Structural and Multidisciplinary Optimization (2022) 65:202
https://doi.org/10.1007/s00158-022-03295-w

RESEARCH PAPER

Improving the diversity of topology-optimized designs by swarm intelligence

Tsz Ho Kwok[1]

Received: 29 March 2022 / Revised: 23 May 2022 / Accepted: 7 June 2022 / Published online: 7 July 2022

Abstract
Although additive manufacturing can produce nearly any geometry, users have limited choices in the designs. Topology optimization can create complex shapes, but it provides only one solution for one problem, and existing design exploration methods are ineffective when the design space is huge and high-dimensional. Therefore, this paper develops a new generative design method to improve the diversity of topology-optimized designs. Based on the observation that topology optimization places materials along the principal directions to maximize stiffness, this paper creates a rule of principal direction and applies it to swarm intelligence for form-finding. The shapes got by the swarming process possess both randomness and optimality. After they are further optimized, the final designs have high diversity. This is the first time integrating structural stiffness as a swarm principle to influence the collective behavior of decentralized, self-organized systems. The experimental results show that this method can generate interesting designs that have not been seen in the literature. Some results are even better than those got by the original topology optimization method, especially when the problem is more complex. This work not only allows users to choose unique designs according to their preference, but also helps users find better designs for their application.

Keywords Generative design · Form-finding · Swarm intelligence · Principal stress · Topology optimization · Additive manufacturing

1 Introduction

Additive manufacturing (AM) technologies enable the fabrication of complex geometries that are difficult to produce with traditional methods. This gives unprecedented flexibility to the design of high-performance products. One method that uses this flexibility is topology optimization (TO), which optimizes the spatial-material distribution in a design-domain to maximize the structural performance. However, TO provides only one optimized solution for one problem, and designers have limited control in creating original designs. Regarding this lack of diversity, generative design (GD) has been proposed to explore innumerable solutions, some of which may not have been seen or thought of before. GD uses a set of parametric-defined rules to generate solutions and optimizes them with certain analysis methods, such as TO. This is illustrated in Fig. 1 using one degree-of-freedom (DoF) and two DoF solution spaces. Ideally, the generated solutions should be scattered around different local optima so that the optimized results have high diversity. This design method has the potential not only to find better solutions but also to create aesthetic designs.

Finding an effective scatter may be easier when the search space has a lower dimension. However, GD constructs free-form geometry and topology, and thus many problems for GD have a high-dimensional search space. Exploring such a large design space to get meaningful results is challenging, and the designs generated by the current GD algorithms still look similar in terms of human perception (Oh et al. 2019). To explain this insufficient diversity, a design-domain of 120×80 unit squares (i.e., 9, 600 DoF) is employed, see Fig. 2. If the volume fraction is 0.4 (or 40%), the solid isotropic material with penalization (SIMP) method (Andreassen et al. 2011) initializes all

Responsible Editor: Lei Wang

✉ Tsz Ho Kwok
tszho.kwok@concordia.ca

[1] Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, 1455 de Maisonneuve Blvd. W., Montreal H3G 1M8, QC, Canada

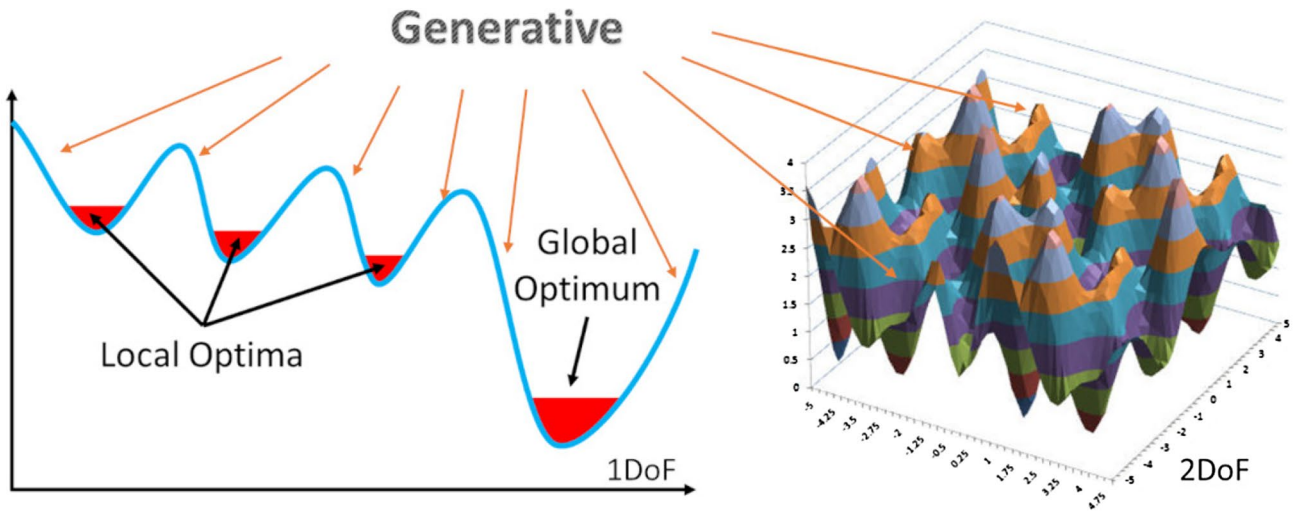

Fig. 1 Generative design explores solutions in different design spaces

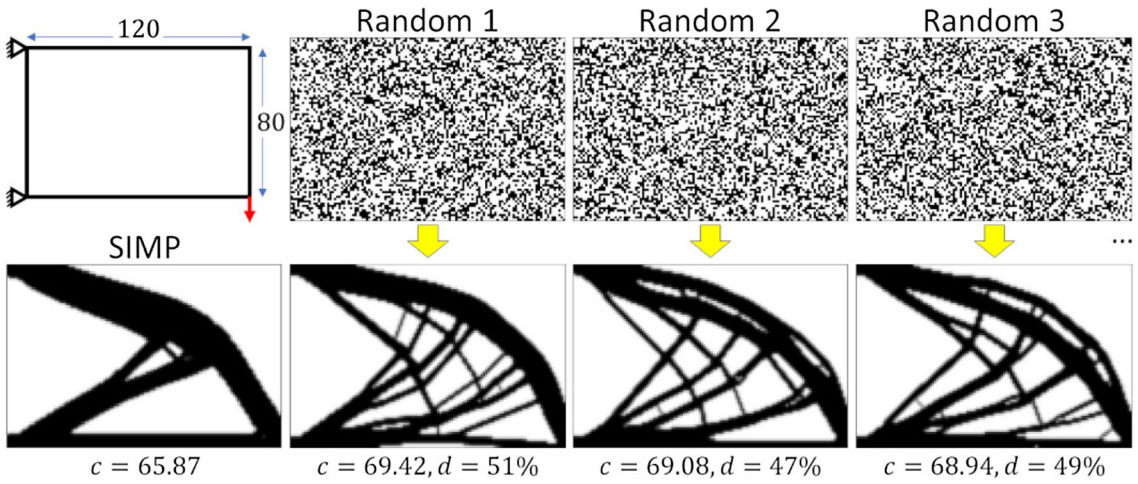

Fig. 2 Optimized results from random material distributions. c is the compliance (proportional to the amount of deformation) and d is the percentage pixel-wise difference from the original SIMP result

the cells with 0.4 density and iterates to find the optimal structure. Here, 100 random material distributions are used as the inputs for the SIMP method to mimic the generative process, and they are generated by assigning materials randomly to the cells until the volume fraction reaches 0.4. The three results that are the most different from the SIMP result and from each other are shown in Fig. 2 (see Fig. 12 for all results). The difference is measured by treating each material distribution as a high-dimensional point and calculating the distance between them. Apparently, different inputs lead to different optimized designs (local optima). However, even though these three are the most diverse results, their overall shapes still look quite similar. In fact, all 100 results are similar to each other. Although one may test more combinations and hope to see some diverse results, there are ${}_{9600}C_{3840} = 7 \times 10^{2803}$ combinations, so it would take a long time to test even a fraction of them. Therefore, there is a need to develop some intelligent ways to generate meaningful solutions to better use the potential of GD.

The above testing reveals that randomness alone is not enough to have high diversity, and the generated initial shapes must also be at least suboptimal. As TO focuses on the material layout, the generation should find a set of random forms that are already structurally meaningful. Although there are GD algorithms for form-finding (Youssef et al. 2018), they have minor consideration for structural optimality, and thus their uses are mostly in architectural applications. This motivates the present

research to include structural concerns in the generative process, aiming to create structurally suboptimal initial shapes. In studying the fundamentals in both fields of TO and GD, two observations are made. First, in order to maximize structural stiffness, materials should be placed along the direction of principal stresses (Wu et al. 2018; Sales et al. 2021). Second, among various form-finding techniques, swarm intelligence is good at creating free-form and non-repeated geometries, when the swarm was seen as the motion of particles and their paths are connected to each other forming a structure-like system (Agirbas 2019). Both share some similarity in terms of directions or trajectory, and this paper hypothesizes that combining them balances randomness and optimality in GD. Therefore, the objective of this paper is to apply principal stress from mechanics to the form-finding process in swarm intelligence, with the goal of increasing the diversity of the topology-optimized designs. The contributions are summarized as follows:

1. Using the same inputs for topology optimization, a physical field is generated for swarm intelligence. A set of swarming parameters that can consider both randomness and optimality is then defined for design generation.
2. With the physical field, a rule of principal direction is created for swarm intelligence as a steering force driving particles to move along the principal directions.
3. The trajectory of particles is converted to the inputs for topology optimization, and a means of diversity measure is used to sort the optimized results for design inspiration.

The main outcome of this paper is to generate diverse designs, but it will show that the present method can also find better designs than the original SIMP method. This is the first time fully integrating structural stiffness into swarm intelligence for generative design.

1.1 Why diversity?

TO enforcing discrete 0/1 solutions is inherently a non-convex problem (Abdelhamid and Czekanski 2022), which may have multiple feasible regions and multiple locally optimal solutions within each region. The global optimum can only be known after finding all the local optima, but it usually cannot be found in a reasonable amount of computing time, and thus we are contented with sufficiently good solutions. The diversity of topology-optimized designs thus refers to how the locally optimal designs differ from each other, and there are many reasons to increase this diversity, just to name a few: First, human has both needs and wants. For example, human only need water but want all kinds of drinks; mass-produced goods are effective, but many want one-of-a-kind products. Similarly, we may only need one design to work, but we also want the freedom to choose from the qualified options. It does not have to be the best design to satisfy all the requirements, and there could be multiple (locally) optimized designs that have similar performance. Increasing the diversity helps provide these options as many and as distinct as possible. Second, TO and GD have different purposes – the present work is a GD method. While they may have the same goal – to provide the optimal design for a set of requirements, TO is used in later phases of design where an initial design is already set, but GD is used in the early phase of design to give initial design possibilities for automated ideation. In other words, GD uses TO, but not the other way around, and GD builds on the foundation of TO to arrive at smarter and more innovative solutions. It is well known that diversity is the key to innovation. Third, seeking globally optimal solutions for non-convex problems can take exponential time with the number of variables, and many global optimization methods (e.g., branch and bound) use random or statistical sampling to improve performance. Diversity sampling is one of the effective ways to speed up the searching process.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 presents the methodology and implementation details. The experimental results are presented in Sect. 4. Finally, Sect. 5 concludes the paper and discusses the future works.

2 Literature review

This paper is mainly related to generative design and topology optimization. Their closely related works are reviewed.

