RESEARCH PAPER

CrossMark

Topology optimization of conductive heat transfer problems using parametric L-systems

Teemu J. Ikonen[1] (ID) · Gilles Marck[1] · András Sóbester[1] · Andy J. Keane[1]

Abstract

Generative encodings have the potential of improving the performance of evolutionary algorithms. In this work we apply parametric L-systems, which can be described as developmental recipes, to evolutionary topology optimization of widely studied two-dimensional steady-state heat conduction problems. We translate L-systems into geometries using the turtle interpretation, and evaluate their objective functions, i.e. average and maximum temperatures, using the Finite Volume Method (FVM). The method requires two orders of magnitude fewer function evaluations, and yields better results in 10 out of 12 tested optimization problems (the result is statistically significant), than a reference method using direct encoding. Further, our results indicate that the method yields designs with lower average temperatures than the widely used and well established SIMP (Solid Isotropic Material with Penalization) method in optimization problems where the product of volume fraction and the ratio of high and low conductive material is less or equal to 1. Finally, we demonstrate the ability of the methodology to tackle multi-objective optimization problems with relevant temperature and manufacturing related objectives.

1 Introduction

Electronic devices are packed in increasingly compact spaces, which increases the heat density generated by their components. To prevent overheating, their architecture must be designed with an effective cooling system. The first task of the cooling system is to conduct the heat from the electronic components to a heat sink, using highly conductive material, e.g. copper or aluminum. The availability of conductive material is limited by space constraints and because the manufacturers always wish to reduce the cost of such components. Consequently,

✉ Teemu J. Ikonen
   t.j.ikonen@soton.ac.uk

[1] Computational Engineering and Design,
   University of Southampton, Southampton,
   Hampshire, SO16 7QF, UK

properly distributing the high conductivity material through a finite volume becomes an important topology optimization problem.

Various topology optimization methods have been presented to maximize the thermal efficiency of engineering systems considering conduction, convection and conjugate heat transfer (Dbouk 2017). In this paper, we consider conductive problems, as the first step towards more complicated and realistic problems. The majority of the published papers tackling this topic consider steady-state conduction inside a rectangular, two-dimensional design domain. Bejan (1997) defined the so-called "volume-to-point", or "area-to-point", design problem where a finite design domain, with a uniformly distributed heat generation rate, is filled with high and low conductivity materials. The objective is to minimize the average or maximum temperature over the domain by distributing a limited amount of high conductivity material, to efficiently transfer the produced heat to the heat sink, which is a short section of the domain boundary. The remaining boundary conditions are adiabatic. This problem has been extensively studied and has become a popular benchmark in the field of thermal engineering.

To solve the problem, Bejan (1997) applied constructal theory, which is based on observations from the nature. According to this theory, the solutions are constructed from blocks with different designs and sizes, and, for each scale, their geometric details are determined theoretically to minimize their conductive resistance, which is a non-dimensional expression of their maximum temperature. Li et al. (1999) modified Evolutionary Structural Optimization (ESO) heuristics, initially developed for structural optimization, to be suitable for a conductive steady-state heat transfer problem. Gao et al. (2008) presented modified bi-directional ESO heuristics for a similar optimization problem, and studied both design-dependent and independent heat load cases. In these studies, (B)ESO heuristics were applied to a design problem where the heat sink extends over the entire domain boundary. Later, Marck (2012) applied ESO to the problem defined by Bejan (1997). Cheng et al. (2003) studied the problem using the *bionic optimization approach*, where the conductive material domain is iteratively expanded near regions where their temperature gradients are the highest and removed from regions where they are the smallest. The cellular automaton is another approach to the problem and its first application is due to Boichot et al. (2009). The algorithm aims at minimizing thermal gradients, or heat fluxes, at the boundary between high and low conductive materials. The authors, as well as Marck (2012), describe the method as being a simple way of obtaining a reasonable tree-like solution, which, however, is likely to be sub-optimal.

According to the dedicated scientific literature, the most promising methods to solve this design problem are based on the density interpolation approach (Bendsøe 1989). These methods were initially developed for structural topology optimization, where the discretization of the Partial Differential Equation (PDE) is typically conducted using the Finite Element Method (FEM). Gersborg-Hansen et al. (2006) were the first to obtain the design sensitivities from the Finite Volume Method (FVM), and used them in conjunction with topology optimization. Marck et al. (2012) used the SIMP method, with an aggregated objective function approach, in a multi-objective optimization study in order to minimize both average and variance temperatures over the design domain. Dirker and Meyer (2013) tested a variety of objective functions and problem related parameters of the SIMP method, and reported their results using non-dimensional measures for thermal conductivity and 'definiteness', i.e. how well intermediate material is eliminated from the final design. Their results show that the final design is highly dependent on the penalization coefficient value. Dede (2009) and Burger et al. (2013) applied the SIMP-based methodology to a three-dimensional volume-to-point design problem. The Method of Moving Asymptotes (MMA) (Svanberg 1987) is used as the underlying gradient-based optimizer in all of the papers cited in this paragraph.

Apart from constructal theory, the aforementioned studies are based on gradient-based approaches. However, evolutionary algorithms have also been applied to this problem. Xu et al. (2007) used separately both Genetic Algorithms (GA) and simulated annealing (Kirkpatrick et al. 1983) to seek the optimal combination of discretized design domain elements. Later, Boichot and Fan (2016) show that their carefully tuned GA yields discrete designs having lower non-dimensional thermal resistances than studies using the cellular automaton (Mathieu-Potvin and Gosselin 2007; Boichot et al. 2009; Marck 2012), constructal theory (Bejan 1997; Ghodoossi and Eġrican 2003) and ESO (Marck 2012), and similar thermal resistances to a study carrying out the SIMP approach (Marck et al. 2012). However, their algorithm requires an order of five million function evaluations to reach full convergence.

Faster convergences with evolutionary algorithms may be obtained by using alternative strategies to parameterize the design space. Pedro et al. (2008) parameterized the geometry of a tree-like structure (with parameters defining branch angles and lengths), and used a GA to minimize the maximum temperature over the finite-size domain. One of their conclusions points out that the level of geometric complexity has a considerable effect on the optimized objective function value.

Another class of alternative strategies is *generative encodings*, in which a genotype typically is a "construction recipe" that indirectly defines the actual design (cf. genotype-phenotype distinction in biology). To solve the conductive heat transfer problem, Lohan et al. (2017) implemented the space colonization algorithm (Runions et al. 2005), in which the design space is first seeded with a set of attraction points, and then a branching structure is iteratively constructed to "colonize" these points. Their objective is to minimize the thermal compliance of the design domain, which is discretized using both structured and unstructured meshes. Guo et al. (2018) proposed a generative encoding approach based on artificial neural networks. The approach uses a variational autoencoder (Kingma and Welling 2013), the purpose of which is to reduce the dimensionality of the design via its latent layers, and deep convolutional neural networks (Krizhevsky et al. 2012), to prevent the appearance of disconnected high conductive material in the designs.

In comparison to direct encodings, often used in evolutionary optimization, generative encodings are found to be more compact and have better scalability, due to their natural capability to produce hierarchical geometries with self-similar features (Hornby and Pollack 2001; Stanley and Miikkulainen 2003; Kobayashi 2010). One type of

generative encoding is based on L-systems (Lindenmayer 1968a, b), and their turtle interpretation (Prusinkiewicz 1986), originally developed to model the (topological) development of trees and other organisms. Hornby and Pollack used parametric L-systems as a parameterization method to perform an evolutionary topology optimization of a table (Hornby and Pollack 2001) and a locomoting robot (Hornby and Pollack 2002). They note that, in both applications, the L-systems-based algorithms yield better designs and faster convergence than corresponding algorithms with direct encoding. Kobayashi (2010) used L-systems to describe the venation topology of artificial cordate leaves, and evolved them using multi-objective optimization, to minimize their mass and pressure drop (as well as pressure drop and maximum temperature). In addition, Sabbatini et al. (2015) applied L-systems to multi-objective optimization: their objective was to minimize the vibration amplitude and mass of a stiffened plate structure.

Nearly all of the earlier mentioned studies tackling the conductive heat transfer problem report that their optimized designs feature branching tree structures. Moreover, Dede (2009) describes his designs to have self-similar features.

Therefore, the purpose of our study is to implement L-systems as the parameterization method in evolutionary topology optimization, and to apply this methodology to the described conductive heat transfer problem. As far as we are able to ascertain, the only similar approach in the literature is the one by Kobayashi (2010). However, he defines the optimization problem to represent design optimization of an artificial cordate leaf, instead of the cooling of an electrical device as originally defined by Bejan (1997). Therefore, his design domain has the shape of a leaf and he uses the pressure drop as one of the objectives. The pressure drop is meaningless in the context of the design problem studied here. Due to these aspects, his results cannot be benchmarked against other studies on the original design problem. We benchmark our single-objective optimization results against relevant studies in the literature, and demonstrate the use of the methodology in multi-objective optimization with relevant temperature and manufacturing related objectives.

2 Optimization problem

In this paper, we study the conductive steady-steate heat transfer problem defined by Bejan (1997). The problem represents an electrical device that is cooled down by a limited amount of high conductive material aiming at driving the produced heat to a heat sink, located at the boundary of the finite size volume.
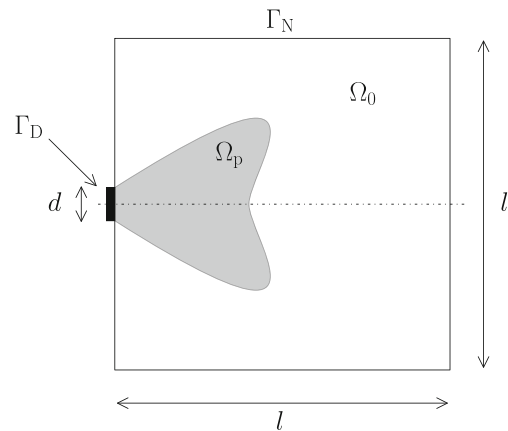

Fig. 1 The optimization problem of optimally distributing a limited amount of high conductive material (domain Ω_p), to minimize the average, or maximum, temperature in domain $\Omega_p \cup \Omega_0$

Let us consider a two-dimensional square-shape design domain Ω, with a side length l. The domain consists of two subdomains, Ω_p and Ω_0, such that $\Omega_p \cup \Omega_0 = \Omega$ and $\Omega_p \cap \Omega_0 = \varnothing$ (Fig. 1). Subdomains Ω_p and Ω_0 denote high and low conductive materials with thermal conductivities k_p and k_0, respectively. The latter represents an area of the device that is filled with electrical components, and thus is defined to have heat-generation rate[1] q. The design domain is bound by Dirichlet and Neumann boundary conditions, Γ_D and Γ_N (Fig. 1). The Dirichlet boundary condition (heat sink) is located in the middle of the left-hand side boundary and has a width of d, whereas the remaining boundary conditions are adiabatic (Neumann). Thus, the governing equations for steady-state conductive heat transfer in the domain are

$$\begin{cases} \nabla \cdot (k\nabla T) + q & = 0 \text{ on } \Omega \\ (k\nabla T) \cdot \mathbf{n} & = 0 \text{ on } \Gamma_N \\ T & = 0 \text{ on } \Gamma_D, \end{cases} \tag{1}$$

where $\mathbf{n}$ is the outward normal vector of the boundary, k is the local thermal conductivity, that is k_p or k_0 corresponding to subdomains Ω_p and Ω_0 and q is the local heat generation rate, that is q_0 in domain Ω_0 and 0 elsewhere.

