direction. Angular deflection  $\theta$  is defined in Fig. 2B following the constant curvature assumption with an associate arc represented by arc length  $l$  and arc curvature  $\frac{\theta}{l}$  to describe the actuator motion with driving pressure  $P$ .

To simulate the actuator's deformation using FEM, a stationary study in 3D solid mechanics is performed in COMSOL MultiPhysics® on the half bellow module thanks to the geometrical symmetry. As shown in Fig. 2C, the wider end is set as a fixed boundary, while the other end is connected to a solid cylinder to smooth element deformation. Air pressure actuation is modeled as a uniformly distributed load on the internal surface of the half module. Besides, an incompressible Neo-Hookean model is applied to the material [6]. After simulation, the angular deflection of half module is calculated based on the displacement of points at the end of the cylinder.

The FEM model establishes a relationship between the design parameters of the bellow SPA module and its angular deflection with respect to the actuation pressure, which can be written as  $\theta = f_1(r, t, P, R, l)$ . It can also be used to generate a dataset for data-driven modeling method. A supervised ANN is used to learn the relationship between input (design parameters and actuation pressure) and output

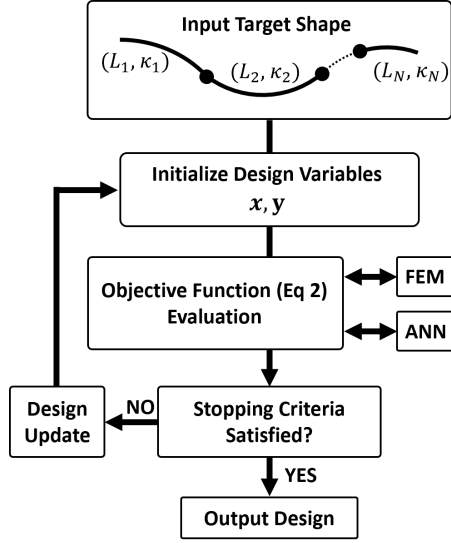


Fig. 3. The design optimization process of bellow SPAs for shape-matching objectives.

angular deflection data to generate an ANN model that can be written as  $\theta = f_2(r, t, P, R, l)$ . The accuracy of this ANN model is largely dependent on the dataset scale and collection method, resulting in different levels of resource costs.

Both models can be used to predict the response of soft robots in design optimization, where the FEM consumes computational resources during optimization and the ANN model moves most of it to the front.

### III. DESIGN OPTIMIZATION

Given the target shape of the actuator, our goal is to find the optimal design parameters for the bellows SPA such that the deformed actuator approximates the shape when pressurized and satisfies all hardware constraints. The target shape is usually approximated by arcs and line segments using a piecewise constant curvature segmentation algorithm that follows the  $G^1$  continuity [17]. In this work, we prescribe the number and parameters of each arc segment and assume that the actuator is in a linear shape before actuation.

As the bellow SPA is parameterized on its modular design, each arc segment can be designed by stacking modules with the same geometric parameters and fan-shape filling position. For stacking convenience, all modules share the same inner radius  $r$  and wall thickness  $t$  but may have different average radius  $R$  and module length  $l$ . Thus, the deformed actuator shape is defined by the shared and respective geometric parameters of each segment, in addition to its actuation pressure.

#### A. optimization Problem Formulation

The process of the shape-matching design optimization of bellow SPAs is shown in Fig. 3. Assuming the actuator has  $N$  arc segments with arc length and curvature pairs  $[(L_1, \kappa_1), (L_2, \kappa_2), \dots, (L_N, \kappa_N)]$  different from each other, we wish to determine the shared parameters  $\mathbf{x} = [r, t, P] \in \mathbb{R}_+^3$  and a set of respective geometric parameters  $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N] = [(R_1, l_1), (R_2, l_2), \dots, (R_N, l_N)] \in \mathbb{R}_+^{2N}$

of the actuator that can approximate the configuration of each segment as closely as possible.