2.1 Generative design (GD)

The five classic GD techniques are swarm intelligence (Agirbas 2019), cellular automata (Feijs and Toeters 2018), genetic algorithms (Vaissier et al. 2019), shape grammars (Wang et al. 2020), and L-system (Zhang et al. 2020). They are mostly rule-based methods that create new generations based on some rules to determine the new state of each parameter according to its current states as well as the neighbors'. One the one hand, if a solution is represented as a point in a n-dimensional space, where n is the number of parameters, the generative process can be used to search for solutions. For example, Felkner et al. (2013) encoded the shape, topology, and sizing variables in a particle and generated architectural designs of truss structures. This class of works is mainly used to find the best solution, such as particle swarm optimization (Li et al. 2014). On the other hand, if the generations are done in the Euclidean space, they can be used to create emergent geometries as a form-finding process

(Chen 2015). Agirbas (2019) further created non-Euclidean geometries by substituting the particles using metaballs or Voronoi cells. These methods are mainly used in architecture, like façade designs considering the level of daylight (Pantazis and Gerber 2018), but not so much for engineering performance. Although Tsiliakos et al. (2012) took the stress into account during the material growth, he only put more materials in the high-stress area, and the results were not optimized.

With the growth of machine learning, it is also applied to GD constructing a generator that learns the probability distribution of data and generates new data based on the learned probability distribution. For example, Yoo et al. (2021) integrated deep learning in the conceptual design phase to generate wheel designs and evaluate their engineering performance. The learning-based methods can find the hidden representations (latent variables) from the high-dimensional data, and sampling in the latent space is one way to generate different but reasonable results. This has been employed in the generation of product profiles (Dogan et al. 2019); (Gunpinar et al. 2019), lattice structures (Gupta et al. 2019), and human body models (Huang et al. 2019). Other related works include exploring design spaces for manufacturing considerations (Mirzendehdel et al. 2019) and generating free-form grid shells (Wang et al. 2019).

2.2 Topology optimization (TO)

TO answers the fundamental engineering question (Sigmund and Maute 2013): "how to place material within a prescribed design domain to obtain the best structural performance?" Bendsøe (1989) stated it as "optimizing the spatial material distribution in a design domain for given loads and boundary conditions." There are various TO methods, such as ground structure (Zhang et al. 2017), homogenization (Bendsøe and Kikuchi 1988), SIMP (Zhou and Rozvany 1991; Bendsøe and Sigmund 1999), and level set (Sethian and Wiegmann 2000; Wang et al. 2003; Allaire et al. 2004). The popular TO methods are often gradient-based optimization methods, which start with an initial material distribution and update it repeatedly according to the finite element analysis (FEA) result until the convergence is reached. Anisotropic material properties can also be considered in TO using non-homogeneous failure criteria (Mirzendehdel et al. 2018). To speed up the process, Kwok et al. (2016) converted the optimization problem to a design problem by drawing lines along the direction of principal stress. While FEA only has low-order continuity (C0), research has applied isogeometric analysis (IGA) to TO problems, getting more effective results (Gao et al. 2019).

When the problems are more complex, metaheuristic algorithms can often find a sufficiently good solution with less computational effort (Bianchi et al. 2009). For example, research used genetic algorithm (Wang and Tai 2005), ant colony optimization (Kaveh et al. 2008), differential evolution (Wu and Tseng 2010), harmony search (Lee and Han 2017), bat algorithm (Jaafer et al. 2020) to search for a design with higher structural performance. Recently, there are many uses of machine learning for TO problems too, e.g., generative adversarial network (GAN). A GAN model involves a generator and a discriminator, and they are trained together in a zero-sum game until the discriminator is fooled about half the time, meaning the generator is generating good designs. To take advantage of GAN in TO applications, Oh et al. (2019) generated a set of results by running TO under different parameters and used the results to train the generator. When the discriminator cannot identify the TO results and the generated ones, the generator is able to create near-optimal topological design in one-shot without any iteration (Yu et al. 2019; Cang et al. 2019). The methods were further extended to generate optimal results even when the boundary conditions are different. Nie et al. (2021) developed TopologyGAN that takes an additional input of physical field to the generator of a conditional generative adversarial network (cGAN) to predict the result under unseen boundary conditions. Similarly, Hertlein et al. (2021) used a cGAN trained on randomized boundary conditions, which can also consider build orientations and overhangs in additive manufacturing. These methods applied generative methods to create designs, but they focused on finding the best structure rather than generating diverse solutions.

2.3 Diversity in TO

When there is a well-defined measure of diversity, the optimization can include a constraint based on the diversity metric. For example, Wang et al. (2018) used cross-correlation (CC) and sum of squared differences (SSD) to make sure the newly generated design has a minimum difference from an existing design. Li et al. (2021) developed a diversity metric based on Gaussian process model to maximize the number of independent designs among a population. However, the new design depends on previous designs, and thus these methods cannot run fully in parallel. In addition, it is sometimes difficult to define a metric for perceptual diversity. Instead of adding an extra constraint, Deng and To (2021) presented a parametric level set method using deep learning for TO and generated different designs by changing the parameters. Similarly, Watson et al. (2021) generated different topology-optimized designs by changing the TO settings (e.g., volume fraction), but the results were not comparable (e.g., different volumes), and the number of permutations was small. He et al. (2020) integrated genetic algorithms (GA) into TO, altering the initial and the intermediate structures during optimization. While GA generates new offspring from the

existing population, the present work uses swarm intelligence to create new designs directly from the underlying physical field. Sun and Ma (2020) applied reinforcement learning to alter the search direction in the TO problem, and their method can generate various acceptable design options, but the diversity is not high, as shown in the result section.

In summary, although GD techniques can synthesize various geometries directly, previous works mostly employed them for optimization, treating each design as a point in the solution space, like using bit-array representations. This creates difficulties in generating diverse designs even when the optimization has considered the diversity.

3 Methodology

With the goal of creating diverse designs, this study applies form-finding using swarm intelligence to the generation of various initial geometries for TO. The forms need to be both random and suboptimal to capture as many local optima as possible. These two distinct objectives are achieved using a physical field to influence the swarm behavior. Figure 3 shows an overview of the method. To start with, it uses the same loading and boundary conditions in the TO problem to generate the physical field through a stress analysis. The direction of this field is defined by the principal directions where the normal stress vector is maximized. Then, it inputs the physical field along with other parameters to the swarming process, and it converts the motion paths of the swarm into a material distribution, satisfying the prescribed volume fraction. Next, it randomly generates various parameters within certain ranges, and each set of parameters results in different material distributions. Last, these material distributions serve as the inputs for TO, and it evaluates the optimized results for design inspiration (the least similar ones) or design optimization (the least compliant ones).

The key component is the form-finding with the help of the physical field. It will be presented first, followed by the generative parameters and the rasterization process.

3.1 Form-Finding

Inspired by nature like ant colonies and bird flocking, swarm intelligence is the global collective behavior where the agents interact with each other and their environment locally without a centralized control. Each agent can be seen as a vehicle moving within the design-domain, and they have their own velocity ($\mathbf{v}$) at every time instant. The agents navigate by some basic rules, which are mathematically represented in steering forces. The net steering force is the sum of all forces, i.e.,

$$\mathbf{F} = \sum_k w_k \mathbf{F_k}, \tag{1}$$

where w_k are the weights to balance the importance of forces. The net force is applied to an agent (i) producing a proportional acceleration ($\mathbf{a}$), which changes the agent's velocity:

$$i.\mathbf{v} \leftarrow i.\mathbf{v} + \mathbf{a}, \qquad \text{with } \mathbf{F} = m\mathbf{a}. \tag{2}$$

The mass (m) of an agent in this context is considered non-essential and set to 1. Each agent has its own trajectory, and the aggregation shows a global intelligent pattern, which will be converted to a material distribution.

In the literature, the rules come from the natural principles. For example, the agents would like to move in a similar direction (alignment) and remain close to each other (cohesion) but avoid collisions (separation). This paper also

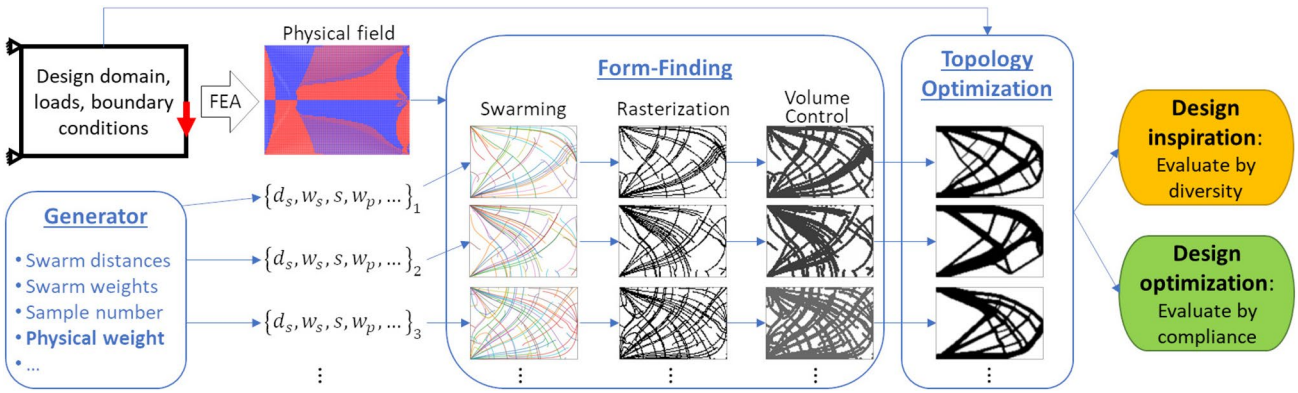

Fig. 3 Overview of the present method. The same conditions for topology optimization are used to generate a physical field, which alongside with other swarming parameters is inputted to swarm intelligence for form-finding. The generated forms are converted to material distributions, which are then optimized for the results. Depending on the metrics evaluating the results, design inspiration and design optimization can be done

introduces a principal direction rule, which is defined on top of a physical field. According to the current positions ($\mathbf{p}$) and velocities ($\mathbf{v}$) of the agents, each rule applies a force to change their motion path, as shown in Fig. 4. The following lists the mathematical formulations of the forces, which are tailored with the consideration of structural stiffness.

Alignment:

$$\mathbf{F_a} = \left(\frac{1}{|\ N_a(i)\ |} \sum_{n \in N_a(i)} n.\mathbf{v} \right) - i.\mathbf{v} \tag{3}$$

with $N_a(i) = \{\ n\ \ |\ \ \|n.\mathbf{p} - i.\mathbf{p}\| \leq d_a\ \ \text{and}\ \ \|i.\mathbf{v}\| \cdot \|n.\mathbf{v}\| > \epsilon\}$

Cohesion:

$$\mathbf{F_c} = \left(\frac{1}{|\ N_c(i)\ |} \sum_{n \in N_c(i)} n.\mathbf{p} \right) - i.\mathbf{p} - i.\mathbf{v} \tag{4}$$

with $N_c(i) = \{\ n\ \ |\ \ \|n.\mathbf{p} - i.\mathbf{p}\| \leq d_c\ \ \text{and}\ \ \|i.\mathbf{v}\| \cdot \|n.\mathbf{v}\| > \epsilon\}$

Separation:

$$\mathbf{F_s} = \left(\frac{1}{|\ N_s(i)\ |} \sum_{n \in N_s(i)} (i.\mathbf{p} - n.\mathbf{p}) \right) - i.\mathbf{v} \tag{5}$$

with $N_s(i) = \{\ n\ \ |\ \ \|n.\mathbf{p} - i.\mathbf{p}\| \leq d_s\}$

Principal direction:

$$\mathbf{F_p} = \frac{ps(i.\mathbf{p})}{ps_{max}} pd(i.\mathbf{p}, i.\mathbf{v}) - i.\mathbf{v} \tag{6}$$

The major difference from the literature is the definition of neighborhood. This paper also considers their instantaneous motion, besides they need to be close to each other ($\|n.\mathbf{p} - i.\mathbf{p}\| \leq d_{a/c/s}$). Continuum mechanics shows that changes in a normal stress do not affect the other orthogonal ones. Therefore, when applying the rules of alignment and cohesion, it does not consider the agents that are moving in the orthogonal or opposite direction as neighbors, i.e., the dot product of their velocity vectors ($\|i.\mathbf{v}\| \cdot \|n.\mathbf{v}\|$) needs to be greater than a threshold, e.g., $\epsilon = 0.2$. For the separation rule, it moves an agent away from its immediate nearby neighbors, so the neighborhood distance of separation (d_s) should be smaller than that of alignment and cohesion (d_a, d_c). In addition, the principal direction rule urges the