In view of optimal design studies, it is convenient to introduce the characteristic function of Ω_p, $\chi_{\Omega_p} : \Omega \rightarrow \{0, 1\}$, defined by

$$\chi_{\Omega_p}(x) = \begin{cases} 1 \text{ if } x \in \Omega_p \\ 0 \text{ if } x \in \Omega_0. \end{cases} \tag{2}$$

---

[1]In fact, Bejan (1997) defined the entire domain to have an evenly distributed heat-generation rate. We have eliminated the heat generation from subdomain Ω_p as the same choice is made by Boichot and Fan (2016), whose results we use as a benchmark. However, the definition by Bejan (1997) is a closer representation of a real chip, which high conductive layer is on top/bottom.

This allows one to define the scalar variables depending on the characteristic function, such as:

- the thermal conductivity: $k(\chi_{\Omega_p}) = k_0 + (k_p - k_0)\chi_{\Omega_p}$
- the heat production rate: $q(\chi_{\Omega_p}) = q_0(1 - \chi_{\Omega_p})$.

Then, the optimization problem becomes:

$$
\begin{aligned}
\min_{\chi_{\Omega_p} \in \mathcal{D}_\phi} \quad & f(\chi_{\Omega_p}) \\
\text{subject to,} \quad & \nabla \cdot \left(k(\chi_{\Omega_p})\nabla T\right) + q(\chi_{\Omega_p}) = 0 \text{ on } \Omega, \\
& (k(\chi_{\Omega_p})\nabla T) \cdot \mathbf{n} = 0 \qquad \text{on } \Gamma_\mathrm{N}, \\
& T = 0 \qquad\qquad\qquad \text{on } \Gamma_\mathrm{D}, \\
\text{where} \quad & \mathcal{D}_\phi = \{\chi_{\Omega_p}, \; |\Omega_p| \le \phi|\Omega|\}
\end{aligned}
\tag{3}
$$

where $\phi \in [0, 1]$ is the volume constraint, restricting the area covered by the domain Ω_p to a fraction of the whole finite-size volume Ω.

Unless mentioned otherwise, the objective function $f(\chi_{\Omega_p})$ is either the average temperature $\bar{T}$ or the maximum temperature $T_{\max}$. In the frame of the former, the objective function is typically defined as

$$
f(\chi_{\Omega_p}) = \frac{1}{|\Omega|} \int_\Omega T \mathrm{d}\Omega.
\tag{4}
$$

However, some authors consider an alternative form of this objective function (Gersborg-Hansen et al. 2006; Lohan et al. 2017), defined by

$$
g(\chi_{\Omega_p}) = \int_\Omega \nabla T \cdot (k\nabla T)\mathrm{d}\Omega.
\tag{5}
$$

By using Green's first identity in conjunction with the Dirichlet and Neumann boundary conditions, it is possible to show that the objective function $g(\chi_{\Omega_p})$ can be rewritten as

$$
g(\chi_{\Omega_p}) = \int_\Omega q(\chi_{\Omega_p})T \mathrm{d}\Omega.
\tag{6}
$$

Consequently, from a design point of view, minimizing $g(\chi_{\Omega_p})$ is strictly equivalent to minimizing $f(\chi_{\Omega_p})$ if, and only if, q is uniformly distributed over Ω or, in other words, it does not depend on the characteristic function χ_{Ω_p}.

3 Methods

Our approach to solve the optimization problem is to parameterize the design space using an L-systems-based generative encoding, and evolve the designs via a Genetic Algorithm (GA). L-systems are a type of formal grammars (Chomsky 1956), and hence categorized as a grammatical approach to generative encodings (Stanley and Miikkulainen 2003).

In this section, we introduce L-systems via examples, and present our approach to map the phenotypes of L-systems into the two-dimensional design domain (Fig. 1). Finally, we briefly describe the numerical method to solve

the temperature field in the design domain, satisfying the governing equations (1), using the Finite Volume Method (FVM).

3.1 L-systems

L-systems were introduced by Lindenmayer (1968a, b), who studied the developmental process of multicellular organisms, in the late 1960s. The fundamental idea of L-systems is that complex objects (e.g. plants) can be modeled by repeatedly modifying a simple object by following a set of predefined rewriting rules.

Using the taxonomy of L-systems, the process is started from a (simple) initial object, called the *axiom* Ω_0. Further, the state of the system after the rewriting rules are applied n times is referred to as its nth *developmental stage* Ω_n. Both the axiom and rewriting rules are defined using an alphabet Σ of letters and/or symbols, which are referred to as *characters*. The left and right-hand sides of a rewriting rule are referred to as the *predecessor* and *successor*, respectively.

Let us consider a simple example,[2] where the alphabet $\Sigma \equiv [a, b]$, the axiom $\omega_0 = b$ and rewriting rules are $P_1 : a \to ab$ and $P_2 : b \to a$. To obtain the first developmental stage, the axiom letter 'b' is converted into 'a' due to the rewriting rule P_2, and therefore $\omega_1 = a$. When the rules are applied further, the following developmental stages are obtained: $\omega_2 = ab$, $\omega_3 = aba$, $\omega_4 = abaab$, $\omega_5 = abaababa, \ldots$.

The L-systems we consider here are deterministic and context-free, i.e. the successor is only dependent on the predecessor, and not on its neighboring characters (DOL-systems). If a character has not been assigned a rewriting rule, it is directly copied to the new sequence (identity rule).

In L-systems, these sequencies of characters are interpreted into graphs that represent organisms. While several interpretation formalisms have been presented in the literature, we focus here on the turtle interpretation, which characteristically produces branched tree geometries. For an extensive review of L-systems and other types of interpretation formalisms, the reader may wish to consult the text book by Prusinkiewicz and Lindenmayer (2012).

3.1.1 Turtle interpretation

In the turtle interpretation, the sequences of characters are interpreted into geometries via a moving turtle (cf. the turtle feature in the programming language LOGO). The orientation of the turtle is defined by its axial coordinates and heading. Each letter or symbol in the sequence is a

---

[2]This example is, in fact, equivalent to formal grammars, but we describe it here using the taxonomy of L-systems.

Fig. 2 Visualization of a growing plant, interpreted from the L-system defined in (7). The variable n is ordinal of the developmental stage. Thus, $n = 0$ corresponds to the axiom ω_0, and $n = 1 \ldots 3$ to the three subsequent developmental stages of the plant

(a) $n = 0$ (b) $n = 1$ (c) $n = 2$ (d) $n = 3$
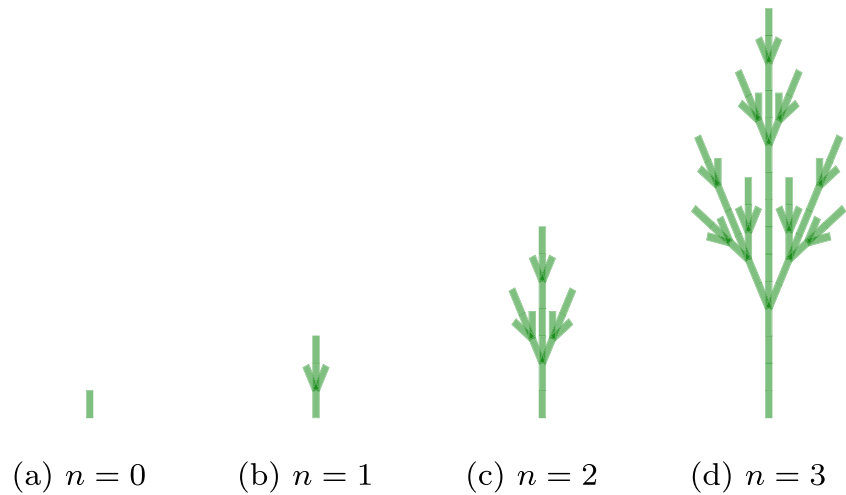
command for the turtle, such as 'move ahead by distance d' or 'turn clockwise by angle $\Delta\theta$'. The moving turtle draws the lines of the geometry while executing the series of commands.

This section demonstrates the development of an example plant using L-systems and the turtle interpretation. The example is presented by Prusinkiewicz and Lindenmayer (2012). Let us consider an alphabet Σ containing the letters F and X, and symbols '+', '−', '[' and ']'. Our example plant is defined by the following input:[3]

$$
\begin{array}{lll}
\text{Axiom:} & \omega_0 = X \\
\text{Rules:} & P_1 : F \to FF \\
& P_2 : X \to F[+X][-X]FX \\
\text{Parameters:} & \Delta\theta = 25.7°
\end{array}
\tag{7}
$$

The process is started by generating the character sequence of the desired developmental stage, in the same way as in the previous example. Following the axiom ω_0, the next two developmental stages of the system are:

$$
\omega_1 = F[+X][-X]FX
\tag{8}
$$

and

$$
\begin{aligned}
\omega_2 = {} & FF[+F[+X][-X]FX][-F[+X][-X]FX] \\
& FFF[+X][-X]FX.
\end{aligned}
\tag{9}
$$

Next, the sequences are translated into geometries using the turtle interpretation. The characters have the following meaning for the turtle:

– letters move the turtle forward by step size d (the moving turtle draws a line, having the width w),
– symbol '+' turns the turtle anti-clockwise by angle $\Delta\theta$,
– symbol '−' turns the turtle clockwise by angle $\Delta\theta$,
– symbol '[' commands the turtle to stack its orientation, and

– symbol ']' returns the turtle to the previously stacked orientation.

The last two symbols enable the creation of branches, as the turtle may return to a previously visited location. The axiom ($n = 0$) and the first four developmental stages of the system ($n = 1 \ldots 3$) are interpreted into plants in Fig. 2.

3.1.2 Parametric L-systems

Using the above described turtle commands, L-systems are restricted to produce geometries consisting of lines segments with lengths that are integer multiples of the step size d, and further, the angles between the line segments are restricted to be integer multiples of the turning angle $\Delta\theta$.

More complex geometries, free of these restrictions, can be generated by parametric L-systems (Prusinkiewicz and Hanan 1990), in which symbols are associated with numerical values. In this work, we use the following parametric symbols:

– $\$(\theta)$ turns the turtle by angle θ (positive direction being anti-clockwise),
– $@(c_s)$ changes the prevailing step size to $d_i = d_{i-1}c_s$, and
– $\&(c_w)$ changes the prevailing line width to $w_i = w_{i-1}c_w$.

Symbols '[' and ']' stack and unstack the prevailing attributes, step size d_i and line width w_i, in the same way as the orientation. To demonstrate the use of parametric symbols, let us consider another example L-system, which is defined by the following input:

$$
\begin{array}{ll}
\text{Axiom:} & \omega_0 = X \\
\text{Rules:} & P_1 : F \to FF \\
& P_2 : X \to F\&(\tfrac{2}{3})[\$(\tfrac{\pi}{4})X][\$(-\tfrac{\pi}{12})X]F@(\tfrac{2}{3})X
\end{array}
\tag{10}
$$

---

[3]The value of $\Delta\theta$ was used by Prusinkiewicz and Lindenmayer (2012).