Therefore, the objective is to minimize the difference between designed segment parameters and prescribed target ones. The hardware constraints are specified by (1) the maximum actuation pressure  $P_{\max}$ , which depends on the degree of SPA deformation, the selection of SPA material and air supply capability, (2) the maximum module radius, limited by the minimum arc radius of segments  $\frac{1}{\kappa_{\max}}$ , where  $\kappa_{\max}$  is the maximum curvature of segments, (3) the minimum wall thickness  $t_{\min}$ , which should be set to prevent leakage; (4) the maximum module length, restricted by the segment length, and (5) inherent design constraints of bellow SPA module, namely

$$\begin{cases} \frac{r}{4} \leq t \leq \frac{r}{2} \\ r + t \leq R_i \leq 2r, \quad \forall i \in [N] \\ 4t \leq l_i \leq 4(R_i - r) \end{cases}, \quad (1)$$

where  $[N]$  is the indexing set of  $N$  segments. Therefore, the shape-matching design optimization of bellow SPAs can be formulated as

$$\begin{aligned} \min h(\mathbf{x}, \mathbf{y}) &:= \frac{1}{N} \sum_{j=1}^N d_i(\mathbf{x}, \mathbf{y}_i) \\ \text{s.t.} \quad &\begin{cases} 0 \leq P \leq P_{\max} \\ 0 < r \leq \frac{1}{\kappa_{\max}} \\ \max(t_{\min}, \frac{r}{4}) \leq t \leq \frac{r}{2} \\ r + t \leq R_i \leq \min(\frac{1}{2\kappa_i} + \frac{r}{2}, 2r) \\ 4t \leq l_i \leq \min(L_i, 4(R_i - r)), \quad \forall i \in [N] \end{cases}, \end{aligned} \quad (2)$$

where

$$d_i(\mathbf{x}, \mathbf{y}_i) = w_1 |L_i - m_i l_i| + w_2 \left| \kappa_i - \frac{\theta_i}{l_i} \right|. \quad (3)$$

$m_i = \lceil \frac{L_i}{l_i} \rceil$  is the number of bellow SPA modules in the  $i$ -th segment, and  $w_1 = 1/L_i$ ,  $w_2 = 1/\kappa_i$  are the weighting factors used to balance the difference of units. The value of  $\theta_i$  relies on  $\mathbf{x}, \mathbf{y}_i$  and is obtained from either FEM  $f_1(\mathbf{x}, \mathbf{y}_i)$  or the ANN model  $f_2(\mathbf{x}, \mathbf{y}_i)$ .

#### B. optimization Algorithms

The objective function is highly nonlinear and non-convex due to nonlinear materials, large deformations and distributed actuation loads. Besides, the FEM from commercial software and the ANN model trained from the dataset are both treated as black boxes to the objective function. To this end, we propose two methods to deal with these challenges. One is to run FEM within Bayesian optimization, doing an online sequential inference. The other is to first collect a dataset from FEM and generate an ANN model (as explained in Sec. II), then use GA to perform optimization using the ANN model to predict the actuator configuration.

---

**Algorithm 1** BO with FEM:  $\text{BO}(h(\mathbf{x}, \mathbf{y}), K, \epsilon)$ 

---

```
1: Initialization:
2: Dataset  $\mathbb{D} = \emptyset$ . no prior of the response surface  $\hat{h}(\cdot) : (\mathbf{x}, \mathbf{y}) \rightarrow h(\mathbf{x}, \mathbf{y})$ . Accuracy  $\epsilon$  and maximum number of iteration  $K$ .
3: while  $h(\mathbf{x}^k, \mathbf{y}^k) > \epsilon$  and  $k < K$  do
4:    $k \leftarrow k + 1$ 
5:   Update a response surface  $\hat{h}^k(\mathbf{x}, \mathbf{y})$  from  $\mathbb{D}$ ;
6:    $(\mathbf{x}^k, \mathbf{y}^k) \leftarrow \operatorname{argmax}_{(\mathbf{x}, \mathbf{y})} \alpha^k(\mathbf{x}, \mathbf{y})$ ;
7:   Obtain  $\theta^k = f_1(\mathbf{x}^k, \mathbf{y}^k)$  from FEM and then  $h(\mathbf{x}^k, \mathbf{y}^k)$ ;
8:   Add  $\{(\mathbf{x}^k, \mathbf{y}^k), h(\mathbf{x}^k, \mathbf{y}^k)\}$  to  $\mathbb{D}$ , i.e.  $\mathbb{D} \leftarrow \mathbb{D} \cup \{(\mathbf{x}^k, \mathbf{y}^k), h(\mathbf{x}^k, \mathbf{y}^k)\}$ .
9: end while
10: Output:  $(\mathbf{x}^*, \mathbf{y}^*) = \operatorname{argmin}_{j \in \{1, \dots, k\}} h(\mathbf{x}^j, \mathbf{y}^j)$ 
```