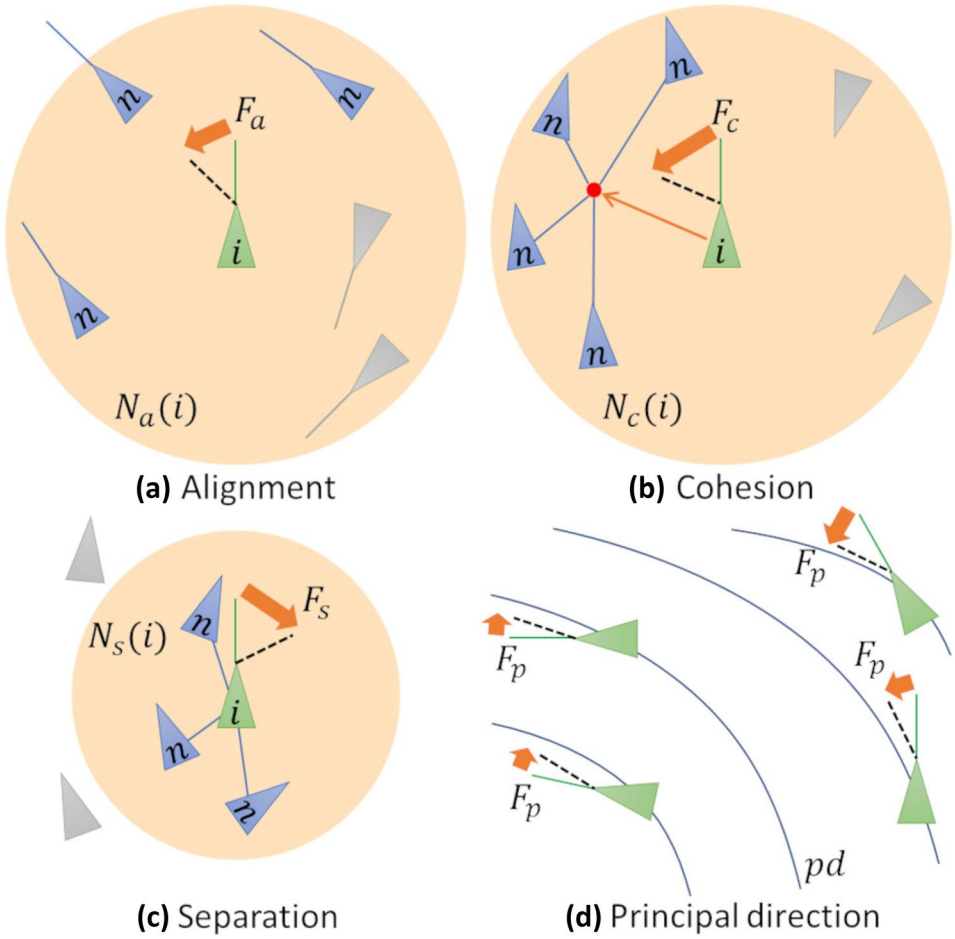

Fig. 4 Four rules of local behavior: **a** alignment, **b** cohesion, **c** separation, and **d** principal direction

agents to follow the physical field, i.e., the principal directions (*pd*). The magnitude of force depends on the value of the principal stresses (*ps*) at the agent's location. ps_{max} is the maximum value of principal stresses throughout the domain. The following presents the method of computing the principal stresses and directions in real-time.

3.1.1 Field of principal direction

Without loss of generality, this work assumes the design-domain to be a two-dimensional (2D) quadrilateral finite mesh, and the loads and supports apply to the nodes of the mesh in a form of nodal forces and nodal boundary conditions. It can then perform a stress analysis by solving the linear equation system: $\mathbf{f} = \mathbf{K}\mathbf{u}$, where $\mathbf{f}$ is the global nodal force vector, $\mathbf{K}$ is the global stiffness matrix based on the Hooke's law, and $\mathbf{u}$ is the output – the global nodal displacement vector. To compute the principal directions for an agent, it is basically the stress recovery process for the element (e) where the agent is located. Given the global displacement vector, it can get the element displacement vector ($\mathbf{u}_e \subset \mathbf{u}$). As a quadrilateral has four vertices and each has two degrees-of-freedom (DoF), $\mathbf{u}_e$ is a 8×1 vector. The Hooke's law for a plane stress problem is as follows:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \mathbf{B}_e(x,y)\mathbf{u}_e \tag{7}$$

where σ_1 and σ_1 are the normal stresses, τ is the shear stress, E is the Young's modulus, ν is the Poisson's ratio, and $\mathbf{B}_e$ is the strain-displacement matrix, i.e., $[\varepsilon_1\ \varepsilon_2\ \gamma]^\top = \mathbf{B}_e(x,y)\mathbf{u}_e$. Mapping the element to a unit square with the bottom-left corner at the origin, $\mathbf{B}_e$ is a 3×8 matrix as follows:

$$\mathbf{B}_e(x,y) = \begin{bmatrix} y-1 & 0 & 1-y & 0 & y & 0 & -y & 0 \\ 0 & x-1 & 0 & -x & 0 & x & 0 & 1-x \\ x-1 & y-1 & -x & 1-y & x & y & 1-x & -y \end{bmatrix}$$

where (x, y) is the coordinate of any location inside the unit square, i.e., $0 \le x, y \le 1$. Then, it can get the stresses at the agent's location via Eq. 7 by mapping $i.\mathbf{p}$ to the local coordinate (x, y) of the element where the agent is located.

The principal direction is the orientation at which the shear stress is zero by rotating the stress tensor through an angle (θ_p). The only stresses at this orientation are the normal stresses, which are known as the principal stresses. The angle (anti-clockwise) can be found based on the original stresses:

$$\theta_p = \frac{1}{2}\tan^{-1}\left(\frac{2\tau}{\sigma_1 - \sigma_2}\right).$$

Thus, one of the two principal directions is at the angle θ_p and the other one is at the angle $\theta_p + \frac{\pi}{2}$, i.e.,

$$\overrightarrow{\theta_{p1}} = [\cos\theta_p,\ \sin\theta_p], \qquad \overrightarrow{\theta_{p2}} = [-\sin\theta_p,\ \cos\theta_p].$$

Correspondingly, the two principal stresses are computed via the transformation equations:

$$\sigma_{p1} = \sigma_1\cos^2\theta_p + \sigma_2\sin^2\theta_p + 2\tau\cos\theta_p\sin\theta_p$$
$$\sigma_{p2} = \sigma_1\sin^2\theta_p + \sigma_2\cos^2\theta_p - 2\tau\cos\theta_p\sin\theta_p$$

The only exception is when $\sigma_1 - \sigma_2 = 0$, there are no principal directions, and the stresses in all directions are the same. Here, the principal direction rule does not result in any steering force, i.e., $F_p = 0$. Otherwise, *pd*() returns the principal direction that is the closest to the agent's moving direction and *ps*() returns the corresponding principal stress:

$$pd(i.\mathbf{p}, i.\mathbf{v}) = \begin{cases} \mathrm{sgn}(i.\mathbf{v}\cdot\overrightarrow{\theta_{p1}})\,\overrightarrow{\theta_{p1}}, & \text{if } |\, i.\mathbf{v}\cdot\overrightarrow{\theta_{p1}}\,| \ge |\, i.\mathbf{v}\cdot\overrightarrow{\theta_{p2}}\,| \\ \mathrm{sgn}(i.\mathbf{v}\cdot\overrightarrow{\theta_{p2}})\,\overrightarrow{\theta_{p2}}, & \text{else.} \end{cases}$$

$$ps(i.\mathbf{p}) = \begin{cases} |\,\sigma_{p1}\,|, & \text{if } |\, i.\mathbf{v}\cdot\overrightarrow{\theta_{p1}}\,| \ge |\, i.\mathbf{v}\cdot\overrightarrow{\theta_{p2}}\,| \\ |\,\sigma_{p2}\,|, & \text{else.} \end{cases}$$

Let it be noted that $\overrightarrow{\theta_{p1}}$, $\overrightarrow{\theta_{p2}}$, σ_{p1}, and σ_{p2} are calculated based on $i.\mathbf{p}$. This is done in real-time, and the only required input is the global displacement vector $\mathbf{u}$.

3.1.2 Generative parameters

GD creates and explores solutions based on a set of input parameters, so it is important to set up these parameters properly. From the above, the variables in the form-finding process include the weights among the steering forces (alignment: w_a, cohesion: w_c, separation: w_s, principal direction: w_p), the distances defining an agent's neighborhood (d_a, d_c, d_s), the number of agents (m), and their initial position and velocity ($\{\mathbf{p}\}, \{\mathbf{v}\}$). In addition, there are two maximum bounding values: one on the speed and one on the steering force. The maximum speed controls how far an agent can move in one step. To balance the piecewise linearity of the trajectory and the computation time, it has the size of a quadrilateral (q), i.e., $s_{max} = q$, and $q = 1.0$ in this paper. This is applied to Eq. 2: whenever $|\,\mathbf{v}\,| > s_{max}$, $\mathbf{v}$ is scaled by $s_{max}/|\,\mathbf{v}\,|$. The maximum steering force limits the acceleration of an agent and thus prevents sudden turns or potential oscillations. To have a smooth trajectory, a steering force can change the direction of an agent at most about 0.2 rad (weak constraint). If the average speed of an agent is 0.6, it can calculate the maximum force via the arc length formula, i.e., $f_{max} = 0.6 \times 0.2 \approx 0.1$. This applies to the total

force in Eq. 1 and each of the forces in Eq. 3–6. Whenever the magnitude of a force is greater than f_{max}, the limit scales it down to f_{max}.

Some variables are fully random, like the initial location (**p**) and velocity (**v**). The agents are spread freely throughout the design-domain and take an arbitrary direction to start with in each generation. The relative weights of the steering forces (w_a, w_c, w_s, w_p) are random too, as long as none of them overwhelm the others and they sum to unity. Therefore, they are given a value between 0.5 and 1.0, and their sum is normalized to 1.0. Generally speaking, high alignment weight (w_a) gives more parallel struts; high cohesion weight (w_c) leads to more main struts; high separation weight (w_s) introduces more branches; and high physical weight (w_p) results in principal stress lines. The neighborhood distances (d_a, d_c, d_s) should be related to the domain size such that the collective behavior can be observed. If the distances are too small, the agents are independent of each other – creating many branches in the design; but if the distances are too large, all agents are interrelated and there will be only one global behavior – resulting in a few main struts in the design. In addition, the separation distance should be smaller than the other two, as mentioned before. They have a value less than 10% of the size (L) of the design-domain, i.e., the distances of alignment and cohesion (d_a, d_c) have a floating-point value between $0.06L$ to $0.1L$, and the separation distance (d_s) has a value between $0.02L$ to $0.06L$. To account for irregular shapes of the design-domain, the size L is the square root of the domain's area, e.g., for a 120×80 rectangular domain, $L = \sqrt{120 \times 80} \approx 100$. The number of agents (m) controls the topological complexity of the initial shape, i.e., the more agents, the more branches. Although increasing the complexity can explore more interesting results, too many agents will lead to a high volume. It should relate to the size of the design-domain and the volume fraction (vf), but since the length of each agent's trajectory is unpredictable, it prefers a large enough range that can produce different volumes. This paper sets m to be between $(vf)L$ and L. Table 1 summarizes the values/ranges for each variable, and Fig. 5 shows the operation flow chart of the algorithm for a set of generative parameters.