Fig. 3 Visualization of the parametric L-system defined in (10)

(a) $n = 0$ (b) $n = 1$ (c) $n = 2$ (d) $n = 3$
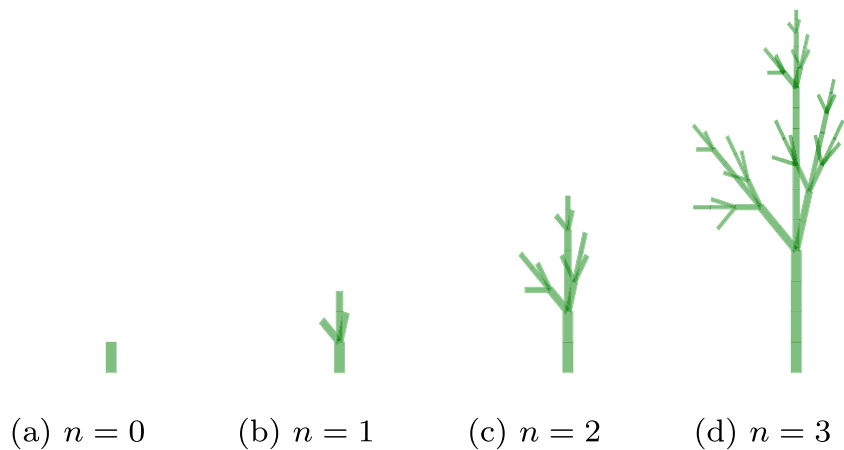
The graphical interpretation of the L-system is shown in Fig. 3 (see the topological congruence to Fig. 2). While the topology of the plant is defined based on the order of the characters in the axiom and the rewriting rules, the parameters associated with the symbols define its shape. In the next section, we describe the means of evolving the parametric L-systems, to minimize the objective functions defined in Section 2.

3.1.3 Optimization via an evolutionary algorithm

Evolutionary algorithms are optimization heuristics that are inspired by Darwinian natural evolution. They evolve a population of candidate designs, via mathematical operators mimicking natural selection, recombination and mutation, to find the best adoption to a simulated environment.

In this study, the candidate designs are parametric L-systems, which we evolve via a GA[4] to minimize the objective function, representing the environment. A key implementation detail is how to encode a parametric L-system into a numerical format suitable for GAs. Here, we use a modified version of the numerical representation defined by Kobayashi (2010), which he developed to represent the venation topology of an artificial cordate leaf.

We encode the axiom and rewriting rules, as well as some additional variables sequentially into a vector $\mathbf{x}$ of real numbers, with $x_i \in [0, 1] \forall i$, as

$$\mathbf{x} = [\underbrace{x_{a,1}\ x_{a,2}\ \dots\ x_{a,N_a}}_{\text{Axiom } \omega_0}\ \underbrace{\mathbf{y}_1\ \mathbf{y}_2\ \dots\ \mathbf{y}_{N_P}}_{\text{Rewriting rules } P_i}\ \underbrace{x_{c,1}\ x_{c,2}\ \dots\ x_{c,N_v}}_{\text{Additional variables}}].$$

(11)

The axiom ω_0 consists of N_a letters, each of which are represented by a real number $x_{a,i}$. The interval $[0, 1]$ of the real number is divide into equally sized segments that represent the letters in the alphabet Σ. As an example, if the

---

[4]Genetic programming could be a suitable alternative method. Its main benefit is that L-systems could directly be represented as programs without such numerical representations that are requred by GAs.

alphabet contains letters $\{A, B, C, D\}$, the encoding is the following:

$$\begin{aligned}
&\text{if } x_i \in [0, \tfrac{1}{4}) \ \rightarrow A \\
&\text{if } x_i \in [\tfrac{1}{4}, \tfrac{1}{2}) \ \rightarrow B \\
&\text{if } x_i \in [\tfrac{1}{2}, \tfrac{3}{4}) \ \rightarrow C \\
&\text{if } x_i \in [\tfrac{3}{4}, 1] \ \rightarrow D
\end{aligned}$$

(12)

Each letter σ_i in the alphabet Σ, containing a total of N_P letters, is assigned a rewriting rule in the format

$$P_i : \sigma_i \rightarrow \beta_{i,1} \beta_{i,2} \dots \beta_{i,14},$$

(13)

where the successor of the rule consists of tokens $\beta_{i,1} \dots \beta_{i,14}$, which are represented by

$$\mathbf{y}_i = \begin{bmatrix} x_{i,1} & x_{i,2} & \dots & x_{i,14} \end{bmatrix}.$$

(14)

The successor is decoded from the vector $\mathbf{y}_i$ as:

– tokens $\beta_{i,1}$ and $\beta_{i,8}$:

$$\begin{aligned}
&\text{if } x_{i,j} \in [0, \tfrac{1}{2}), \qquad \beta_{i,j} = [\\
&\text{else if } x_{i,j} \in [\tfrac{1}{2}, 1], \quad \beta_{i,j} = \lambda
\end{aligned}$$

– tokens $\beta_{i,2}$ and $\beta_{i,9}$:

$$\beta_{i,j} = \$(g(x_{i,j}, \theta_{\min}, \theta_{\max}))$$

– tokens $\beta_{i,3}$ and $\beta_{i,10}$:

$$\beta_{i,j} = @(g(x_{i,j}, c_{s,\min}, c_{s,\max}))$$

– tokens $\beta_{i,4}$ and $\beta_{i,11}$:

$$\beta_{i,j} = \&(g(x_{i,j}, c_{w,\min}, c_{w,\max}))$$

– tokens $\beta_{i,5}, \beta_{i,6}, \beta_{i,12}$ and $\beta_{i,13}$:

$$\begin{aligned}
&\text{if } x_{i,j} \in [0, \tfrac{1}{N_a+1}), \qquad \beta_{i,j} = A \\
&\text{if } x_{i,j} \in [\tfrac{1}{N_a+1}, \tfrac{2}{N_a+1}), \quad \beta_{i,j} = B \\
&\text{if } x_{i,j} \in [\tfrac{2}{N_a+1}, \tfrac{3}{N_a+1}), \quad \beta_{i,j} = C \\
&\dots \\
&\text{if } x_{i,j} \in [\tfrac{N_a}{N_a+1}, 1], \qquad \beta_{i,j} = \lambda
\end{aligned}$$

– tokens $\beta_{j,7}$ and $\beta_{j,14}$:

if $\beta_{i,j-6} = [, \quad \beta_{i,j} =]$
if $\beta_{i,j-6} = \lambda, \quad \beta_{i,j} = \lambda$,

where λ is an empty token. Further, g is a scaling function, defined as

$$g(x, c_{\min}, c_{\max}) = c_{\min} + x(c_{\max} - c_{\min}), \tag{15}$$

where $c_{\min}$ and $c_{\max}$ are the minimum and maximum bounds, respectively, of the design variable associated with a parametric symbol.

Before going into the encoding of the additional variables, let us introduce two new design variables. First, the non-dimensional extent variable is defined as

$$c_{\text{extent}} = l_{\text{branch}}/\sqrt{l^2 + (l/2)^2}, \tag{16}$$

where l_{branch} is the distance between the starting point of the turtle and the point in its path that is furthest away from the starting point (see Fig. 7). The phenotypes are scaled in order to fit the parameter l_{branch} to satisfy (16).

Second, the majority of the optimized results in the literature (e.g. Boichot and Fan 2016 and Marck et al. 2012) consists of tree-like structures, where the width of the branches decreases when moving away from the heat sink. This supports the physical behavior involving branches becoming wider when approaching the heat sink, since they drive larger heat flux quantities collected through the domain. The parametric symbol $\&(c_{\text{w}})$ enables changes in the prevailing width between steps, but not during a step. Therefore, we introduce a new variable $c_{\text{t,j}}$, specific to the letter σ_j in the alphabet, which changes the prevailing width during a step linearly from w_{i-1} to $w_i = w_{i-1}c_{\text{t,j}}$. These variables induce the structural components of the phenotype to have a trapezoid shape, and thus we refer to them as trapezoid variables.

The last N_{v} elements of the vector $\mathbf{x}$ represent additional variables, which are 1) the vertical coordinate y_0 of the starting point of the turtle (see Fig. 7), 2) the initial heading θ_0 of the turtle (see Fig. 7), 3) the age n of the L-system,

Table 1 Minimum and maximum values of additional variables and variables associated with parametric symbols

Parameter	Min	Max
Change in the heading θ (rad)	$-\pi/2$	$\pi/2$
Relative change in step size c_{s} [–]	0.5	2.0
Relative change in width c_{w} [–]	0.5	1.0
Vertical coordinate y_0 (mm)	0	$l/10$
Initial heading θ_0 (rad)	0	$\pi/2$
Age n [–]	2	4
Extent c_{extent} [–]	0.3	1.0
Trapezoid variables $c_{\text{t,i}}$ [–]	0.4	1.0

Table 2 Parameters controlling the evolutionary process

Parameter	Value
Population size	150
Tournament pool size	8
Crossover rate	0.8
Mutation rate	0.04
Crossover type	'two-point'
Mutation type	'Gaussian'
Elitism	1 elite individual

4) the extent variable c_{extent} and 5) trapezoid variables $c_{t,1} \ldots c_{t,N_{\text{P}}}$. The age n is an integer variable, encoded in the same way as the axiom letters (see the example in (12)), whereas the other additional variables are scalar variables, encoded via the scaling function g (15).

As a summary, the design variable vector $\mathbf{x}$ has a total length of

$$N_{\text{total}} = N_{\text{a}} + 14N_{\text{P}} + 4 + N_{\text{P}}. \tag{17}$$

In this study, we use an L-system design space, in which the axiom consists of four letters ($N_{\text{a}} = 4$), and the alphabet contains letters $\{A, B, C, D\}$, as well as the symbols described above. The number of encoded rewriting rules is equal to the number of letters in the alphabet ($N_{\text{P}} = 4$). We define variables associated with the parametric symbols and additional variables to be bound between the minimum and maximum values listed in Table 1.

In an earlier study (Ikonen and Sóbester 2018), we performed a statistical analysis of different control parameters on the map L-systems-based topology optimization method, using five test cases with low computation cost. We reported the best-performing parameter combinations as a Pareto front in the space of the average number of objective function evaluations and ranking based on average optimized fitnesses. For lack of a better guess of the most suitable control parameters, we here use a Pareto-optimal combination of control parameters[5] from the statistical analysis (although its L-systems-based parameterization is different from the one we use here). The control parameters are listed in Table 2. Because the parameterization contains many scalar variables, we use a Gaussian mutator,[6] with mean $\mu = 0$ and standard deviation $\sigma = 0.3$. The mutation operator may set a real variable x_i of vector (11) outside its bounds [0, 1], in which case it is repaired by adding/subtracting the appropriate integer number, e.g. -0.1 becomes 0.9. The optimization runs are terminated when no improvement is found over the last 50 consecutive generations.