---

1) *BO with FEM:* BO is an efficient method for solving black-box and time-consuming optimization problems [18]. It iteratively samples data of variables  $\mathbf{z}$  and the corresponding objective function evaluations  $g(\mathbf{z})$ . The dataset  $\mathbb{D} = \{\mathbf{z}, g(\mathbf{z})\}$  is used to create a probabilistic model, also known as the response surface  $\hat{g}(\cdot) : \mathbf{z} \rightarrow g(\mathbf{z})$ , that learns and mimics the behavior of the objective function. The learned model is used to determine the next set of variables to evaluate and add the new data to update the model.

The most commonly used model is the Gaussian process (GP), which places a probability distribution over the objective function with mean and variance describing the model uncertainty. The acquisition function  $\alpha(\mathbf{z})$  uses information from GP to determine the next set of variables by maximizing itself

$$\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}} \alpha(\mathbf{z}) \quad (4)$$

and updating the posterior distribution over the objective functions based on the newly added data. This method of choosing the following evaluation points based on previous results keeps the number of evaluations small, making it suitable for evaluations using FEM. Algorithm 1 shows how to combine Bayesian optimization with FEM for our design optimization problem where  $\mathbf{z} := (\mathbf{x}, \mathbf{y})$  and  $g(\mathbf{z}) := h(\mathbf{x}, \mathbf{y})$ . The iteration process continues until finding a set of variables where the objective value is below  $\epsilon$  or the number of iterations reaches its maximum value  $K$ .

Algorithm 1 has been proven to work well for many optimization problems [19], [20]. However, with the increasing number of segments (design variables), it gets harder to build GP and search for the next points with the acquisition function. It requires the scale of  $\mathbb{D}$  (equivalent to the number of iterations) to grow exponentially to keep the same quality of model approximation [21], making it computationally infeasible.

To alleviate computational burden, a bi-level variant is introduced, Algorithm 2, to improve the performance of Bayesian optimization [22] considering our optimization problem is decomposed w.r.t. variable  $\mathbf{y}$ . The optimization problem is constructed into the upper and lower level, opti-

---

**Algorithm 2** Bi-level BO with FEM

---

```
1: Initialization:
2: Dataset  $\mathbb{D} = \emptyset$  and no prior of the response surface  $\hat{h}(\cdot) : \mathbf{x} \rightarrow \min_{\mathbf{y} \in \mathcal{Y}_i} h(\mathbf{x}, \mathbf{y})$ . Accuracy  $\epsilon$  and maximum number of iteration for outer loops  $K_{ou}$  and inner loops  $K_{in}$ .
3: while  $h(\mathbf{x}^k, \mathbf{y}^k) > \epsilon$  and  $k < K_{ou}$  do
4:    $k \leftarrow k + 1$ 
5:   Update a response surface  $\hat{h}^k(\mathbf{x})$  from  $\mathbb{D}$ ;
6:    $\mathbf{x}^k \leftarrow \operatorname{argmax}_{\mathbf{x}} \alpha^k(\mathbf{x})$ ;
7:   for  $i = 1, \dots, N$  do
8:      $\mathbf{y}_i^k \leftarrow \operatorname{argmin}_{\mathbf{y}_i} d_i(\mathbf{x}^k, \mathbf{y}_i)$  via  $\text{BO}(d_i(\mathbf{x}^k, \cdot), K_{in}, \epsilon)$ ;
9:   end for
10:  Sum up  $d_i(\mathbf{x}^k, \mathbf{y}_i^k)$  to obtain  $h(\mathbf{x}^k, \mathbf{y}^k)$ ;
11:  Update dataset  $\mathbb{D}$ , i.e.  $\mathbb{D} \leftarrow \mathbb{D} \cup \{(\mathbf{x}^k, \mathbf{y}^k), h(\mathbf{x}^k, \mathbf{y}^k)\}$ .
12: end while
13: Output:  $(\mathbf{x}^*, \mathbf{y}^*) = \operatorname{argmin}_{j \in \{1, \dots, k\}} h(\mathbf{x}^j, \mathbf{y}^j)$ 
```