Table 1 List of variables and generative parameters

Variable	Value/Range	Note
L, q, vf	$\sim 100, 1.0, 0.4$	Problem defined
s_{max}	q	Speed limit
f_{max}	$0.1q$	Force limit
w_a, w_c, w_s, w_p	[0.5 1]	Weights, $\sum_k w_k = 1$
d_a, d_c	$[0.06\ 0.1]L$	Neighbor distances
d_s	$[0.02\ 0.06]L$	Neighbor distance
m	$[vf\ 1]L$	Number of agents

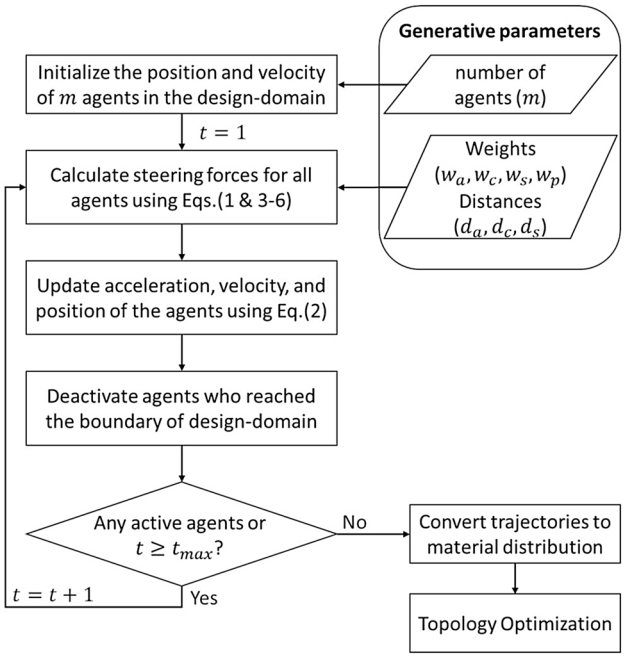

Fig. 5 An operation flow chart of generating one material distribution by swarming

The termination condition for the motion of the agents is when they reach the boundary of the design-domain, and then they become inactive. After all agents are no longer active, the present method collects and converts their trajectories to the initial shapes for optimization, which is detailed in the following.

3.2 Rasterization and volume control

The TO tool used in this paper is the SIMP method (Andreassen et al. 2011), which computes the optimal material density for each quadrilateral element. Therefore, the trajectories of agents need to be converted to the material density in the elements (see Fig. 3). In 2D cases, it is like the rasterization process that turns curves into pixels, where the trajectories are the curves and the quadrilateral mesh is an image. By checking which elements the agents have passed by, it assigns these elements with solid materials, while the rest empty. This material distribution has an arbitrary volume, and it is most likely not the same as the desired volume fraction. Although the volume difference is not a problem for the TO method, as the process can add or remove materials automatically, the difference cannot be large. Especially, if the initial volume was much lower than the desired volume, the optimization would evenly put materials in the whole domain, which would lessen the influence of the initial shape to the result and thus lower the diversity. Here, this method applies a dilation operation repeatedly to the rasterized image until

the volume is equal to or larger than the required volume. When the input volume is higher than desired, the method proportionally scales the density of all elements down to reduce it.

3.3 Design inspiration and optimization

As mentioned above, GD is used in the early phase of design to give initial design possibilities for automated ideation, and thus its major goal is to generate multiple design suggestions, allowing the designers and engineers to try out quickly different designs and make changes. This is design inspiration, and the present framework lists out the most diverse designs for the purpose. To measure the diversity of the generated results, it quantifies the similarity between two structures (ρ, ϱ) in terms of a distance measure that is defined on a high-dimensional feature space, i.e., each quadrilateral element is a dimension, using the Frobenius norm:

$$\delta(\rho, \varrho) = \|\rho - \varrho\|_F = \sqrt{\sum_e \mid \rho_e - \varrho_e \mid^2} \tag{8}$$

where ρ and ϱ are in a form of density vector. Diversity is a measure of how an individual design differs from the others. Therefore, while sorting the designs by diversity, each successor needs to compare with all its predecessors. Searching for the next most diverse design is to find the one that has the maximum cumulative distance from all the designs in the sorted list (A), i.e.,

$$\arg\max_\rho \ : \ \zeta(\rho) = \sum_{\varrho \in A} \delta(\rho, \varrho)$$

The concept is like from a set of 3D points, finding the largest triangle for an edge, then the largest tetrahedron for the triangle, and so forth. The sorting gives the ordering of diversity according to the quantitative point of view, and users can choose the inspiring ones from the top of the sorted list. The first design on the list is the result generated by the original SIMP method.

The magnitude of this distance measure (δ) is different per problem. It depends on the dimension of the design-domain and the volume fraction (vf). To make it comparable among different problems, this paper normalizes the value and reports it as a percentage via dividing by the possible maximum difference. For example, two material distributions on a 120×80 design-domain with $vf = 0.4$ are the most different from each other when none of their materials share an element, meaning that there are $120 \times 80 \times 0.8$ distinct elements between them. Using the Frobenius norm, the possible maximum difference is the square root of the number of distinct elements (each has a difference of 1), and the normalized value is

$$d(\rho, \varrho) = \frac{\delta(\rho, \varrho)}{\sqrt{A \cdot vf}} \times 100\% \tag{9}$$

where A is the area of the design-domain. Although this measurement may not be the same as the perceptual difference and their relationship is probably neither linear nor proportional, the larger the value, the more likely to be perceptually different.

GD can also find the best option based on certain metrics for design optimization. For example, it can use the same goal as in TO to find the stiffest structure. Recall that the objective of a common TO problem is to minimize the compliance of the structure, and its mathematical formulation reads as:

$$\begin{aligned} \arg\min_\rho \ & c(\rho) = \mathbf{u}^\top \mathbf{K}\mathbf{u} = \textstyle\sum_e E(\rho_e)\mathbf{u}_e^\top \mathbf{k}_e \mathbf{u}_e \\ s.t. \ & \mathbf{f} = \mathbf{K}\mathbf{u}, \\ & 0 \le \rho_e \le 1, \\ & \textstyle\sum_e \rho_e = V. \end{aligned} \tag{10}$$

where V is the desired volume, and $E(\rho_e)$ returns the value of Young's modulus based on the density of the element, e.g., $E(\rho_e) = (\rho_e)^p E_0$, with E_0 being the Young's modulus of the material and p being a penalization power to favor binary outputs. Each generated design has a compliance value $c(\rho)$, and sorting them in an ascending order, the first one is the best design for the problem.

4 Results

The common practice of GD uses the power of cloud computing to generate many shapes at once, i.e., one set of generative parameters corresponds to one design, and there is no interrelationship between the designs. The generator can create distinct sets of parameters and pass them to independent processors to compute the designs in parallel. Therefore, generating multiple designs is essentially taking the same time as generating one design plus some overheads.

The present method is implemented in Python, and the Python version of SIMP method (Andreassen et al. 2011) for TO is used. It is tested on a PC running 64-bit Windows 10 equipped with Intel Core i5-6500 CPU@3.20GHz, and 8GB RAM. The swarming step takes only 3 seconds on a 120×80 design-domain, and TO takes 128 seconds on average, which is the bottleneck. The contribution of this paper is introducing physical meaning into the swarming process but not developing new TO methods, and this work only uses publicly available codes for the TO. If a more efficient TO solver (Gao et al. 2021) is used, it can speed up the entire process.

As a proof-of-concept, this paper focuses on studying the diversity of the optimized designs and does not implement

cloud computing or 3D examples, but it is not hard to estimate the performance in those situations. With the overhead time, using cloud or parallel computing would take about 140 seconds to generate all designs. For 3D cases, the swarming time increases proportionally with the size of domain because the agents need to travel longer, but the computational complexity is the same. The bottleneck is still the TO step, e.g., a high-performance solver (Wu et al. 2016) takes 146 seconds on a $200 \times 100 \times 100$ design domain.

The following will first validate the use of principal direction and then evaluate the performance of the present generative method in improving diversity. It will also study a couple of test examples to show the capability of the framework, and all are on macroscale. Unless otherwise stated, all experiments use the volume fraction 0.4, the penalization power 3.0, and the density filtering with the filter radius 1.5 (by the element size), which controls the minimum size of the features in the optimized design to make sure it is printable by additive manufacturing. The Young's modulus of the material uses $E = 1$ Pa, and each applied force has a magnitude of 1 N. This paper only generates 100 designs for each example, but the present method can generate far more designs than that. One can always generate more and more designs to improve diversity, but using a relatively small number here can show the effectiveness of the present generative method in creating diverse results.

4.1 Validation of principal direction

Part of the hypothesis in this paper is that using principal direction as a steering force considers optimality and helps get close to different local optima. To check its validity, the present method first applies only the steering force of the principal direction rule without randomness. This experiment is on the symmetric cantilever beam problem as shown in Fig. 6a. The design-domain is a 120×80 rectangle, with the top-left and bottom-left corners fixed and a downward force at the middle of the right edge. This problem has an analytic optimum, which is shown in Fig. 6b. Without randomness, the agents need to be initialized meaningfully in this experiment. The analytic optimum shows that all the curves of the structure connect to one of the fixed supports. Therefore, this experiment also places the agents at the location of the fixed supports and assigns their directions at every $\pi/36$ rad. Since the support points are at the corners (right angle), there are in total 34 valid agents. All of them come out from the supports and end at the domain boundary. The trajectories of the agents are shown in Fig. 6c. From the result, here makes two observations. First, given the symmetric nature of the problem and initialization, the trajectories are also symmetric. This reveals that the method is reliable and repeatable, and using the rule of principal direction is a valid add-on to the swarming process. Second, the trajectories have a similar shape to the analytic optimum. In fact, the curves in the analytic optimum are a subset of the curves in the trajectories. Although they are not the same, this can already disclose that the principal directions indeed relate to optimality, and including this rule is a rational cho ice.

4.2 Effectiveness of inspiration

As the main purpose of this paper is to improve the diversity of designs generated, this section looks more closely into the generative process to evaluate its effectiveness. This paper studies the diversity by two trends of data plotted against each generation as shown in Fig. 7, which compares the performance between (a) random material distribution and (b) the present generative method. The data come from the asymmetric cantilever beam problem used in Fig. 2. The first trend (dotted curve) shows the maximum difference between all generated designs, including

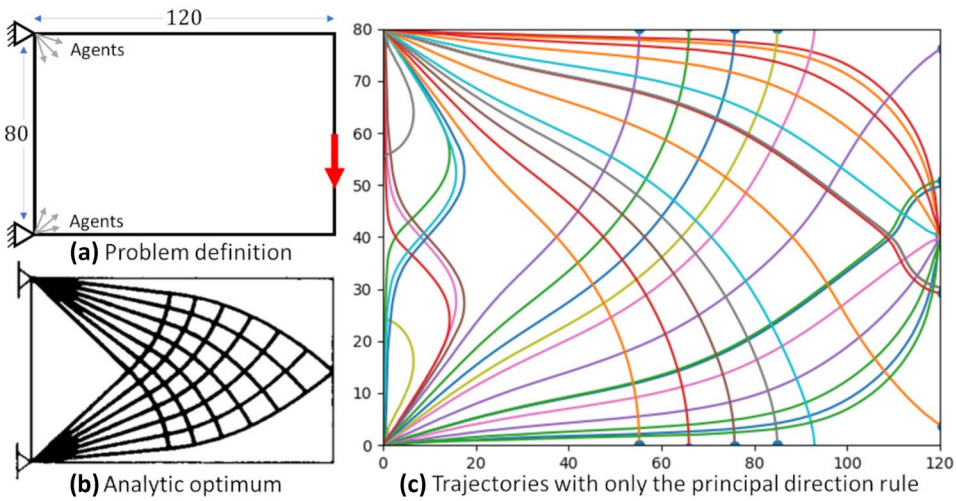

Fig. 6 Applying only the steering force of the principal direction rule. **a** The problem definition and the agents are initialized at the location of fixed supports. **b** The analytic optimum for the cantilever beam problem. **c** The trajectories of the agents following only the principal direction