---

[5]cf. parameter combination #4 in Table 4 of the paper by Ikonen and Sóbester (2018).

[6]Swap mutator was used in the statistical analysis.

The GA is implemented using the open-source Python package DEAP (Distributed Evolutionary Algorithm for Python) (Fortin et al. 2012).

3.2 Mapping

The optimization problem introduced in (3) deals with a volume constraint, expressed as a fraction ϕ of the total area $|\Omega|$. So far, the parametrization via L-systems and its turtle interpretation, introduced in Section 3.1, does not take care of this volume constraint and, therefore, the purpose of this subsection is to explain *i*) how our method handles it and *ii*) how the final design is mapped onto a Cartesian grid.

Figure 4 shows how an L-system element *i* is built from two nodes, respectively $P_{i,0}$ and $P_{i,1}$, and two widths, respectively $w_{i,0}$ and $w_{i,1}$ (knowing that $w_{i,0} \geq w_{i,1}$). The resulting geometry is an isosceles trapezoid where the points $P_{i,0}$ and $P_{i,1}$ are located in the middle of its bases. However, this is not sufficient to properly map the whole L-system structure, since two consecutive elements may form a notch at their junction (as shown in the lower-right corner of Fig. 4). The existence of such notches in the structure drastically reduces its thermal performance. In order to overcome this problem, we add two additional isosceles trapezoids at each extremity of the main one. Their smallest base is half of their largest base, and their height is either $w_{i,0}/4$ or $w_{i,1}/4$ depending on the extremity involved. We
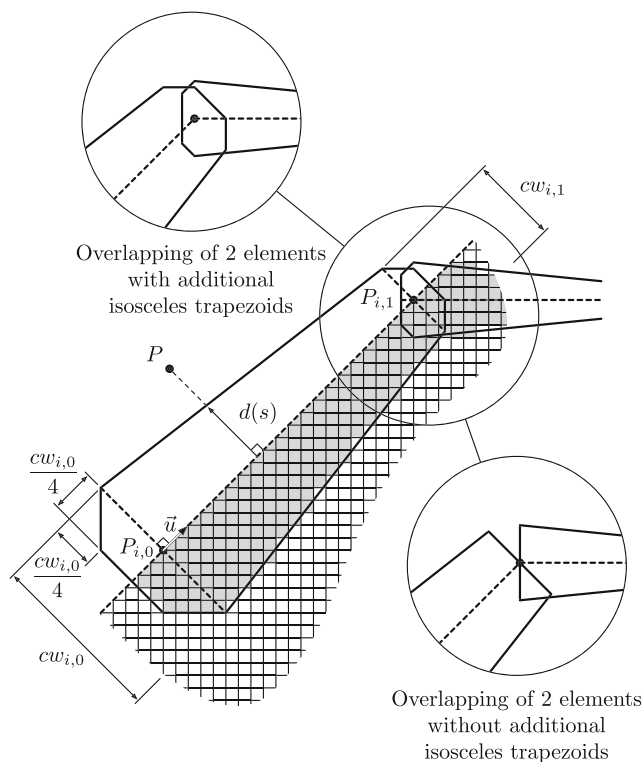
selected this strategy, and these parameters, after several tests on different configurations, as they demonstrated the most robust behavior in creating appropriate conductive paths.

After building each element of an L-system structure using this approach, we merge the resulting set of polygons into a single polygon *L*. This operation relies on the open-source Python library Shapely (Gillies et al. 2007). We handle the volume constraint by adjusting the widths of each trapezoidal base with a correction factor *c*, such that $|L(c)| = \phi|\Omega|$. The correction factor *c* can be computed by using a simple bisection method: in our case, we have implemented Brent's method, which we found to be capable of finding an acceptable root for *c* in usually less than ten iterations.

However, due to the nature of the L-system mapping strategy, the volume constraint would never be fully saturated using this strategy only, for two reasons: *i*) the first element next to the heat sink always has its largest base outside the domain Ω and *ii*) the random nature of the GA is likely to allocate some parts of the L-system structure outside the domain Ω (see Fig. 5). In order to saturate the volume constraint, we add a buffer layer around the west and south boundaries, defining a new domain β and such that $\beta \cap \Omega = \varnothing$. As shown in Fig. 5, this allows us to define two new domains:

- $\beta_p = L \cap \beta$, by analogy with the domains Ω_p and Ω. Note that $\beta_p \cap \Omega_p = \varnothing$ by construction and that β_p is not necessarily connected.
- $\alpha_p = L \setminus (\Omega_p \cup \beta_p)$, which are the parts of *L* outside both domains Ω and β.

Then, we compute the appropriate correction factor *c* using the following steps:

1. filtering of trapezoidal elements, the width or length of which is ten times smaller than the width of a single
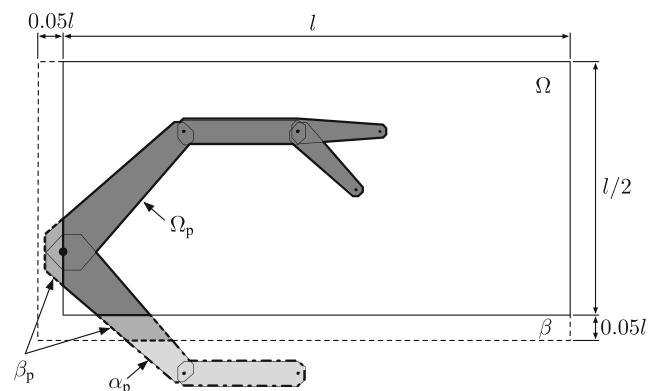

Fig. 4 Definition and mapping of an element *i* on a Cartesian grid

Fig. 5 Domain Ω and its buffer layer β located at its west and south boundaries. The L-system structure defines a domain *L* such as $L = \alpha_p \cup \beta_p \cup \Omega_p$ and $\alpha_p \cap \beta_p \cap \Omega_p = \varnothing$

cell in the Cartesian design grid. This step speeds up the mapping process without loss of accuracy.

2. finding c' such that $|L(c')| = \phi|\Omega|$.
3. finding c such that $|\Omega_p(c)| = |\Omega_p(c') \cup \beta_p(c')|$.

Using this approach, the saturation level of the volume constraint is the following:

– if $\alpha_p(c') = \varnothing$, then $|\Omega_p(c)| = \phi|\Omega|$, meaning that the volume constraint is saturated,
– if $\alpha_p(c') \neq \varnothing$, then $|\Omega_p(c)| < \phi|\Omega|$, meaning that the volume constraint is not saturated. Consequently, the objective function is penalized, since not all the available high-conductive material is exploited in the design domain.

One advantage of this method is that it naturally penalizes L-system structures that are not fully enclosed within $\Omega \cup \beta$.

Once the appropriate correction factor c has been determined, the last operation is to map the scaled L-system elements included within the domain Ω to the design grid, which is a Cartesian grid made of $N_x \times N_y$ square cells. In other words, we identify the design cells with centers lying inside the scaled L-system structure Ω_p. This could be done by invoking a *Shapely* routine (which checks if a point is inside a polygon). However, we observed this approach to be inefficient from a computational point of view, mainly due to the complexity of the domain Ω_p. Consequently, we have implemented another approach where we map each trapezoidal element, intersecting with Ω, separately using following steps:

1. we identify the design cells belonging to the bounding box of the trapezoidal element i (see Fig. 4),
2. for each center point of these cells, generically denoted as P,

   (a) we compute the non-dimensional abscissa s of its projection along the trapezoidal middle line

   $$s = \frac{\overrightarrow{P_{i,0}P} \cdot \overrightarrow{u}}{||\overrightarrow{P_{i,0}P}||}, \tag{18}$$

   where $\overrightarrow{u}$ is the unit vector between the points $P_{i,0}$ and $P_{i,1}$. If $0 \leq s \leq 1$, the projection of P is between the points $P_{i,0}$ and $P_{i,1}$ and,

   (b) we compute the width $d_i(s)$ of the trapezoidal element i at the abscissa s using equation

   $$d_i(s) = \frac{c}{2}\left(w_{i,0} + s(w_{i,1} - w_{i,0})\right). \tag{19}$$

   (c) Finally, we assign the design cell to the domain Ω_p, if $0 \leq s \leq 1$ and $||\overrightarrow{P_{i,0}P} \times \overrightarrow{u}|| \leq d(s)$. Otherwise, we assign it to the domain Ω_0.

Moreover, we omit the testing of design cells that have already been assigned to the domain Ω_p when mapping an earlier trapezoidal element. The tips of an element, shown in Fig. 4, are mapped using the same method (each tip also has a trapezoidal shape). Finally, if the aspect ratio of a trapezoidal element i is large, it is divided into several sections in order to reduce the size of the bounding boxes to test (and consequently the number of empty cells).

3.3 Finite volume method

Once the design cells belonging to the domain Ω_p have been identified, the objective function must be evaluated. In order to do so, the temperature field over the domain Ω is computed thanks to the FVM (Patankar 1980). The technical implementation in the frame of this problem is fully detailed in the paper by Marck et al. (2012). However, it is worth noting that a staggered scheme is used between the design grid (*solid* lines in Fig. 6) and the FV one (*dash* lines in Fig. 6), which is a different approach from Boichot and Fan (2016) who implemented a centered Finite Difference scheme. The staggered scheme has been selected because of two reasons:
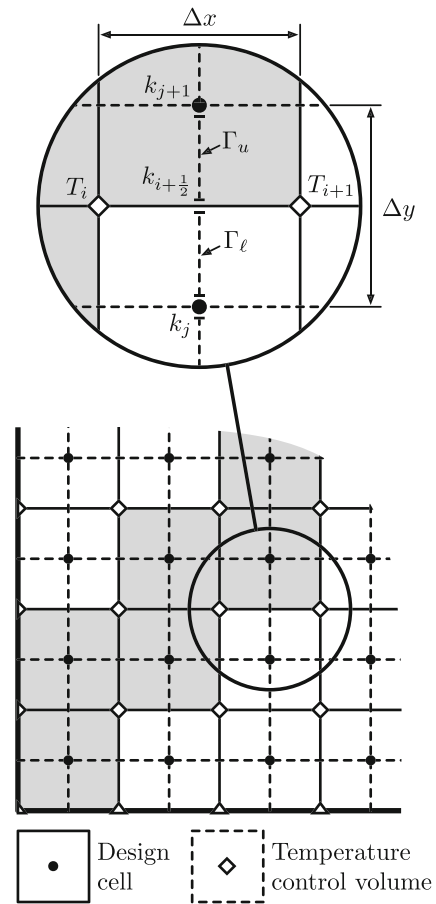

Fig. 6 Design and Finite Volume grids with a staggered scheme. The FV center matches the boundary conditions, but the thermal conductivity $k_{j+\frac{1}{2}}$ at the interface between two control volumes must be evaluated using a numerical filter

– As underlined by Patankar (1980), in such configuration the center of the FV elements spatially matches the boundary conditions (see Fig. 6). Consequently, setting the Dirichlet boundary condition for the heat sink is numerically straightforward, while ensuring an accurate result.

– The SIMP method widely uses staggered schemes, in order to avoid the so-called checkerboard problem. We benchmark most of our results against SIMP results and, thus, using the same discretization and solver allows more accurate comparisons.