---

mizing over  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. During each iteration of the upper level,  $\mathbf{x}$  is obtained from the upper-level acquisition function. Then, it is treated as constant in the lower level optimization, which is decomposed into  $N$  independent BO over  $\mathbf{z} := \mathbf{y}_i$  and  $g(\mathbf{z}) := d_i(\mathbf{x}, \mathbf{y}_i)$ , where  $d_i$  is defined in formula (3). Then, the optimal value of  $d_i$  is summed up to update the objective value of the upper level. This process continues until the upper-level cost  $h$  is below the desired value  $\epsilon$  or the number of iterations reaches  $K_{ou}$ .

Although bi-level Bayesian optimization can solve Problem (2) with a desired low cost, the use of FEM prohibits the bi-level BO for a design problem with many segments. To tackle the curse of dimensionality, a method combining the ANN model with GA is introduced in the next part for solving high-dimensional design problems without calling FEM too many times.

2) *GA with ANN model:* GA is a well known evolutionary algorithm based on the concept of biological evolution [23]. It evaluates a large number of design points per iteration, which is suitable to be combined with the ANN model since the ANN model conducts evaluation efficiently.

As explained in Algorithm 3, the ANN model based on the dataset of  $M$  feasible points with their  $\theta$  values obtained from FEM is first generated. This step takes most of the time and it depends on the computational hardware, the complexity of FEM simulation and the dataset scale.

Once the ANN model is obtained, the algorithm proceeds as standard GA. A feasible generation  $\mathcal{S}$  is initialized with a fixed size of population. For each individual  $n$  inside population  $\mathcal{S}$ , the corresponding cost  $h(\mathbf{x}^n, \mathbf{y}^n)$  is obtained based on the ANN model. If the cost of the best performed point  $(\mathbf{x}^*, \mathbf{y}^*)$  of the current population members is within accuracy  $\epsilon$  or the weighted change  $\Delta h^k$  over the last several iterations is less than  $\tau$ , the algorithm terminates and returns  $(\mathbf{x}^*, \mathbf{y}^*)$  as the optimal design parameter. Otherwise, the next generation is created by preserving elites and selecting parents to produce their crossover and mutation children.

---

**Algorithm 3** GA with ANN model

---

```
1: Initialization:
2: Generate  $M$  feasible points to train a model using ANN:
    $\theta = f_2(r, t, P, R, l)$ ;
3: Set accuracy  $\epsilon$ , function tolerance  $\tau$  and the maximum
   generation number  $K$ .
4: while  $k < K$  or  $h(\mathbf{x}^*, \mathbf{y}^*) < \epsilon$  or  $\Delta h^k < \tau$  do
5:    $k \leftarrow k + 1$ 
6:   Create a new generation  $\mathcal{S}$ ;
7:   for  $n \in \mathcal{S}$  do
8:     for  $i = 1, \dots, N$  do
9:       Obtain  $\theta_i^n = f_2(\mathbf{x}^n, \mathbf{y}_i^n)$  and then  $d_i(\mathbf{x}^n, \mathbf{y}_i^n)$ ;
10:    end for
11:    Sum up  $d_i(\mathbf{x}^n, \mathbf{y}_i^n)$  to obtain  $h(\mathbf{x}^n, \mathbf{y}^n)$ ;
12:  end for
13:   $(\mathbf{x}^*, \mathbf{y}^*) = \operatorname{argmin}_{(\mathbf{x}, \mathbf{y}) \in \mathcal{S}} h(\mathbf{x}, \mathbf{y})$ ;
14:  Compute a weighted change  $\Delta h^k$ 
15: end while
16: Output: Best solution  $(x^*, y^*)$ .
```

---

#### IV. EXPERIMENTAL RESULTS

The above optimization algorithms are analyzed through the design of starfish-inspired soft robots, which presents challenges in terms of scalability related to the number of segments and parameter value ranges, and these challenges can also be found in other applications such as soft grippers and multi-segment soft arms [2][5]. The starfish design space is defined by setting the leg length between 100-150 mm and the range of motion within  $2\pi$ . Assuming that each leg consists of constant curvature arc segments, a script is used to randomly generate the length and curvature of each segment within the specified range.