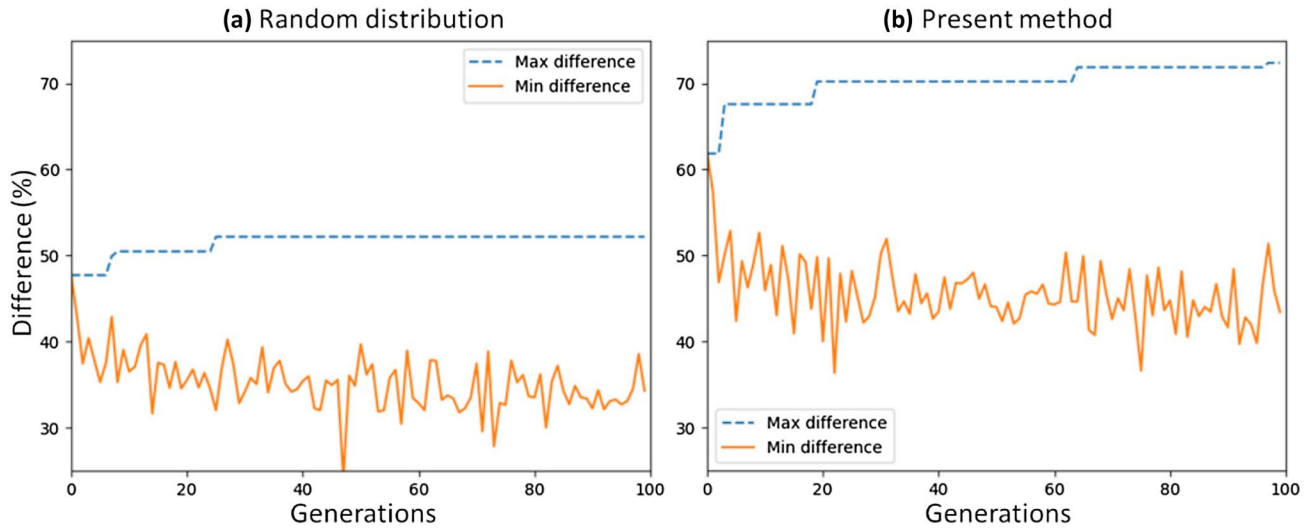

Fig. 7 Line plots of differences against generations created by **a** random material distribution and **b** the present method

the latest one, e.g., the generation #40 considers all the designs from 1 to 40 and the SIMP one. Its mathematical formulation is

$$f_1(x_i) = \max_{j,k} d(x_j, x_k), \text{ s.t. } j,k \leq i. \tag{11}$$

Whenever there is a rise in this trend, it means that the newly generated design has increased the overall diversity because of the extended range of difference. The second trend (solid curve) represents the minimum difference between the latest design and all the previous designs, i.e.,

$$f_2(x_i) = \min_{j} d(x_i, x_j), \text{ s.t. } j < i. \tag{12}$$

It measures how much difference comparing the new one with its fellow designs. If the value of a design is zero, it is the same as one of the previously generated designs. The larger the value for each design, the more varieties there are in the results. By analogy with finding a triangle from a set of points, the maximum and minimum differences (f_1,f_2) are like the longest and the shortest lengths of the triangle. If both lengths are large, the size of the triangle is large.

Overall, the differences of the designs generated by the present method are much higher than that of the random distribution, and in fact the maximum difference of the random distribution is only at a similar level to the present method's minimum difference. For the random distribution, the first generation has a 47% difference from the SIMP result. The maximum difference has increased to 50% at the generation #9 and reaches a plateau (52%) at the generation #26. Although the maximum difference settles in the early generations, the minimum difference maintains a level of 30–40% throughout all the generations. This reveals that random inputs can indeed generate various designs, but there is not much further inspiration after the generation #26. In the present method, the first generation already has a 62% difference comparing with the SIMP result, which is even greater than the highest from the random distribution. Its maximum difference keeps increasing the whole time and achieves 72% at the generation #98. This is a 40% improvement in diversity compared with the random distribution. The minimum difference is also higher and around 40–55% in the majority. Figure 8 shows the four most diverse designs sorted by the method detailed in Sec. 3.3. They are obviously

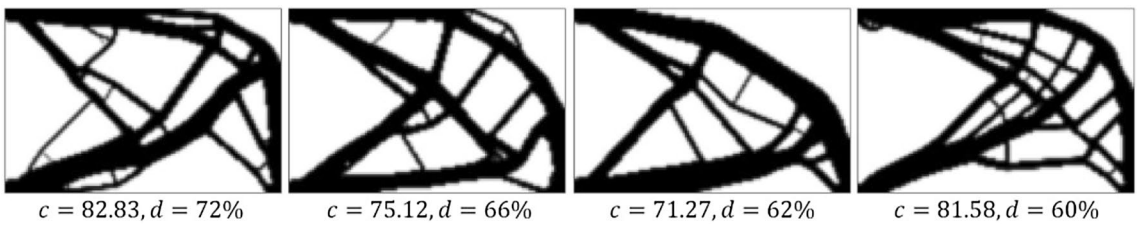

Fig. 8 Top four inspiring designs generated by the present method for the asymmetric cantilever problem shown in Fig. 2. c is the compliance and d is the percentage difference from the original SIMP result

different from the results generated by the random distribution as shown in Fig. 2. As a result, the present generative method can indeed explore more diverse designs efficiently and effectively by swarming.

Here also makes a comparison with another generative algorithm developed by Sun and Ma (2020). They applied exploration algorithms of reinforcement learning (RL) to TO problems. One method is to use the upper confidence bound to change the sensitivity analysis of the SIMP method for obtaining different results. The optimized result got by the original SIMP method has four thick struts on the perimeter and four thin struts between the thick ones (see Fig. 9). Although it does not have the same complexity as the analytic optimum in Fig. 6, it is the optimal solution under the domain resolution and settings, and it has a compliance of 58.62. Both methods can also generate a variety of designs, and Fig. 9 shows six of them. While the designs generated by the RL-based method (Sun and Ma 2020) still look like the original SIMP result in terms of the overall shape and complexity, the present method has created plenty of interesting features, many of which have never been seen in the literature. Although the designs may look unusual, they are indeed the final optimized designs, and they have similar compliance to those generated by the RL-based method. The compliance values are mostly between 60 and 65, and with this around 10% increase only, users can select their preference from a bunch of inspiring designs. The maximum difference (f_1) among all 100 generated designs is 68%. One may also notice that the results from the RL-based method are always symmetric. This is because that the sensitivity only changes the way of updating the solution, but a symmetric problem will always result in a symmetric design. In contrast, the present method inputs different material distributions, and thus it has more control of the results. If symmetry is necessary, one can apply symmetry boundary conditions to half of the design-domain or make a mirror copy of the inputs, and then the outputs will be symmetry. A shows another comparison between the generative methods on the asymmetric cantilever problem.

4.3 Design optimization

In the previous examples, the compliance value is always higher when the generated designs differ from the result produced by the original SIMP method. This is because the previous cases are relatively simple, e.g., the design-domain is just a rectangle, and thus a uniform material distribution is a good initial guess to reach the global optimum. To show the capability of the present method in finding better solutions, this section tests a slightly more complex situation – a concave design-domain shaped like the letter 'L'. The concavity comes from assigning 60×60 passive elements in the top-right of a 100×100 domain, as shown in Fig. 10. It fixes the two corners at the top of 'L' and applies a downward force at the middle of the bottom-right end. Under this configuration, the result produced by the original SIMP method has a compliance value of $c = 263.75$ according to Eq. 10.

The compliance values of 100 designs generated by the present method are plotted as a line chart in Fig. 10 too. By drawing a horizontal line at the value of 263.75, it finds three designs having a lower compliance value than the original SIMP result. The lowest one is 257.60 at the generation #34, which is visualized in Fig. 10 alongside with other two from the generations #26 and #64. Although it is not a vast improvement, this shows that the original SIMP method does not always get the optimal result, and the present method can find better solutions. In terms of diversity, the three results are only about 50% different from the original SIMP result, which is not high, meaning that they share some common features in supporting the loads effectively. The maximum difference (f_1) among all generated designs is 72%.

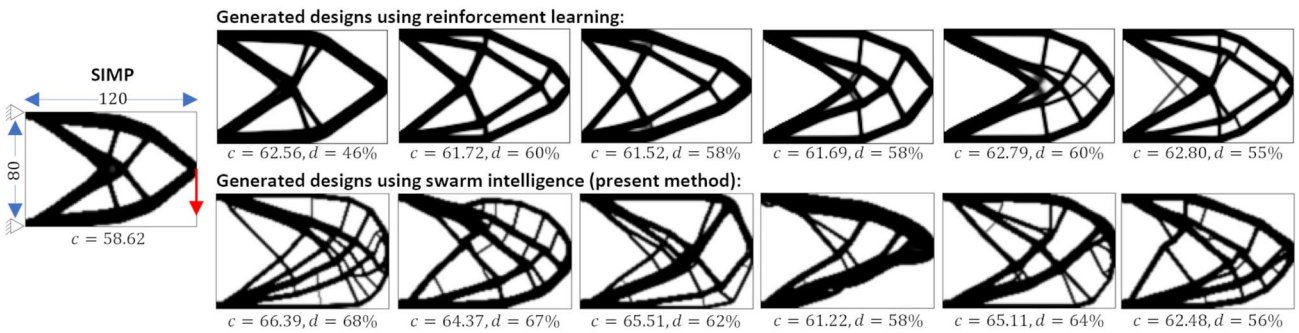

Fig. 9 Comparison of generated designs by (top) reinforcement learning (Sun and Ma 2020) and (bottom) the present method . c is the compliance and d is the percentage difference from the original SIMP result (leftmost)

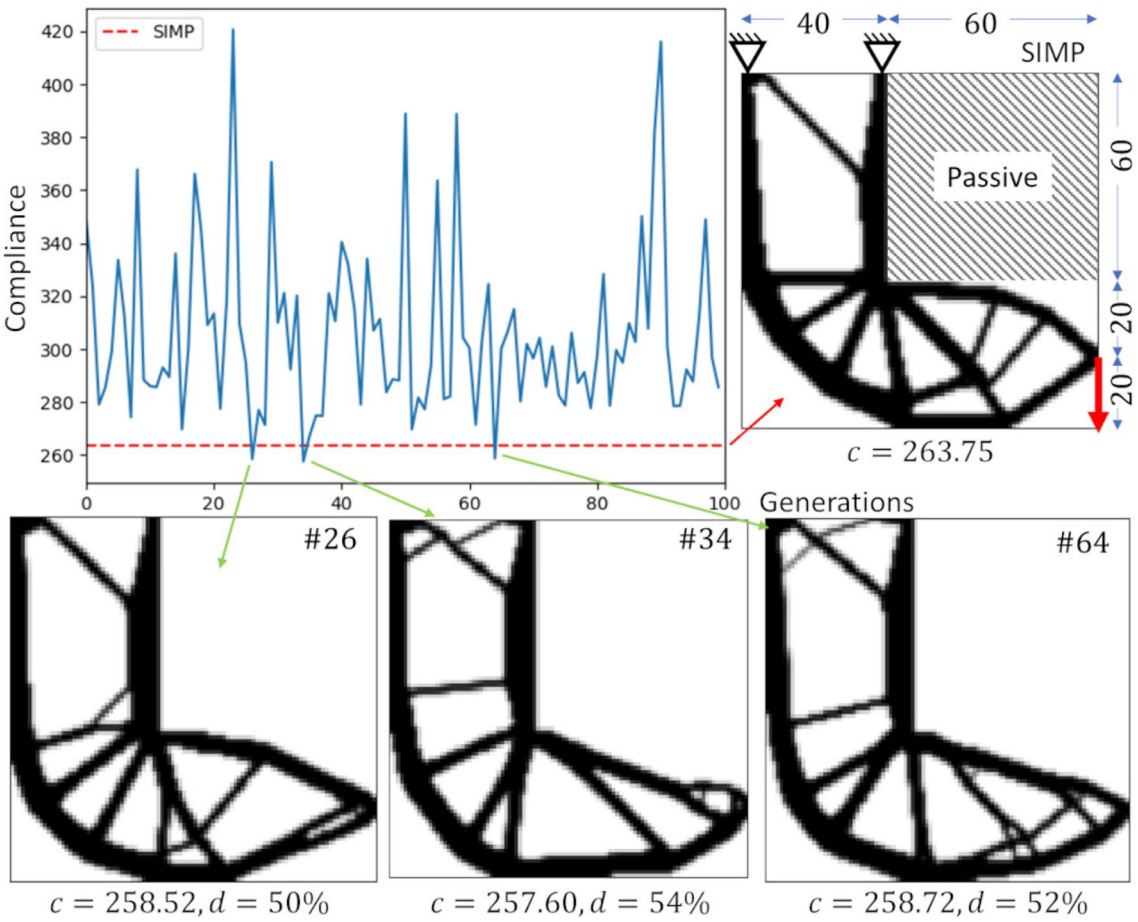

Fig. 10 Design optimization on a L-shape domain. c is the compliance and d is the percentage difference from the original SIMP result (top-right)