Using a staggered scheme in conjunction with the FVM requires an additional filtering step in order to compute the thermal conductivity at the interface between two adjacent control volumes (see the papers by Gersborg-Hansen et al. (2006) and Marck et al. (2012) for further explanations). Figure 6 shows an example between two horizontal control volumes bounding the temperatures T_i and T_{i+1}: the cells holding the design information required to compute the thermal conductivity $k_{j+\frac{1}{2}}$ at their interface are located above and below them. Following the approximation caused by the FVM and deriving the heat flux along the intersecting boundaries Γ_u and Γ_ℓ, it can be shown that the energy quantity Q exchanged by both control volumes is

$$Q = -\Delta y \frac{k_j + k_{j+1}}{2} \frac{\partial T}{\partial x} \qquad (20)$$

where k_j and k_{j+1} are the thermal conductivities linked with their corresponding design cells (respectively k_0 and k_p in this example). Consequently, this leads us to consider an arithmetic average of the thermal conductivity along the control volume boundaries, also known as the Voigt average

$$k_{i+\frac{1}{2}} = \frac{k_j + k_{j+1}}{2}. \qquad (21)$$

This result is the opposite of the one reached for heat conduction using collocated grids, where the harmonic average (or Reuss average) provides the adequate conductance at the control volume interfaces (Patankar 1980).

Figure 7 provides an example of the whole process, from the L-system definition and scaling (upper part) to the mapping and the objective function evaluation (lower part). The mapping step, that is projecting the L-system onto the Cartesian grid, takes approximatively the same amount of time as solving the temperature. Depending on the L-system complexity, evaluating the fitness of one individual from its genotype takes between 0.2 to 4 seconds using a single Central Processing Unit (CPU) (the longest evaluation times typically occur during the first generation of the a GA process). We execute optimization runs in parallel using 16 CPUs.
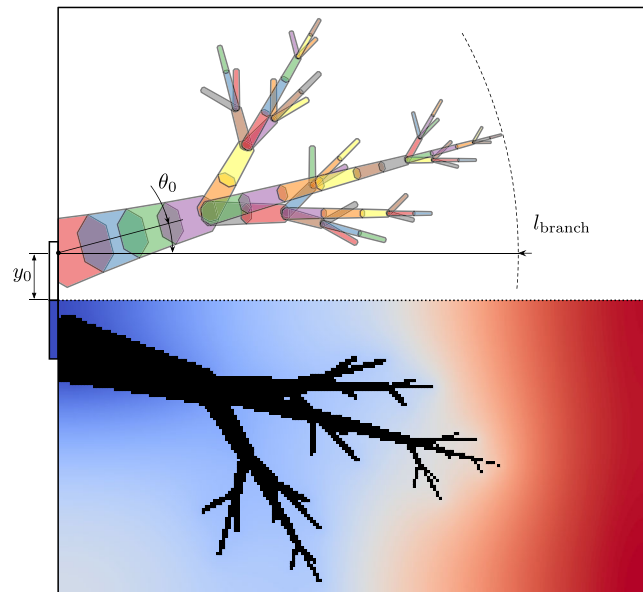

Fig. 7 **Upper part** - trapezoidal elements composing the L-system structure based on the modified example of Prusinkiewicz and Lindenmayer (2012) (see Fig. 3d). The influence of additional variables y_0, θ_0 and l_{branch} is shown as well. **Lower part** - L-system mapping (in *black*) and the corresponding temperature field computed thanks to the FVM (in *red* and *blue*)

4 Results

In this section, we present, first, the results of single-objective optimization, which we benchmark against the most relevant studies in the dedicated scientific literature, and, second, demonstrate the suitability of the methodology to tackle multi-objective problems.

4.1 Single-objective optimization

Let us first select relevant reference results from the literature. As we mentioned in the introduction, Boichot and Fan (2016) show that their GA-based algorithm yields lower non-dimensional thermal resistances than the studies using cellular automata, constructal theory and ESO (see references in the introduction). From this point, this is refered as the direct encoding method. The non-dimensional thermal resistance, specific to a reference temperature T_{ref}, is defined as

$$R_{\{T_{\text{ref}}\}} = \frac{T_{\text{ref}} - T_{\text{sink}}}{q_0 A / k_0}, \qquad (22)$$

where T_{sink} is the temperature of the heat sink, q_0 is the heat generation rate within the domain Ω_0, and A is the area of the domain[7] (Bejan 1997). The reference temperature T_{ref} is either $\bar{T}$ or T_{max}, depending on the objective function

---

[7]In our case, the design domain has a square shape, and therefore $A = l^2$.

studied. The purpose of this non-dimensional thermal resistance is to enable comparison of optimized designs with different problem parameters such as q_0, A, k_p/k_0 or ϕ.

Boichot and Fan (2016) also indicate that their results are similar to those obtained by Marck et al. (2012) using the SIMP method. Therefore, we choose to benchmark our results against these two studies.

Looking more closely into the comparison by Boichot and Fan (2016), the comparability of non-dimensional thermal resistances between these two studies is in fact limited, due to the following reasons. First, Marck et al. (2012) define the heat generation to occur in both domains Ω_0 and Ω_p, whereas Boichot and Fan (2016) set it only within the domain Ω_0. Second, the heat sinks have different sizes; Marck et al. (2012) define it to be 1% of the left boundary, whereas Boichot and Fan (2016) define it to be 20% (of the same boundary). Third, the discretizations of the design domain are different. Marck et al. (2012) used a staggered grid of 200×100 elements, whereas Boichot and Fan (2016) used a centered grid of 100×50 elements.

To ensure a fair comparison between the three methods, we test our L-systems-based method, presented in this work, on the same optimization problems that were studied by Boichot and Fan (2016) and generate the corresponding results using the SIMP method implemented by Marck et al. (2012). In particular, the sensitivity filter used is based on the same discrete convolution product between sensitivity and design fields, with a radius of $1.25\Delta x$ (or $1.25\Delta y$). This value has proven to be a suitable trade-off avoiding the so-called checkerboard problem, while allowing the formation of sufficiently thin structures at the extremities of the high conductivity network. The optimization problems are defined based on the objective function, the conductivity ratio k_p/k_0 and the volume fraction ϕ (see Table 3), while the other problem parameters are kept constant: $q_p = 0$ W/m^2, $q_0 = 10$ kW/m^2, $l = 0.1$ m, $d = 0.2l$ and $k_0 = 1$ W/(mK).

Depending on the method under consideration, the grid does not have the same size:

- as mentioned earlier, the direct encoding results reached by Boichot and Fan (2016) carry out 100×50 centered grids,
- in the analysis using the L-systems-based approach, we use a staggered grid of 200×100 elements, providing a suitable trade-off between the design accuracy and a fast mapping,
- SIMP method uses a 400×200 grid. Indeed, this approach required a filtering step in order to avoid the so-called *checkerboard* problem, which artificially aggregates high-conductivity elements together. Consequently, the thinest branches that the SIMP method is able to produce have the same width as the one coming from the L-system approach, ensuring a meaningful comparison between both designs.

Finally, after convergence, all the optimized designs (including the ones obtained by Boichot and Fan 2016) are evaluated using the same staggered grid of 800×400 elements and the same FVM solver (see Section 3.3), to ensure a fair comparison between each method. Increasing the number of FVs allows reducing the numerical errors coming from the discretization scheme. Therefore, for the same test case, the differences between the objective function value of each method do not come from some numerical artifacts, but only from the differences in the designs. This involves an additional method able to map results from the coarsest grids (that hold 100×50, 200×100 or 400×200 elements) to the finest one (800×400): it is capable of remeshing any design with a superior number of elements, increasing the initial cell number by the same multiple of two in each direction (x and y).

Figure 9 shows the design comparison of optimization problems #1-6, where the objective is to minimize the average temperature $\bar{T}$. In these optimization problems, the SIMP method (Marck et al. 2012) yields optimized designs with 2.7 to 20.6% lower average temperatures than the direct encoding method (Boichot and Fan 2016). Therefore, we normalize all results in the figure with respect to those obtained by the SIMP method. In fact, the results of the direct encoding method become increasingly worse in comparison to the SIMP method when $\phi k_p/k_0$ increases. The corresponding numerical values are listed in Fig. 10j.

As the L-systems-based method is stochastic,[8] we have repeated the optimization runs 30 times for each problem. An example set of convergence histories from these runs, for the optimization problem #2, is illustrated in Fig. 8, along with the distribution of optimized objective function values. In Figs. 9 and 10, we report the mean of the optimized objective function values and its 95% confidence

Table 3 Identification of the selected test cases and their k_p/k_0 and ϕ parameters: problems #1-6 minimize the average temperature (*left-hand* side), whereas problems #7-12 minimize the maximum temperature (*right-hand* side)

$f(\chi_{\Omega_p})$	#	k_p/k_0 [–]	ϕ [–]	#	$f(\chi_{\Omega_p})$
$\bar{T}$	1	2	0.3	7	T_{max}
	2	10	0.1	8	
	3	10	0.3	9	
	4	10	0.5	10	
	5	50	0.3	11	
	6	250	0.3	12	

---

[8]The GA-based algorithm by Boichot and Fan (2016) is also stochastic, but they only provide between one and three solutions for each case.

Fig. 8 The convergence history of 30 optimization runs and the distribution of the optimized objective function values for problem #2 are plotted on the *left-hand* side. On the *right-hand* side plot, the histogram represents the actual distribution of the results, whereas the continuous line an estimate of their normal distribution
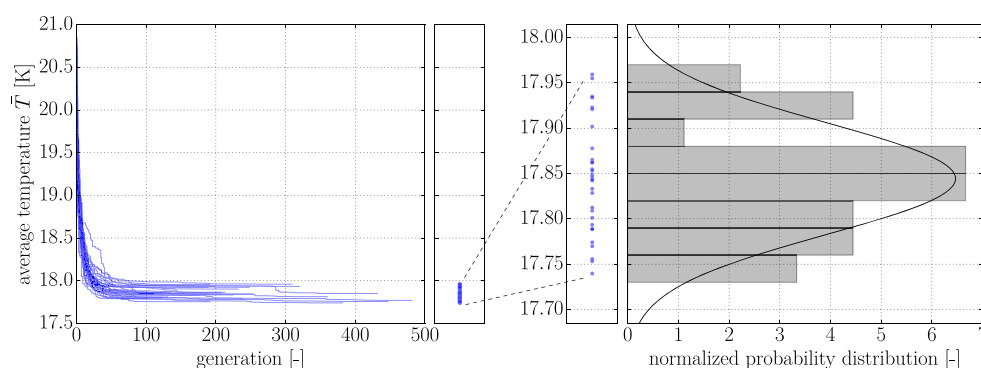
interval, calculated by multiplying the standard error by 1.96. The L-systems-based method yields on average better designs than the direct encoding method (Boichot and Fan 2016) in problems #1-6. These conclusions are statistically significant.