To evaluate the performance of optimization algorithms for the shape-matching design optimization of bellow-SPAs, we vary the dimension of the design variables and examine the convergence performance in terms of the trade-off between speed and accuracy. Specifically, we compare the minimum objective value and running time for each algorithm as a function of the number of iterations.

The design problems with different dimensions of parameters are defined as follow: for low-dimensional designs, a single leg is considered, consisting of  $N = 1$  or 2 or 3 arc segments; while for high-dimensional designs, there are five legs all different from each other, each consisting of 1 or 2 or 3 arc segments, resulting in a total of  $N = 5$  or 10 or 15 arc segments. As FEM simulation needs to be run for each evaluation of the deformation for one arc segment, we decide to perform (bi-level) BO with FEM on low-dimensional design optimizations, and GA with ANN model on high-dimensional ones (we use BO-FEM, BiBO-FEM and GA-ANN to represent each algorithm for simplicity in the following content). We choose Agilus30 as the actuator material in design tasks [24]. Based on our previous characterization [8], the boundary of the actuation pressure and wall thickness are set to be  $P_{\max} = 10$  kPa and  $t_{\min} = 1$

mm.

The algorithms are implemented in MATAB and executed on a laptop with an Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz and 16.0 GB RAM. To generate an ANN model based on a FEM dataset, it takes approximately 6 hours (the table of parameter ranges and intervals for FEM dataset collection is presented in supplementary video). For the stopping criteria, we specify the maximum number of iterations for (bi-level) BO and the function tolerance for GA. The  $K$  of BO-FEM are set to be 300, 500, and 1000 for the corresponding low dimensional design problems. For all low dimensional design problems of BiBO-FEM, the  $K_{ou}$  and  $K_{in}$  are set to be 30 and 15 for the upper and lower levels, respectively. As for GA-ANN, the function tolerance  $\tau$  is set to be  $1 \times 10^{-6}$ , which halts the algorithm if the average change of the objective value in the last 50 generations is less than it.

The evaluation experiment is performed 3 times for each design problem, one of which is shown in Fig. 4(A) to (F), where the minimum objective values and corresponding computational times are plotted against the number of iterations (the remaining results are included in the supplementary video). We also calculate the mean and standard deviation of the final minimum objective value, its respective iteration number, and running times for each design problem, as shown in Table I. We focus on the minimum target value range between 0 and 0.1, as it is where the design parameters lead to viable solutions for shape matching. The shape of the design parameters based on objective values at different levels in this range is compared with the target shape in Fig. 4(G).

The results of low-dimensional design problems using BO-FEM and BiBO-FEM are shown in Fig. 4(A) to (D) and Table I. Firstly, all experiments reach their final minimum objective values (or a value very close to it with a difference smaller than 0.05) before half of the maximum number of iterations, validating BO's property of keeping the iteration number small. Second, the results using BO-FEM show that only the final minimum objective value of 0.0307 for  $N = 1$  is below 0.1, which is 2.8 times the corresponding result using BiBO-FEM, 0.0117. While the final minimum objective values for  $N = 2$  and 3 are 0.1382 and 0.2453, respectively, which are 5 and 9.8 times the corresponding results of 0.0276 and 0.0251 using BiBO-FEM. This demonstrates that BO-FEM cannot handle the scalability of the number of segments, and is only suitable for the design problem of  $N = 1$ . Third, in the design problems using BiBO-FEM, the running time required to achieve the final minimum objective value for  $N = 2$  is longer than  $N = 3$ , which is 25252 s and 23031 s, respectively. This is because there is one experiment for  $N = 2$  achieving its final minimum 0.0414 at iteration 20 after reaching 0.0473 at iteration 7. Apart from this exception, in general, the running time of BiBO-FEM increases with the increase of the number of segments. However, due to the high dependence the FEM model complexity, the running time of BiBO-FEM in relation to the number of iteration is very unpredictable.