4.4 Other examples and statistics

This section tests the present method on other examples, including the symmetric cantilever beam, a bridge, a bicycle, a chair, and a skateboard truck. Figure 11 shows the problem definitions, the original SIMP results, and the top twelve diverse designs generated by the present method. It is worthy to remind that the TO settings (e.g., penalization and filter radius) are the same for all results and the only difference is the input material distribution, which is generated by various swarming conditions. Each generated design is a final optimized design instead of an intermediate one during the optimization iterations. This means that even though they may not be the global optimum, they are indeed the local optima. Table 2 lists the statistics, and it uses two metrics to measure the diversity. One is the maximum difference f_1 from Eq. 11 including all designs (i.e., $i = 100$), and the other one is the minimum difference f_2 from Eq. 12, but since f_2 has one value for each design, it reports the mean value $\overline{f_2}$. The table also lists the best compliance (c_{best}) got in each example.

L-shape Besides the optimized designs shown in Fig. 10, there are also inspiring designs as shown in Fig. 11a. This is a challenging problem because of the eccentric load and the passive region impeding direct connections between the load and the fixed points, and the original SIMP result has a compliance value of $c = 263.8$. Even though the passive elements largely reduce the DOF and the domain is essentially a long rectangle with a small width, the present method can still generate diverse designs visually and quantitatively. The maximum difference (f_1) among all 100 generated designs is 72% and the mean of minimum differences ($\overline{f_2}$) is 48%. However, any designs that do not share similar features with those in Fig. 10 are having a much higher compliance value (up to a 60% increase) in this challenging problem.

Bridge This problem has a 100×50 rectangular design-domain with the bottom-left and bottom-right corners fixed and a force being applied at the middle of the bottom edge (see Fig. 11b). The original SIMP result has a thick arc connecting the two fixed points, and there are four struts linking the load to the middle of the arc. It has a compliance value of $c = 11.99$. The results generated by the present method have

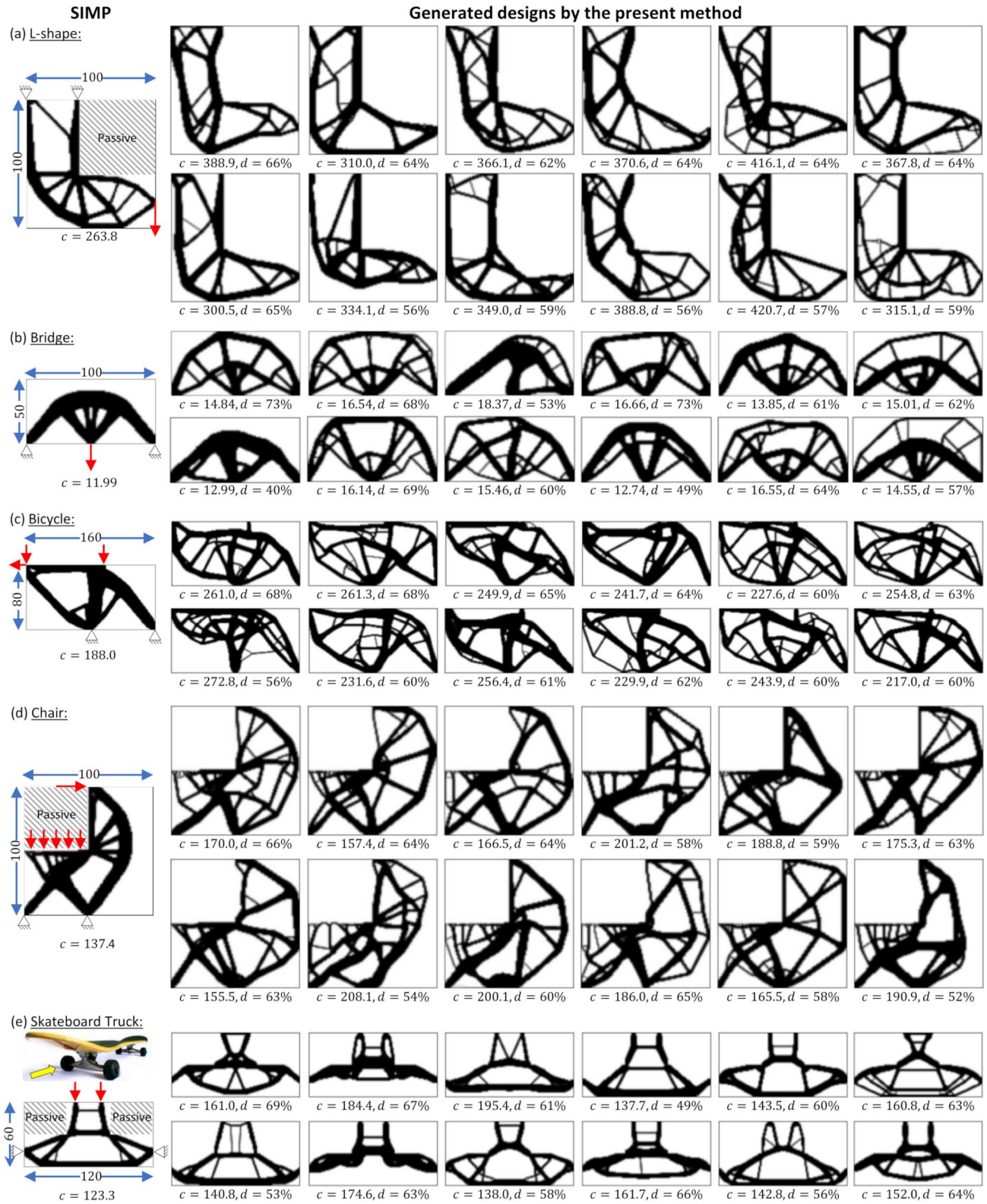

Fig. 11 Design inspiration on various examples. c is the compliance and d is the percentage difference from the original SIMP result (leftmost)

Table 2 Statistics for various examples

Example	Size	f_1	$\overline{f_2}$	c_{best}
Sym. Cantilever	120×80	68%	46%	58.62
Asym. Cantilever	120×80	72%	46%	65.87
L-shape	100×100 - 60×60	72%	48%	257.6*
Bridge	100×50	76%	40%	11.98*
Bicycle	160×80	72%	49%	188.01
Chair	100×100 - 50×50	69%	46%	135.93*
Skateboard truck	120×60 - $2(40 \times 30)$	77%	37%	121.4*

f_1 is the maximum difference among all designs, and $\overline{f_2}$ is the mean of minimum differences from each design. c_{best} is the lowest compliance obtained, and the values with an asterisk * meaning that they are better than the original SIMP result

different topology and shapes: some of them have two arcs, and some look like a spider net. In this example, $f_1 = 76\%$, $\overline{f_2} = 40\%$, and the lowest compliance found among all generated designs is $c_{\text{best}} = 11.98$ – a bit better than the original SIMP result.

Bicycle To mimic the situation where a person sitting on a bicycle frame and holding the front handles, this example uses a 160×80 design-domain with two loads applied: one at the top-middle pointing down and one at the top-left with both x and y directions (see Fig. 11c). It also fixes the domain at two points on the bottom to simulate the centers of the wheels. The original SIMP result is basically a set of straight struts connecting the four load/fixed points. This is neat and rigid ($c = 188.0$) but does not have much sense of aesthetic. In contrast, the generated designs have different styles and interesting shapes. For example, the 3rd design has a smile like that on a comedy drama mask, and the 7th one looks like the supertree from Singapore. The maximum and minimum differences are $f_1 = 72\%$ and $\overline{f_2} = 49\%$.

Chair The design-domain of this test case is a 100×100 square (see Fig. 11d), with a 50×50 passive region defined in the top-left for the sitting space. There are two loads simulating a person sitting and leaning back in the chair: a downward load of 2 N in total distributed uniformly in the seat, and a load applied at the top-middle to the right. The original SIMP result has an organic shape in this example, combining tree branches under the seat and fan shapes in the back, and it has a compliance value of $c = 137.4$. The present method can further improve both diversity and stiffness. The generated chairs have round, triangular, and various polygonal shapes. The differences are $f_1 = 69\%$ and $\overline{f_2} = 46\%$. The lowest compliance got is $c_{\text{best}} = 135.9$.

Skateboard truck A skateboard truck is used to attach the wheels to the skateboard deck, and it is fastened to the underside of the board (see Fig. 11e). The skateboard truck must be durable to withstand high impact landings, yet lightweight to perform ollies and flip tricks in mid-air. Therefore, the skateboard truck should be both optimal and stylish. This example uses a 120×60 design-domain, with two 40×30 passive regions specified in the top-left and top-right corners. Two downward loads apply at the position of fasteners serving as the weight of skaters, and two fixed points are at the wheel positions. The original SIMP result has a ladder structure in the top and a trapezoid with two triangles in the bottom. It has a compliance value of $c = 123.3$. To avoid uneven balancing of weight, the present method enforces a y-axis symmetry at the middle by mirror copying the input material distributions before TO. All the outputs are symmetric, expect the 10th design has a small strut in the middle, causing a slight asymmetry. The results can be regular, like the 4th and 6th designs, and they can be animated, e.g., the 3rd design looks like cat ears, and the 12th resembles a bat. The differences are $f_1 = 77\%$ and $\overline{f_2} = 37\%$, and the lowest compliance is $c_{\text{best}} = 122.9$.

5 Conclusion

This paper presents a new generative method to improve the diversity of topology-optimized designs by capturing as many local optima as possible in the topology optimization (TO) problem. The method is developed based on two observations: (1) topology optimization places materials along the principal directions to maximize the overall stiffness, and (2) swarm intelligence generates forms using the trajectory of agents. This paper applies the rule of principal direction to the swarm intelligence, so that the form-finding process can also consider the structural stiffness and balance randomness and optimality. It varies the swarming parameters to get different input material distributions, which are then optimized to get the final designs. The experiments have used various examples of both convex and concave domains, including the benchmarking cantilever, L-shape, and bridge problems, as well as a bicycle, a chair, and a skateboard truck. The results show that the present method can generate diverse designs that have not been seen in the literature. A means of measuring diversity by calculating their difference is also used to sort the designs, so that users can view the designs that are the most different from each other and select their preference among them. In terms of optimality, the present method can also find a few designs that have lower compliance than the results generated by the original solid isotropic material with penalization (SIMP) method.