The effectiveness of L-systems-based method against the SIMP method seems to be dependent on the dimensionless coefficient $\phi k_p/k_0$ (Fig. 9). When $\phi k_p/k_0 \leq 1$ (optimization problems #1-2), the L-systems-based method yields lower objective function values than the SIMP method, whereas, when $\phi k_p/k_0 \geq 3$ (optimization problems #3-6), the optimized objective function values are higher. Looking at the optimized designs in Fig. 10, the complexity of the designs seems to be related to the dimensionless coefficient $\phi k_p/k_0$. We name two potential reasons why the L-systems-based method cannot find as good designs as the SIMP
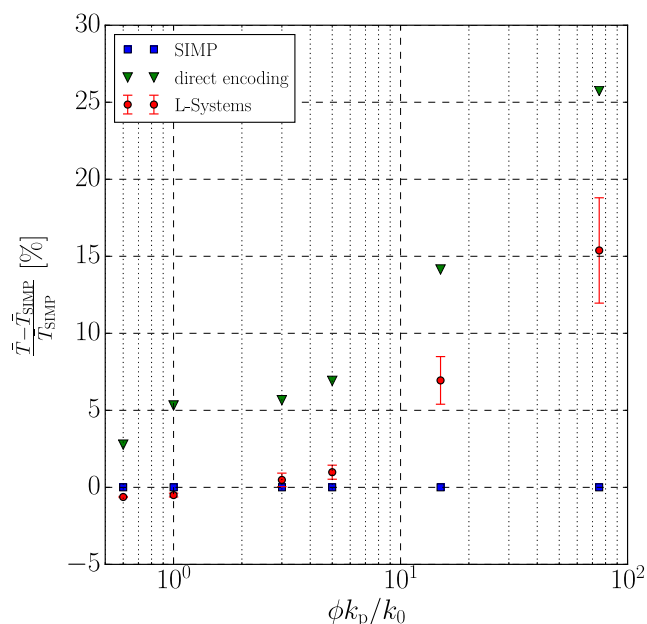
method in optimization problems where $\phi k_p/k_0 \geq 3$: 1) the parameterization is not flexible enough to define designs with required geometrical complexity (see Fig. 10f as a reference) and/or 2) the method fails to fine-tune the details of these designs as it does not use the gradient information of the objective function.
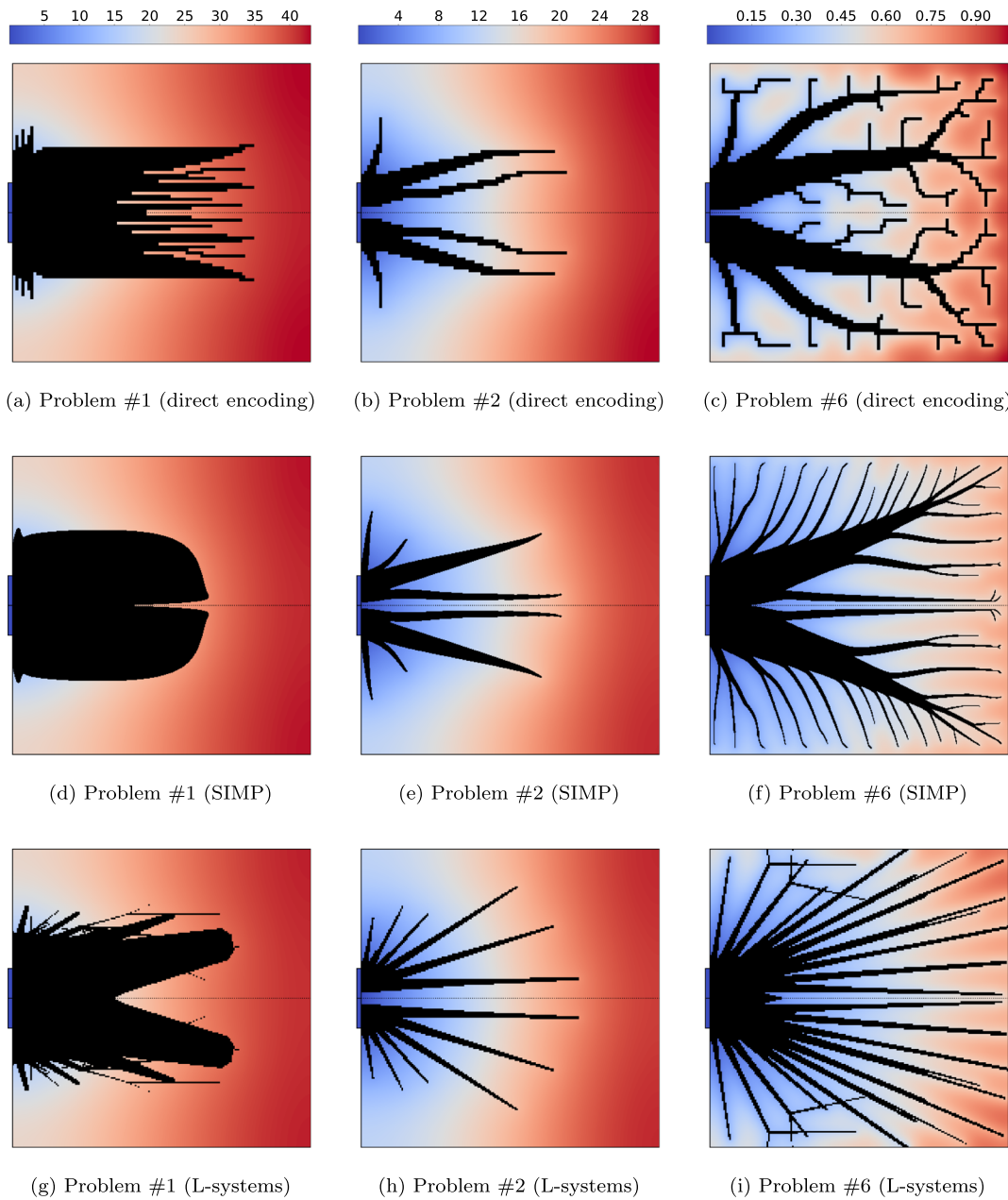
Figure 10 presents a comparison of optimized designs for three representative optimization problems (#1, #2 and #6).[9] In optimization problem #1 (Subfigures a, d, g), the L-systems-based method yields a design where the North and South boundaries of the high conductive material are clearly coarser than in the reference designs.

In optimization problem #2 (Subfigures b, e, h), all methods yield designs where the high-conductive material is distributed in patterns having only radial branches starting from the heat sink, with no bifurcations in the outmost regions of the domain. However, the numbers of radial branches in these designs are different, ranging from 6 to 12 – the design obtained by the L-systems-based method having the most branches.

In optimization problem #6 (Subfigures c, f, i), the design obtained by the L-systems-based method has a similar radial pattern of high-conductive material, whereas the corresponding designs with the direct encoding and SIMP methods have a bifurcating tree structure with three different scales. Despite having a fundamentally different topology, the average temperature of the design by the L-systems-based method is only 1.7% higher than the corresponding design reached by the SIMP method. From a physical point of view, the radial pattern effectively directs the heat flux towards the heat sink in the entire high conductive material domain Ω_p.

Let us next examine the results for optimization problems #7-12, where the objective is to minimize the maximum temperature T_{max}. Here, we only benchmark the L-systems-based method against the direct encoding method, as the SIMP method would require transforming the min-max

Fig. 9 Benchmarking of the L-systems-based method against the direct encoding (Boichot and Fan 2016) and the SIMP method (Marck et al. 2012) for problems #1-6. $\bar{T}_{SIMP}$ is the average temperature optimized by the SIMP method

---

[9]The presented designs corresponding to the L-systems-based method are the best of 30 repeated optimization runs (the same applies later to Fig. 11).
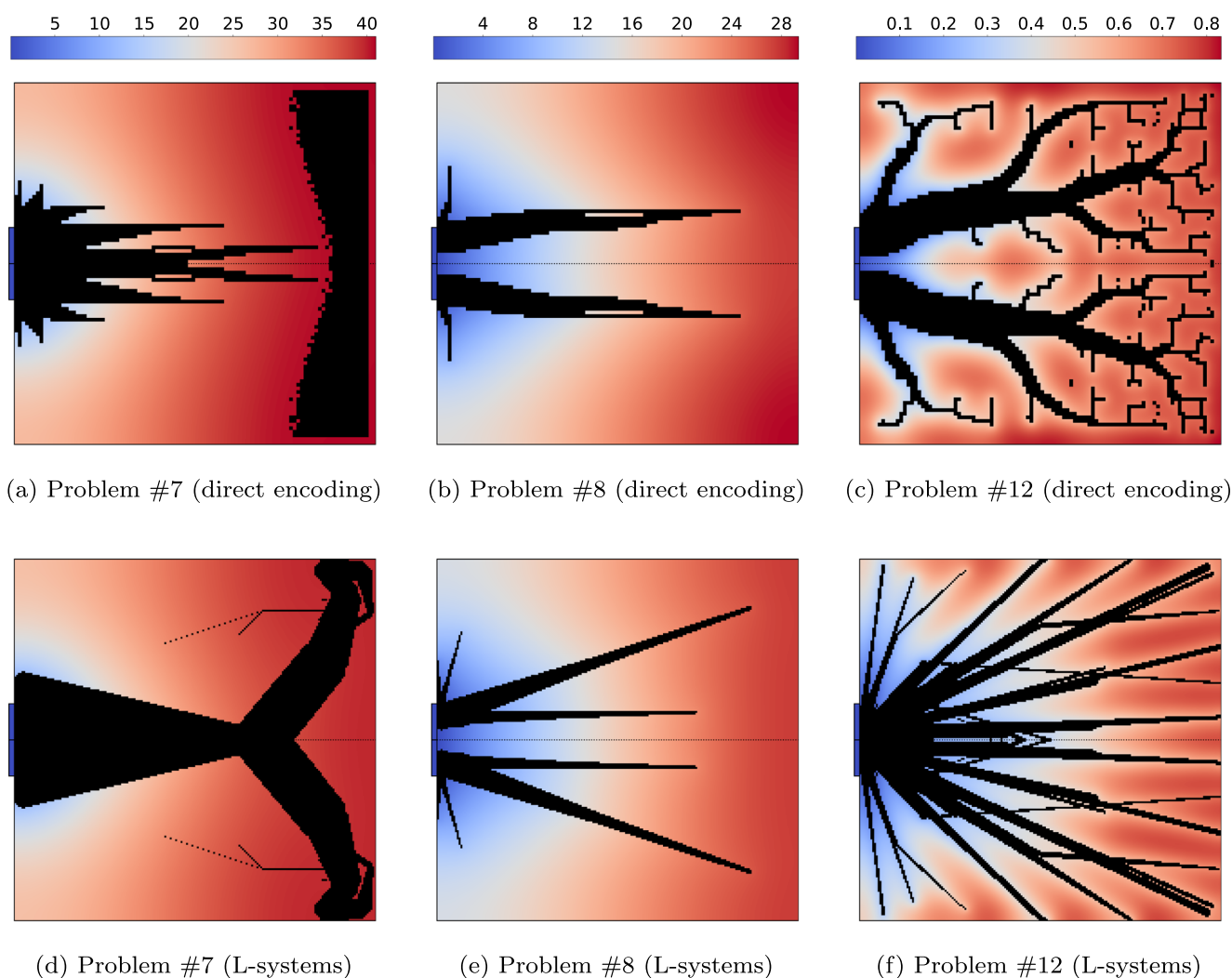
(a) Problem #1 (direct encoding) (b) Problem #2 (direct encoding) (c) Problem #6 (direct encoding)