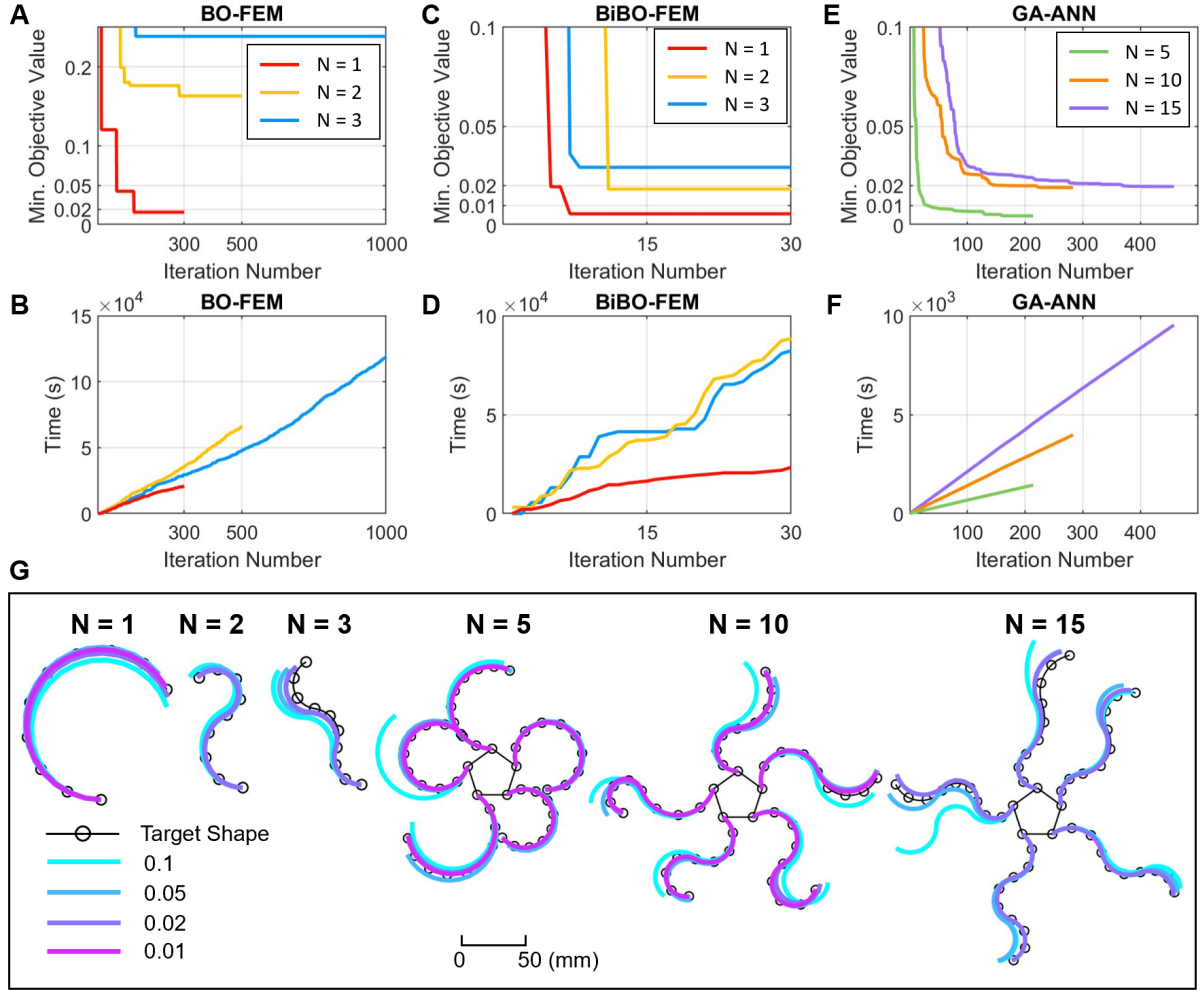


Fig. 4. The evaluation results of optimization algorithms. The iteration number vs min. the objective value of (A) BO-FEM, (C) BiBO-FEM, and (E) GA-ANN for their different design problems. (B) (D) (F) The iteration number vs running time of the corresponding algorithms. (G) The comparison of the designed shapes of objective values at different levels with the target shapes, where the design parameters of the low-and high-dimensional problems are from BiBO-FEM and GA-ANN, respectively.

Additionally, for the design problem of  $N = 3$  using BiBO-FEM, the average running time of 23031 s required to reach the final minimum objective value is very close to our time for collecting the FEM dataset to generate the ANN model, indicating that GA-ANN could be a more efficient approach for design problems with more than 3 segments.