Despite the promising results, the present method has a few limitations. To begin with, the current generator is completely random, and it could produce some similar input material distributions, which should be filtered or avoided to have more diverse results. Certain design exploration

methods, like a data-driven approach (Kang et al. 2021), will make the present method more effective. Next, the current measure of diversity is using a pixel-wise comparison, which may not be the same as the perceptual difference. A future work is to develop a more accurate aesthetic measure, e.g., based on user survey. It can then create a recommendation system to predict the user preference. In addition, the current swarming can take structural stiffness into account with the rule of principal direction, but it has not yet considered other structural performance, like strength, stability, and failure mode, and manufacturing criteria. Future work should explore other rules for the swarm intelligence to consider various structural performance and free-of-supports for additive manufacturing. Finally, this paper only shows that the present method can work with the SIMP method, but the trajectories need not to be converted to pixels only. Extending it to other TO methods will be another future work.

Appendix A 100 results

This section lists all the 100 results generated by three different methods on the asymmetric cantilever problems. The results from random material distributions are shown in Fig. 12, by the reinforcement learning (Sun and Ma 2020) is shown in Fig. 13, and by the present method (swarm intelligence) is shown in Fig. 14. It can be seen that the present method has much more diverse designs.

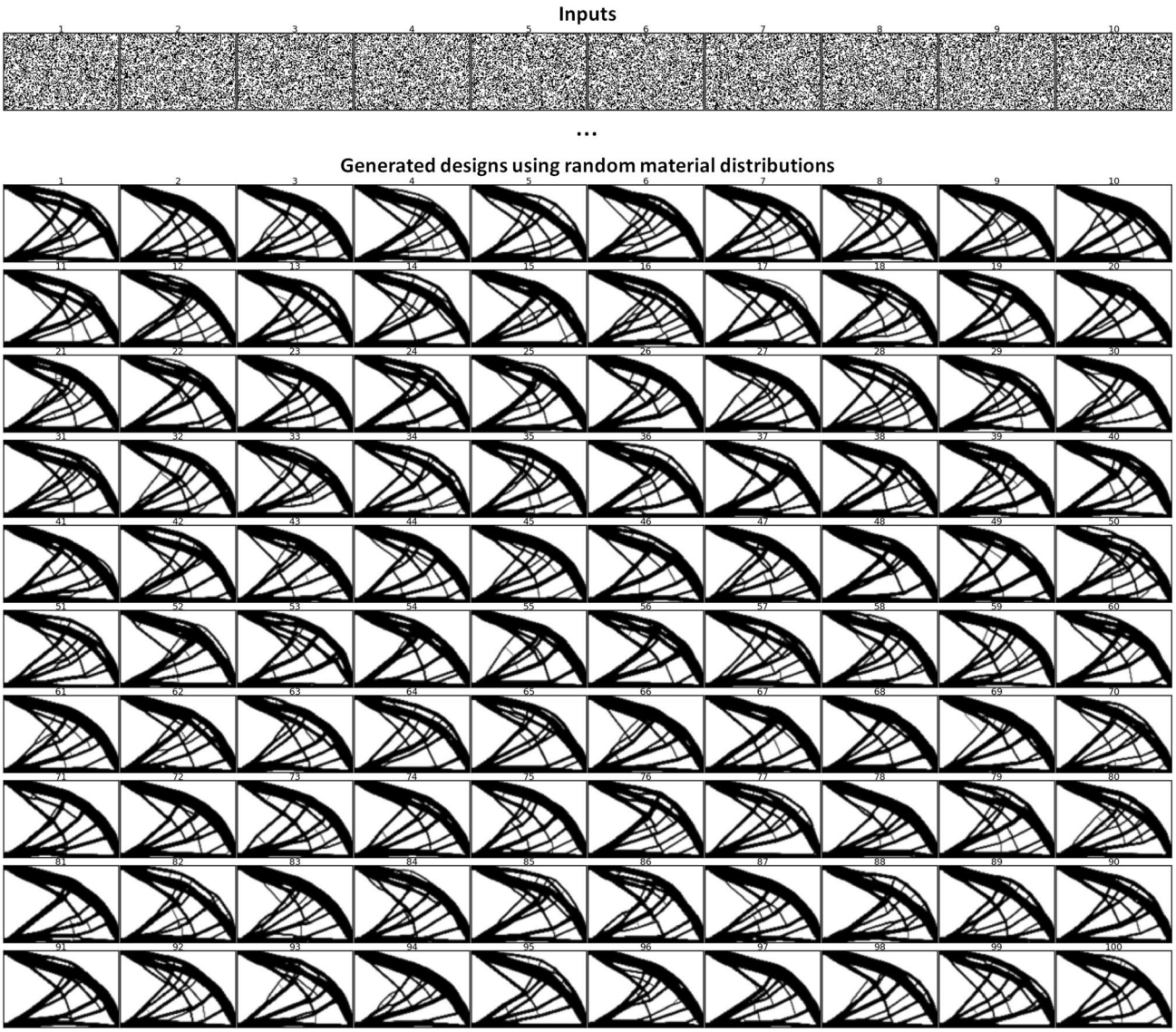

Fig. 12 100 results from random material distributions. 10 of the inputs are shown in the top

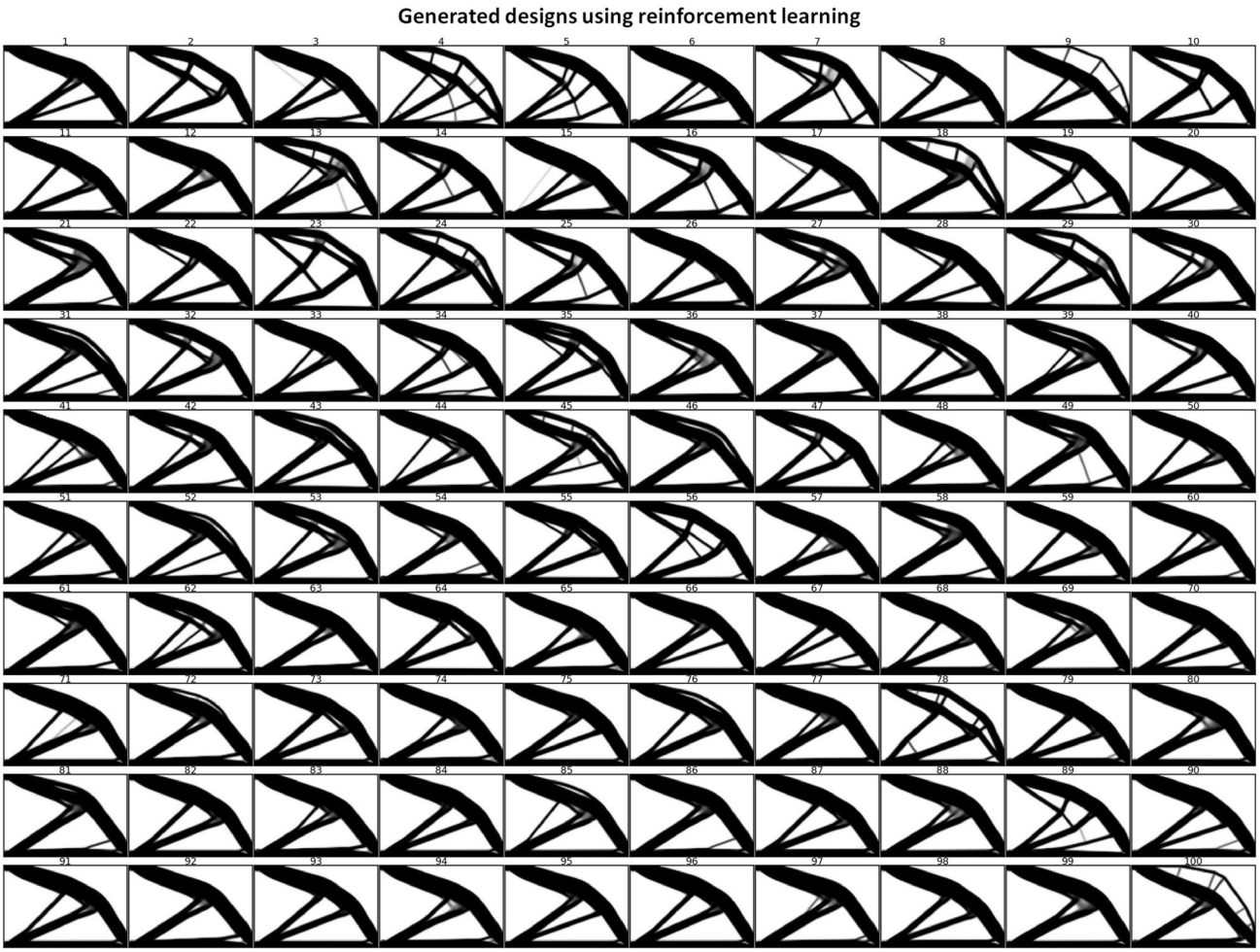

Fig. 13 100 results generated by the reinforcement learning method (SIMP_UCB) (Sun and Ma 2020)

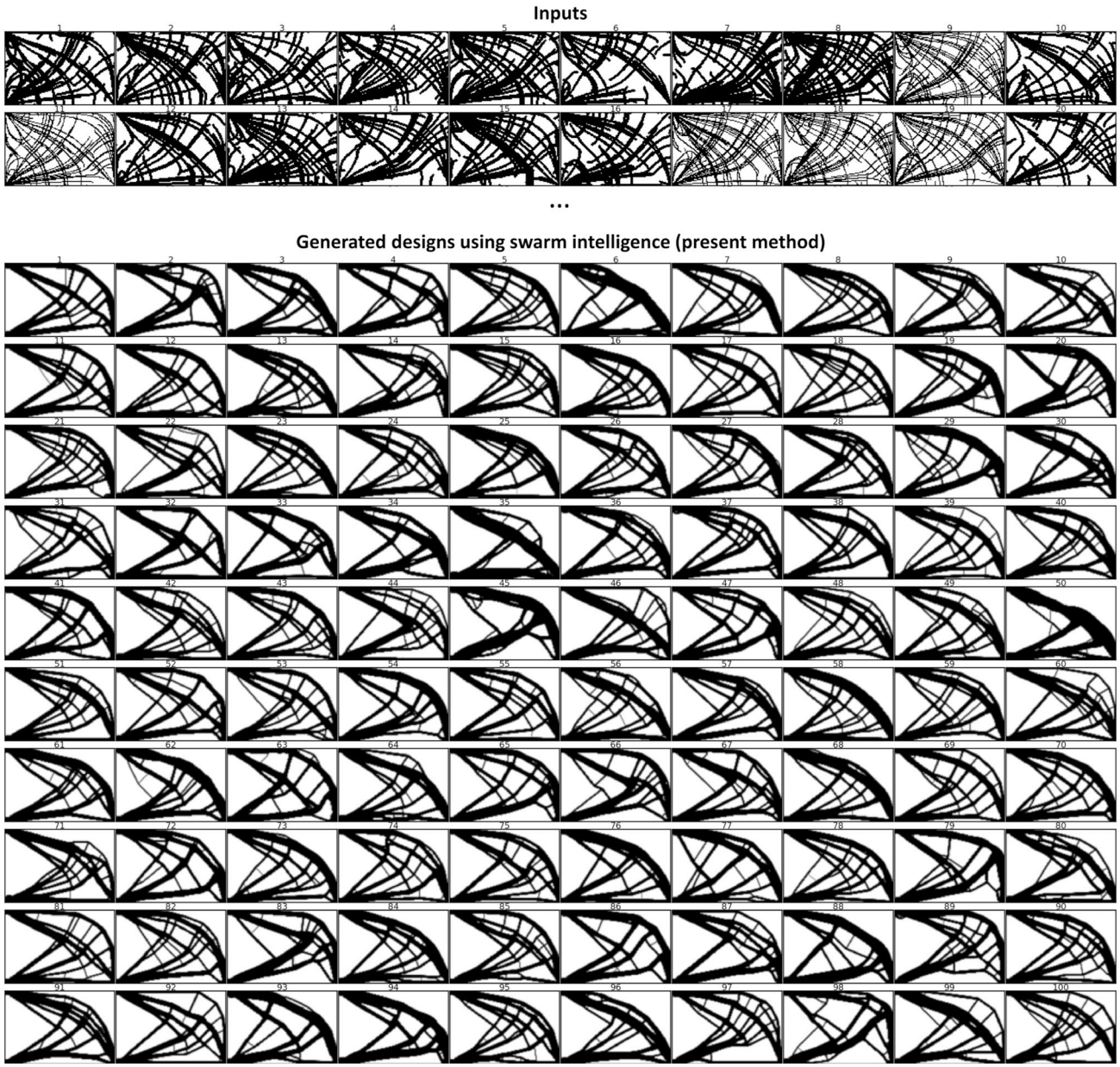

Fig. 14 100 results generated by the present method. 20 of the inputs are shown in the top

Acknowledgements This paper acknowledges the support of the Natural Sciences & Engineering Research Council of Canada (NSERC) grant #RGPIN-2017-06707.