(d) Problem #1 (SIMP) (e) Problem #2 (SIMP) (f) Problem #6 (SIMP)

(g) Problem #1 (L-systems) (h) Problem #2 (L-systems) (i) Problem #6 (L-systems)

pb.	SIMP		direct encoding			L-systems		
	$\bar{T}$ [K]	$R_{\{\bar{T}\}} \times 10^3$	$\bar{T}$ [K]	$R_{\{\bar{T}\}} \times 10^3$	$\bar{Q} \times 10^{-6}$	$\bar{T}$ [K]	$R_{\{\bar{T}\}} \times 10^3$	$\bar{Q} \times 10^{-3}$
#1	30.483	609.66	31.328	626.56	~5-7	30.292 ± 0.012	605.84 ± 0.23	30.2
#2	17.934	358.69	18.888	377.75	~5-7	17.844 ± 0.022	356.89 ± 0.44	33.9
#3	7.982	159.64	8.433	168.67	~5-7	8.020 ± 0.035	160.41 ± 0.70	41.2
#4	4.635	92.70	4.955	99.11	~5-7	4.680 ± 0.021	93.61 ± 0.42	41.6
#5	1.797	35.95	2.052	41.03	~5-7	1.922 ± 0.028	38.44 ± 0.56	36.0
#6	0.397	7.95	0.500	9.99	~5-7	0.459 ± 0.014	9.17 ± 0.27	54.7

(j) Results in numeric format, including the optimized average temperature $\bar{T}$ and the corresponding nondimensional thermal resistance $R_{\{\bar{T}\}}$, as well as the average number of required objective function evaluations, $\bar{Q}$. The results obtained by the L-systems-based method are reported as the mean and its 95% confidence interval. The estimate of required objective function evaluations of the direct encoding method is based on Section 2.4 and Fig. 2 in the paper by Boichot and Fan (2016).

Fig. 10 Comparison of designs obtained for three representative problems #1, #2 and #6 (Subfigures a-i), and the results in numerical format for problems #1-6 (Subfigure j). Animations of the design evolutions yielding the best L-systems-based designs for optimization problems #1-6 are attached as supplementary data files

(a) Problem #7 (direct encoding)

(b) Problem #8 (direct encoding)

(c) Problem #12 (direct encoding)

(d) Problem #7 (L-systems)

(e) Problem #8 (L-systems)

(f) Problem #12 (L-systems)

| pb. | direct encoding | | | L-systems | | |
	$T_{\max}$ [K]	$R_{\{T_{\max}\}} \times 10^3$	$\bar{Q} \times 10^{-6}$	$T_{\max}$ [K]	$R_{\{T_{\max}\}} \times 10^3$	$\bar{Q} \times 10^{-3}$
#7	41.029	820.58	$\sim$5-7	40.180 ± 0.093	803.59 ± 1.86	34.3
#8	29.380	587.61	$\sim$5-7	27.665 ± 0.050	553.29 ± 1.00	29.5
#9	12.508	250.17	$\sim$5-7	12.031 ± 0.105	240.61 ± 2.09	42.5
#10	6.740	134.80	$\sim$5-7	6.499 ± 0.058	129.99 ± 1.17	38.8
#11	3.089	61.78	$\sim$5-7	3.329 ± 0.101	66.58 ± 2.01	45.2
#12	0.833	16.66	$\sim$5-7	0.967 ± 0.059	19.34 ± 1.18	60.3

(g) Results in numeric format (see the caption of Fig. 10j for explanations of the symbols).

Fig. 11 Comparison of designs obtained for three representative problems #7, #8 and #12 (Subfigures a-i), and the results in numerical format for problems #7-12 (Subfigure j). Animations of the design evolutions yielding the best L-systems-based designs for optimization problems #7-12 are attached as supplementary data files

problem into a new one involving the $p-$norm operator, which is continuous and differentiable (see Yan et al. 2018). Making comparisons between both formulations would be problematic since they do not involve the same objective functions and because the solutions of the $p-$norm problem depend on the p value (that is usually selected thanks to different numerical tests). Figure 12 shows the benchmarking of the L-systems-based results against the direct encoding ones for optimization problems #7-12. The corresponding numerical data is presented in Fig. 11g. In these figures, we again report the mean and 95% confidence interval (determined based on the
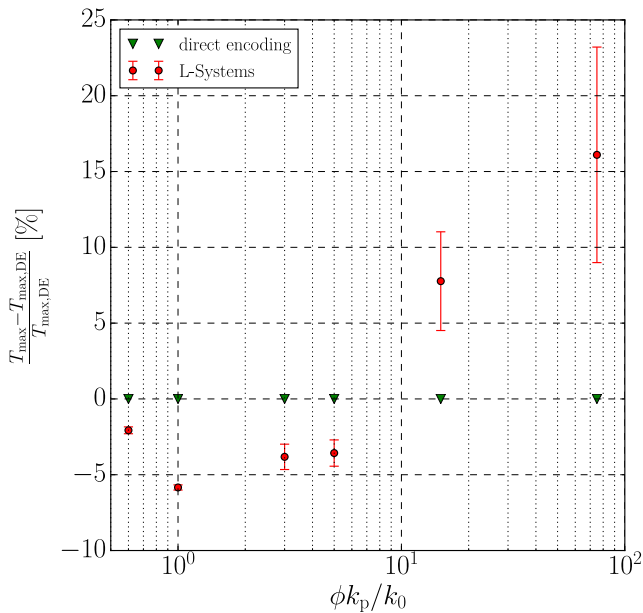
Fig. 12 Benchmarking of the L-systems-based method against the direct encoding method (Boichot and Fan 2016) for problems #7-12. $T_{\mathrm{max,DE}}$ is the maximum temperature optimized by the direct encoding method

standard error) of the results obtained using the L-systems-based method.

In optimization problems #7-10, the L-systems-based method yields on average better results than the direct encoding. In optimization problems #11 and #12, the obtained designs are on average worse than those obtained by the direct encoding. These conclusions are also statistically significant.

It is worth noticing that in optimization problems #9-12 the performance of the L-systems-based method gradually decreases against the direct encoding method as the dimensionless coefficients $\phi k_\mathrm{p}/k_0$ increases. Thus, a crossover value, of around 5 to 15, may exist for the dimensionless coefficient $\phi k_\mathrm{p}/k_0$, above which the direct encoding is, on average, more efficient than the L-systems-based method. However, such trend is here less clear than in the earlier results between the L-systems-based and the SIMP methods in Fig. 9.

Nevertheless, also in optimization problems #11 and #12, the best designs obtained by the L-systems-based method are better than those obtained by the direct encoding method; the objective function values of these designs are 2.901 and 0.747 K, respectively.

The L-systems-based method requires significantly fewer objective function evaluations than the direct encoding method (Fig. 11g) – the difference being two orders of magnitude. However, we want to point out that the convergence criteria of the algorithms are different and the

reporting of the number of required objective function evaluations in the reference study (Boichot and Fan 2016) is limited. Nevertheless, even if the optimization problem #12 is the most demanding from a computational point of view, the entire set of 30 optimization runs requires $30 \times 60.3 \cdot 10^3 \approx 1.8 \cdot 10^6$ function evaluations, which is only around 36% of a single optimization run with the direct encoding. If we consider the set of optimization runs as a multi-start approach, the L-systems-based method also yields a better result than the direct encoding method for problems #11 and #12.

We believe that there are two main reasons why the L-systems-based method outperforms the direct encoding method. First, the designs space of the method is channeled to favorable designs, in which the entire material distribution is fully connected and touches the heat sink. Second, as we mentioned in the introduction, L-systems (like other generative encodings) are construction recipes, which can be used to define diverse design spaces with relatively few design variables and are capable of producing designs consisting of self-similar and hierarchical components.

An example of self-similarity can be seen, for example, in the design in Fig. 10h. Considering either side of the symmetry axis, the material distribution of this design consists of two compositions of three radial spikes. These compositions are similar to each other, but of different scales.

Designs obtained by the L-systems-based and direct encoding methods are shown in Fig. 11a–i for problems #7, #8 and #12. These problems have the same conductivity ratio k_p/k_0 and volume fraction ϕ as problems #1, #2 and #6, respectively, which results were presented earlier in Fig. 10a–i.

In problems #8 and #12, the L-systems-based method also produces designs where most of the high conductive material is distributed in patterns having only radial branches (Fig. 11e and f). However, these branches penetrate deeper in the finite-size volume and their tips are thicker than in Fig. 10h and i, mitigating high temperatures in the outmost regions of the domain, where the temperature increase is the most critical.

On the other hand, in problem #7, the obtained design with the L-systems-based method (Fig. 11d) is significantly different to the corresponding design minimizing the average temperature $\bar{T}$ (Fig. 10g). As the conductivity ratio k_p/k_0 is low, the critical regions for the maximum temperature are located at the two corners furthest away from the heat sink, which the method seeks to fill with high conductivity material.

4.2 Multi-objective optimization

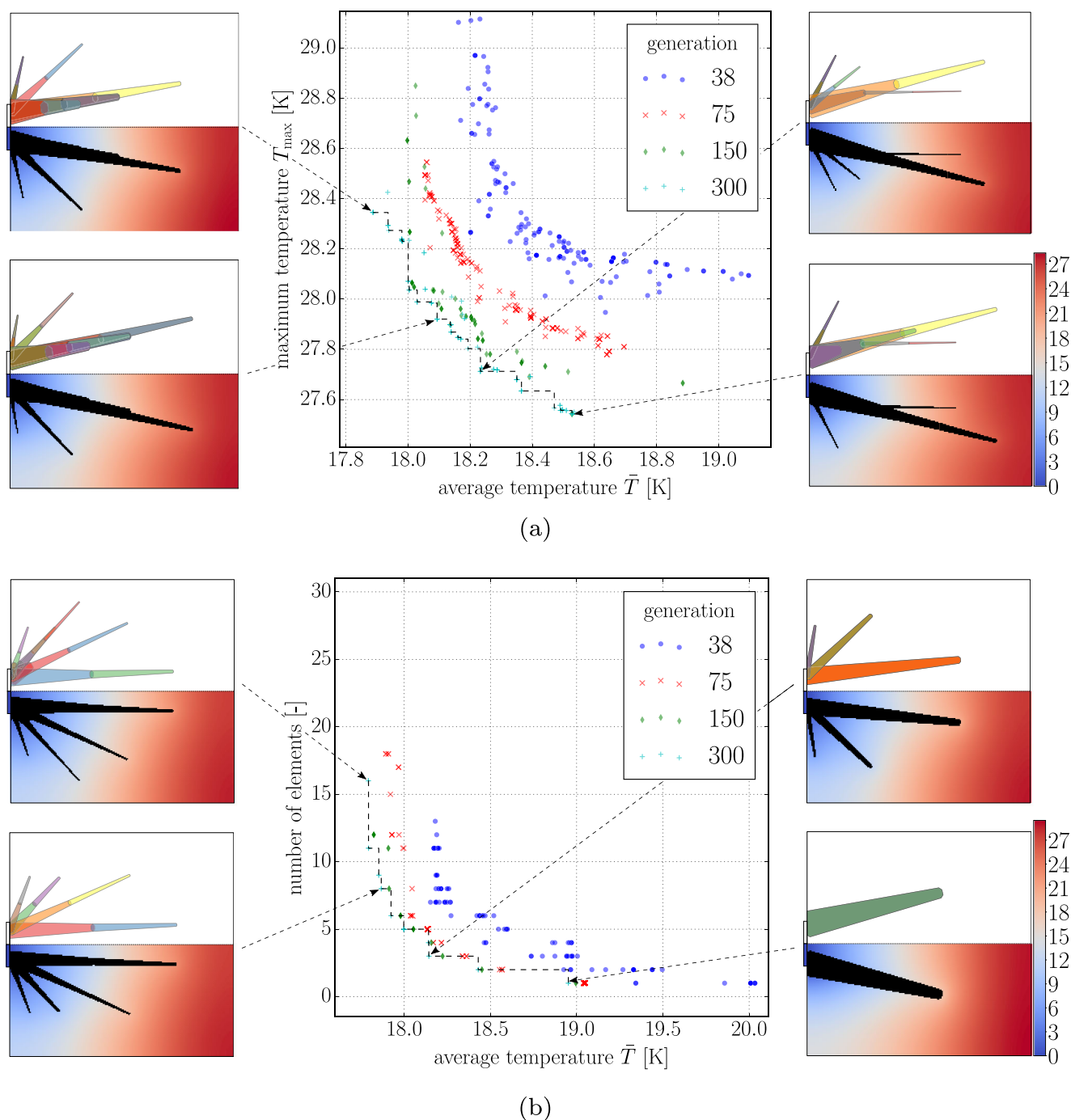The design of realistic engineering systems often quickly becomes multi-objective. Therefore, in this section, we