The results of high-dimensional design problems using GA-ANN are presented in Fig. 4(E)(F) and Table I. Firstly, the average final minimum objective value increases as the number of segments increases. Second, when comparing the performance of high-dimensional design problems using GA-ANN to low-dimensional ones using BiBO-FEM, the mean and standard deviations of the final minimum objective values of the former are generally smaller than those of the latter, indicating that GA-ANN is more robust in terms of objective value convergence. Moreover, the average running time of 1265.2 s for  $N = 5$  is 18.2 times smaller than 23031s for  $N = 3$  (which is equivalent to the time required to generate our ANN model). Although the running time increases with the increase of the number of segments, the longest time of 7652.8 s for  $N = 15$  is still almost 3

TABLE I  
EVALUATION RESULTS OF OPTIMIZATION ALGORITHMS FOR DIFFERENT DESIGN PROBLEMS

Algorithm	Number of Segment(s)	Final Min. Objective Value	Respective Iteration Number	Respective Running Time (s)
BO & FEM	1	$0.0307 \pm 0.0216$	$132.7 \pm 12.4$	$12414 \pm 2105$
	2	$0.1382 \pm 0.0266$	$304.3 \pm 112.5$	$38653 \pm 21439$
	3	$0.2453 \pm 0.0561$	$566.3 \pm 392.9$	$118590 \pm 104372$
Bi-BO & FEM	1	$0.0117 \pm 0.0128$	$7.3 \pm 4.5$	$9899 \pm 10329$
	2	$0.0276 \pm 0.0122$	$12 \pm 7.5$	$25252 \pm 9799$
	3	$0.0251 \pm 0.0036$	$9 \pm 1$	$23031 \pm 5110.7$
GA & ANN	5	$0.0036 \pm 0.0027$	$188.7 \pm 36$	$1265.2 \pm 233.6$
	10	$0.0171 \pm 0.0049$	$233.3 \pm 11.9$	$3169.4 \pm 197.9$
	15	$0.0208 \pm 0.0036$	$369.3 \pm 34.7$	$7652.8 \pm 824$



times smaller than the time for generating the ANN model. Additionally, the running time complexity in relation to the increase of the number of iterations and the number of segments is more proportional for GA-ANN than BiBO-FEM, making the former more time-predictable.

Besides, Fig. 4(G) shows the comparison between the designed shape and the target shape for each design problem. The black line with circular markers represents the target shape, and other colored lines represent the designed shapes based on objective values around 0.1, 0.05, 0.02 and 0.01, respectively. As the objective value decreases and the line color transitions from light blue to magenta, the designed shapes gradually fits the desired shape, demonstrating the effectiveness of our optimization method in Sec III-A (clearer representation and detailed design parameters are included in the supplementary video).

## V. DISCUSSION AND CONCLUSION

Facing with the challenge from the trade-off between speed and accuracy in design optimization of soft robots using black-box modeling methods, we conducted experiments comparing the convergence performance of (bi-level) BO and GA targeting FEM and ANN models for shape-matching design optimization of bellow-SPAs in a starfish-inspired soft robotic scenario.

Results demonstrate that for low-dimensional design problems, BiBO-FEM outperforms BO-FEM by achieving 2.8 to 9.8 times lower objective values within a given running time, indicating that the bi-level/decomposition method is an effective strategy for reducing the modeling complexity of BO. For high-dimensional design problems, GA-ANN not only achieves low objective values efficiently but also demonstrates robustness and time prediction in the convergence of objective values. Overall, considering computational resources, BiBO-FEM and GA-ANN are recommended for low- and high-dimensional design problems, respectively. However, the boundary between low and high dimensions depends on the collection method and dataset size.

One limitation of this study is that we only varied the dimension of design variables, and the stopping criteria were determined empirically. In future work, we plan to conduct a more comprehensive evaluation of optimization algorithms. This evaluation may involve varying optimization constraints, exploring the impact of different maximum iteration numbers for the lower level in BiBO-FEM, and optimizing the parameters (such as population size, crossover fraction, etc.) of GA. Furthermore, it would be valuable to extend the optimization framework to other types of actuators and consider more complex scenarios.