Declarations

Conflicts of interest The author states that there is no conflict of interest.

Replication of Results The codes are made online available on GitHub (https://github.com/thkwok/GDswarm).

References

Abdelhamid M, Czekanski A (2022) Revisiting non-convexity in topology optimization of compliance minimization problems. Eng Comput 39(3):893–915. https://doi.org/10.1108/EC-01-2021-0052

Agirbas A (2019) Façade form-finding with swarm intelligence. Autom Constr 99:140–151. https://doi.org/10.1016/j.autcon.2018.12.003

Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 194(1):363–393. https://doi.org/10.1016/j.jcp.2003.09.032

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines

of code. Struct Multidisc Optim 43(1):1–16. https://doi.org/10.1007/s00158-010-0594-7
Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202. https://doi.org/10.1007/BF01650949
Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71(2):197–224. https://doi.org/10.1016/0045-7825(88)90086-2
Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9):635–654. https://doi.org/10.1007/s004190050248
Bianchi L, Dorigo M, Gambardella LM, Gutjahr WJ (2009) A survey on metaheuristics for stochastic combinatorial optimization. Nat Comput 8:239–287. https://doi.org/10.1007/s11047-008-9098-4
Cang R, Yao H, Ren Y (2019) One-shot generation of near-optimal topology through theory-driven machine learning. Comput Aided Des 109:12–21. https://doi.org/10.1016/j.cad.2018.12.008
Chen Y (2015) Swarm Intelligence in Architectural Design. Master's thesis, University of California, Berkeley, Berkeley, CA, USA
Deng H, To AC (2021) A parametric level set method for topology optimization based on deep neural network. J Mech Des 143(9):091702. https://doi.org/10.1115/1.4050105
Dogan KM, Suzuki H, Gunpinar E, Kim M-S (2019) A generative sampling system for profile designs with shape constraints and user evaluation. Comput -Aided Des 111:93–112. https://doi.org/10.1016/j.cad.2019.02.002
Feijs LM, Toeters MJ (2018) Cellular automata-based generative design of pied-de-poule patterns using emergent behavior: Case study of how fashion pieces can help to understand modern complexity. Int J Des 12(3):127–144
Felkner J, Chatzi E, Kotnik T, Beer M, Kreinovich V, Kruse R (eds) (2013) Interactive particle swarm optimization for the architectural design of truss structures. (eds Beer, M., Kreinovich, V. & Kruse, R.) IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES), 15–22 (Singapore, 2013)
Gao J, Xue H, Gao L, Luo Z (2019) Topology optimization for auxetic metamaterials based on isogeometric analysis. Comput Methods Appl Mech Eng 352:211–236. https://doi.org/10.1016/j.cma.2019.04.021
Gao J, Wang L, Luo Z, Gao L (2021) IgaTop: an implementation of topology optimization for structures using IGA in MATLAB. Struct Multidisc Optim 64(3):1669–1700. https://doi.org/10.1007/s00158-021-02858-7
Gunpinar E, Coskun UC, Ozsipahi M, Gunpinar S (2019) A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics. Comput-Aided Des 111:65–79. https://doi.org/10.1016/j.cad.2019.02.003
Gupta A, Kurzeja K, Rossignac J, Allen G, Kumar P, Musuvathy S (2019) Programmed-lattice editor and accelerated processing of parametric program-representations of steady lattices. Comput Aided Des 113:35–47. https://doi.org/10.1016/j.cad.2019.04.001
He Y, Cai K, Zhao Z-L, Xie YM (2020) Stochastic approaches to generating diverse and competitive structural designs in topology optimization. Finite Elem Anal Des 173:103399. https://doi.org/10.1016/j.finel.2020.103399
Hertlein N, Buskohl PR, Gillman A, Vemaganti K, Anand S (2021) Generative adversarial network for early-stage design flexibility in topology optimization for additive manufacturing. J Manuf Syst 59:675–685. https://doi.org/10.1016/j.jmsy.2021.04.007
Huang J, Kwok T-H, Zhou C (2019) Parametric design for human body modeling by wireframe-assisted deep learning. Comput Aided Des 108:19–29. https://doi.org/10.1016/j.cad.2018.10.004
Jaafer AA, Al-Bazoon M, Dawood AO (2020) Structural topology design optimization using the binary bat algorithm. Appl Sci. https://doi.org/10.3390/app10041481
Kang S, Deng X, Jin R (2021) A cost-efficient data-driven approach to design space exploration for personalized geometric design in additive manufacturing. J Comput Inf Sci Eng 21(6):061008. https://doi.org/10.1115/1.4050984
Kaveh A, Hassani B, Shojaee S, Tavakkoli S (2008) Structural topology optimization using ant colony methodology. Eng Struct 30(9):2559–2565. https://doi.org/10.1016/j.engstruct.2008.02.012
Kwok T-H, Li Y, Chen Y (2016) A structural topology design method based on principal stress line. Comput -Aided Des 80:19–31. https://doi.org/10.1016/j.cad.2016.07.005
Lee S-M, Han S-Y (2017) Topology optimization based on the harmony search method. J Mech Sci Technol 31:2875–2882. https://doi.org/10.1007/s12206-017-0530-5
Li N-J, Wang W-J, Hsu C-C J, Chang W, Chou H-G, Chang J-W(2014) Enhanced particle swarm optimizer incorporating a weighted particle. Neurocomputing 124:218–227. https://doi.org/10.1016/j.neucom.2013.07.005
Li Z, Liu H, Cheng G, Zhou Y (2021) A diversity metric based on gaussian process model for diverse and competitive design. Struct Multidisc Optim 64(5):2975–2997. https://doi.org/10.1007/s00158-021-02967-3
Mirzendehdel AM, Rankouhi B, Suresh K (2018) Strength-based topology optimization for anisotropic parts. Addit Manuf 19:104–113. https://doi.org/10.1016/j.addma.2017.11.007
Mirzendehdel AM, Behandish M, Nelaturi S (2019) Exploring feasible design spaces for heterogeneous constraints. Comput Aided Des 115:323–347. https://doi.org/10.1016/j.cad.2019.06.005
Nie Z, Lin T, Jiang H, Kara LB (2021) Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. J Mech Des 143(3):031715. https://doi.org/10.1115/1.4049533
Oh S, Jung Y, Kim S, Lee I, Kang N (2019) Deep generative design: Integration of topology optimization and generative models. J Mech Des 141:111405. https://doi.org/10.1115/1.4044229
Pantazis E, Gerber D (2018) A framework for generating and evaluating façade designs using a multi-agent system approach. Int J Archit Comput 16(4):248–270. https://doi.org/10.1177/1478077118805874
Sales E, Kwok T-H, Chen Y (2021) Function-aware slicing using principal stress line for toolpath planning in additive manufacturing. J Manuf Process 64:1420–1433. https://doi.org/10.1016/j.jmapro.2021.02.050
Sethian JA, Wiegmann A (2000) Structural boundary design via level set and immersed interface methods. J Comput Phys 163(2):489–528. https://doi.org/10.1006/jcph.2000.6581
Sigmund O, Maute K (2013) Topology optimization approaches. Struct Multidisc Optim 48(6):1031–1055. https://doi.org/10.1007/s00158-013-0978-6
Sun H, Ma L (2020) Generative design by using exploration approaches of reinforcement learning in density-based structural topology optimization. Designs. https://doi.org/10.3390/designs4020010
Tsiliakos M, Achten H, Pavlicek J, Hulin J, Matejovska D (eds) Swarm materiality: A multi-agent approach to stress driven material organization. (eds Achten, H., Pavlicek, J., Hulin, J. & Matejovska, D.) Digital Physicality, Proceedings of the 30th eCAADe Conference, 301-309 (Prague, Czech Republic, 2012)
Vaissier B, Pernot J-P, Chougrani L, Véron P (2019) Genetic-algorithm based framework for lattice support structure optimization in additive manufacturing. Comput-Aided Des 110:11–23. https://doi.org/10.1016/j.cad.2018.12.007

Wang S, Tai K (2005) Structural topology design optimization using genetic algorithms with a bit-array representation. Comput Methods Appl Mech Eng 194(36):3749–3770. https://doi.org/10.1016/j.cma.2004.09.003

Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1):227–246. https://doi.org/10.1016/S0045-7825(02)00559-5

Wang B, Zhou Y, Zhou Y, Xu S, Niu B (2018) Diverse competitive design for topology optimization. Struct Multidisc Optim 57(2):879–902. https://doi.org/10.1007/s00158-017-1762-9

Wang Q-S, Ye J, Wu H, Gao B-Q, Shepherd P (2019) A triangular grid generation and optimization framework for the design of free-form gridshells. Comput-Aided Des 113:96–113. https://doi.org/10.1016/j.cad.2019.04.005

Wang X, Song Y, Tang P (2020) Generative urban design using shape grammar and block morphological analysis. Front Archit Res 9(4):914–924. https://doi.org/10.1016/j.foar.2020.09.001

Watson M, Leary M, Brandt M (2021) Generative design of truss systems by the integration of topology and shape optimisation. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-021-07943-1

Wu CY, Tseng KY (2010) Topology optimization of structures using modified binary differential evolution. Struct Multidisc Optim 42:939–953. https://doi.org/10.1007/s00158-010-0523-9

Wu J, Dick C, Westermann R (2016) A system for high-resolution topology optimization. IEEE Trans Vis Comput Graph 22(3):1195–1208. https://doi.org/10.1109/TVCG.2015.2502588

Wu J, Aage N, Westermann R, Sigmund O (2018) Infill optimization for additive manufacturing-approaching bone-like porous structures. IEEE Trans Vis Comput Graph 24(2):1127–1140. https://doi.org/10.1109/TVCG.2017.2655523

Yoo S, Lee S, Kim S, Hwang KH, Park JH, Kang N (2021) Integrating deep learning into CAD/CAE system: generative design and evaluation of 3D conceptual wheel. Struct Multidisc Optim. https://doi.org/10.1007/s00158-021-02953-9

Youssef AM, Zhai ZJ, Reffat RM (2018) Generating proper building envelopes for photovoltaics integration with shape grammar theory. Energy Build 158:326–341. https://doi.org/10.1016/j.enbuild.2017.09.077

Yu Y, Hur T, Jung J, Jang IG (2019) Deep learning for determining a near-optimal topological design without any iteration. Struct Multidisc Optim 59(3):787–799. https://doi.org/10.1007/s00158-018-2101-5

Zhang X, Ramos AS, Paulino GH (2017) Material nonlinear topology optimization using the ground structure method with a discrete filtering scheme. Struct Multidisc Optim 55(6):2045–2072. https://doi.org/10.1007/s00158-016-1627-7

Zhang Y, Wang Z, Zhang Y, Gomes S, Bernard A (2020) Bio-inspired generative design for support structure generation and optimization in additive manufacturing (AM). CIRP Ann-Manuf Technol 69(1):117–120. https://doi.org/10.1016/j.cirp.2020.04.091

Zhou M, Rozvany GIN (1991) The coc algorithm, part ii: Topological, geometrical and generalized shape optimization. Comput Methods Appl Mech Eng 89(1):309–336. https://doi.org/10.1016/0045-7825(91)90046-9