## REFERENCES

- [1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, p. 467–475, 2015.
- [2] J. Pinski and D. Howard, "From bioinspiration to computer generation: Developments in autonomous soft robot design," *Advanced Intelligent Systems*, vol. 4, no. 1, p. 2100086, 2022.
- [3] E. Nava, J. Z. Zhang, M. Y. Michelis, T. Du, P. Ma, B. F. Grewe, W. Matusik, and R. K. Katzschmann, "Fast aquatic swimmer optimization with differentiable projective dynamics and neural network hydrodynamic models," in *39th International Conference on Machine Learning*, pp. 16413–16427, 2022.
- [4] H. Zhang, A. S. Kumar, J. Y. H. Fuh, and M. Y. Wang, "Design and development of a topology-optimized three-dimensional printed soft gripper," *Soft Robotics*, vol. 5, no. 5, pp. 650–661, 2018. PMID: 29985781.
- [5] F. Chen and M. Y. Wang, "Design optimization of soft robots: A review of the state of the art," *IEEE Robotics & Automation Magazine*, vol. 27, no. 4, pp. 27–43, 2020.
- [6] M. S. Xavier, A. J. Fleming, and Y. K. Yong, "Finite element modeling of soft fluidic actuators: Overview and recent developments," *Advanced Intelligent Systems*, vol. 3, no. 2, p. 2000187, 2021.
- [7] P. Polygerinos, Z. Wang, J. T. B. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, "Modeling of soft fiber-reinforced bending actuators," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 778–789, 2015.
- [8] Y. Yao, L. He, and P. Maiolino, "A simulation-based toolbox to expedite the digital design of bellow soft pneumatic actuators," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pp. 29–34, 2022.
- [9] S. Wang, L. He, A. Albini, P. Zhang, and P. Maiolino, "A magnetorheological elastomer-based proportional valve for soft pneumatic actuators," *Advanced Intelligent Systems*, vol. n/a, no. n/a, p. 2200238.
- [10] G. Runge, J. Peters, and A. Raatz, "Design optimization of soft pneumatic actuators using genetic algorithms," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 393–400, 2017.
- [11] G. Runge, M. Wiese, and A. Raatz, "Fem-based training of artificial neural networks for modular soft robots," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 385–392, 2017.
- [12] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for optimization*. 2019.
- [13] K. K. Vu, C. D'Ambrosio, Y. Hamadi, and L. Liberti, "Surrogate-based methods for black-box optimization," *International Transactions in Operational Research*, vol. 24, no. 3, pp. 393–424, 2017.
- [14] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, p. 8091–8126, 02 2021.
- [15] A. Zolfagharian, M. A. P. Mahmud, S. Gharaie, M. Bodaghi, A. Z. Kouzani, and A. Kaynak, "3d/4d-printed bending-type soft pneumatic actuators: fabrication, modelling, and control," *Virtual and Physical Prototyping*, vol. 15, no. 4, pp. 373–402, 2020.
- [16] F. Connolly, C. J. Walsh, and K. Bertoldi, "Automatic design of fiber-reinforced soft actuators for trajectory matching," *Proceedings of the National Academy of Sciences*, vol. 114, no. 1, pp. 51–56, 2017.
- [17] G. Singh and G. Krishnan, "Designing fiber-reinforced soft actuators for planar curvilinear shape matching," *Soft Robotics*, vol. 7, no. 1, pp. 109–121, 2020. PMID: 31566502.
- [18] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, p. 5–23, 02 2016.
- [19] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, p. 235–284, jun 2008.
- [20] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2020.
- [21] M. Binois and N. Wycoff, "A survey on high-dimensional gaussian process modeling with application to bayesian optimization," *ACM Trans. Evol. Learn. Optim.*, vol. 2, aug 2022.
- [22] E. Kieffer, G. Danoy, P. Bouvry, and A. Naghi, "Bayesian optimization approach of general bi-level problems," *GECCO '17*, (New York, NY, USA), p. 1614–1621, Association for Computing Machinery, 2017.
- [23] S. Mirjalili, *Genetic Algorithm*, pp. 43–55. Cham: Springer International Publishing, 2019.
- [24] Stratasys, *Agilus30 Materials Data Sheet*, 2020.