(a)

(b)

Fig. 13 Results of two multi-objective optimization runs, evolved via NSGA-II. The figures on the sides show representative designs from the Pareto front, indicated by the dashed line

demonstrate the suitability of the L-systems-based method to tackle multi-objective design optimization of heat conductors with both scalar and integer objectives. The purpose is to obtain a set of Pareto optimal designs in the objective space, which represent the best trade-offs between two competing objectives.

As the optimization algorithm, we here apply the NSGA-II (elitist non-nominated sorting genetic algorithm) by Deb et al. (2002). We use the same control parameters as in the single-objective optimization (see Table 2) with slight modifications. The tournament pool size is changed into two, as defined by Deb et al. (2002). The implementation

of NSGA-II in DEAP requires the population size to be a multiple of four (Fortin et al. 2012), so we change it to be 152. Finally, we terminate an optimization after 300 generations.

Let us first examine an optimization run where the objectives are to concurrently minimize the average and maximum temperatures, $\bar{T}$ and $T_{\max}$, that were individually minimized earlier in Section 4.1. We define the conductivity ratio k_p/k_0 and volume fraction ϕ to be the same as in the optimization problems #2 and #7. Figure 13a illustrates the bi-objective optimization process of approaching the Pareto front, as well as representative designs lying at the approximated Pareto front. The designs lying at the ends of the approximated Pareto front compare well with corresponding single-objective results. Thus, the entire Pareto front represents designs that could be considered by a chip manufacturer who would like to reach an average temperature as low as possible, while reducing the hot spots over the component.

Finally, let us dive deeper into the mindset of the hypothetical chip manufacturer. After seeing some of the designs in Figs. 10 and 11, he or she might question whether there is any compromise design that provides a good heat transfer with lesser geometrical complexity. The L-systems-based method provides one possible measure for 'design complexity': the number of steps taken by the turtle. Let us refer to this measure as the number of elements. Figure 13b shows a bi-objective version of solving the optimization problem #2, using the number of elements as the second objective. Clearly, the resulting designs can be chosen to be much simpler in shape, albeit at the expense of conductive performance.

5 Discussion

Evolutionary optimization methods (non-gradient-based methods) using the direct encoding have not gained significant acceptance in the topology optimization community. Munk et al. (2015) list two partial reasons for this, which are the difficulty of ensuring structural connectivity and the excessive use of computational resources. In addition, Sigmund (2011) indicates that non-gradient-based methods require orders of magnitude more function evaluations in comparison to gradient-based methods. The evolutionary topology optimization method, we use in this paper, naturally produces individuals with structural connectivity and requires two orders of magnitude fewer function evaluations than the direct encoding method.

Despite having the flexibility of generating branching tree-structures, the L-systems-based method yielded mostly designs, in which the dominant feature is the radial pattern of high conductive material (see Figs. 10h, i, 11e and f). This observation is in line with the conclusions of the recent study by Yan et al. (2018). The authors

initiated the SIMP method from rank-1 laminates, which resulted in radial material distributions they refer to as *lamellar needle structures*. The authors showed that these structures have lower average and maximum temperatures than branching tree-structures, typically considered as the optimal structural type in the literature, in several test cases.

The interpretation of L-systems into three-dimensional geometries is already an established method in computer graphics to represent biological organisms (Prusinkiewicz and Lindenmayer 2012). Thus, the methodology presented here is extensible to three-dimensional topology optimization, simply by adding two new symbols to the alphabet Σ and the corresponding numerical representation. These symbols command the turtle to pitch up or down or roll with respect to its previous heading.

The method is also applicable to fields outside thermal systems, such as urban planning or designing escape routes in music festival areas, airports or large sports arenas, in other words, in any circumstances that involve volume-to-point or area-to-point problems (Gersborg-Hansen et al. 2006).

6 Conclusions

In this paper, we presented a generative encoding method, based on parametric L-systems, for evolutionary topology optimization of heat transfer structures. When evolved by a GA, the encoding method requires two orders of magnitude fewer function evaluations than the reference method (Boichot and Fan 2016) using direct encoding. We obtain statistical significance that the method yields better results than the direct encoding method in 10 out of 12 tested optimization problems. Further, our results indicate that the method yields lower objective function values than the widely used and well established SIMP method in optimization problems, the dimensionless coefficient $\phi k_p/k_0$ of which is less or equal to 1. The method is suitable for both single and multi-objective optimization, involving scalar and/or integer objective functions.

References

Bejan A (1997) Constructal-theory network of conducting paths for cooling a heat generating volume. Int J Heat Mass Transfer 40(4):799,813–811,816

Bendsøe M. P. (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202

Boichot R, Fan Y (2016) A genetic algorithm for topology optimization of area-to-point heat conduction problem. Int J Therm Sci 108:209–217

Boichot R, Luo L, Fan Y (2009) Tree-network structure generation for heat conduction by cellular automaton. Energy Convers Manag 50(2):376–386

Burger FH, Dirker J, Meyer JP (2013) Three-dimensional conductive heat transfer topology optimisation in a cubic domain for the volume-to-surface problem. Int J Heat Mass Transf 67:214–224

Cheng X, Li Z, Guo Z (2003) Constructs of highly effective heat transport paths by bionic optimization. Sci China Series E: Technol Sci 46(3):296–302

Chomsky N (1956) Three models for the description of language. IRE Trans InformationTheory 2(3):113–124

Dbouk T (2017) A review about the engineering design of optimal heat transfer systems using topology optimization. Appl Therm Eng 112:841–854

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dede EM (2009) Multiphysics topology optimization of heat transfer and fluid flow systems. In: Proceedings of the COMSOL users conference

Dirker J, Meyer JP (2013) Topology optimization for an internal heat-conduction cooling scheme in a square domain for high heat flux applications. J Heat Transfer 135(11):111,010

Fortin FA, De Rainville FM, Gardner MA, Parizeau M, Gagné C (2012) DEAP: evolutionary algorithms made easy. J Mach Learn Res 13:2171–2175

Gao T, Zhang WH, Zhu JH, Xu YJ, Bassir DH (2008) Topology optimization of heat conduction problem involving design-dependent heat load effect. Finite Elem Anal Des 44(14):805–813

Gersborg-Hansen A, Bendsøe MP, Sigmund O (2006) Topology optimization of heat conduction problems using the finite volume method. Struct Multidiscip Optim 31(4):251–259

Ghodoossi L, Eğrican N (2003) Exact solution for cooling of electronics using constructal theory. J Appl Phys 93(8):4922–4929

Gillies S et al (2007) Shapely: manipulation and analysis of geometric objects. https://github.com/Toblerity/Shapely

Guo T, Lohan DJ, Cang R, Ren MY, Allison JT (2018) An indirect design representation for topology optimization using variational autoencoder and style transfer. In: 2018 AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference, p 0804

Hornby GS, Pollack JB (2001) The advantages of generative grammatical encodings for physical design. In: Proceedings of the 2001 congress on evolutionary computation, vol 1. IEEE, pp 600–607

Hornby GS, Pollack JB (2002) Creating high-level components with a generative representation for body-brain evolution. Artif Life 8(3):223–246

Ikonen TJ, Sóbester A (2018) Statistical analysis of control parameters in evolutionary map L-systems-based topology optimization. Struct Multidiscip Optim 58(3):997–1013

Kingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv:1312.6114

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

Kobayashi MH (2010) On a biologically inspired topology optimization method. Commun Nonlinear Sci Numer Simul 15(3):787–802

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

Li Q, Steven GP, Querin OM, Xie YM (1999) Shape and topology design for heat conduction by evolutionary structural optimization. Int J Heat Mass Transf 42(17):3361–3371

Lindenmayer A (1968a) Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. J Theor Biol 18(3):280–299

Lindenmayer A (1968b) Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. J Theor Biol 18(3):300–315

Lohan DJ, Dede EM, Allison JT (2017) Topology optimization for heat conduction using generative design algorithms. Struct Multidiscip Optim 55(3):1063–1077

Marck G (2012) Topological optimization of heat and mass transfer (application to heat exchangers). Ph.D. thesis, MINES ParisTech 5 rue Léon Blum, 91120, Palaiseau, France

Marck G, Nemer M, Harion JL, Russeil S, Bougeard D (2012) Topology optimization using the simp method for multiobjective conductive problems. Numer Heat Transfer Part B: Fund 61(6):439–470

Mathieu-Potvin F, Gosselin L (2007) Optimal conduction pathways for cooling a heat-generating body: a comparison exercise. Int J Heat Mass Transfer 50(15):2996–3006

Munk DJ, Vio GA, Steven GP (2015) Topology and shape optimization methods using evolutionary algorithms: a review. Struct Multidiscip Optim 52(3):613–631

Patankar S (1980) Numerical heat transfer and fluid flow. Series in computational methods in mechanics and thermal sciences. Taylor & Francis

Pedro H, Kobayashi MH, Coimbra CFM, de Silva AK (2008) Effectiveness of complex design through an evolutionary approach. J Thermophys Heat Transfer 22(1):115

Prusinkiewicz P (1986) Graphical applications of l-systems. In: Proceedings of graphics interface, vol 86, pp 247–253

Prusinkiewicz P, Hanan J (1990) Visualization of botanical structures and processes using parametric l-systems. In: Scientific visualization and graphics simulation. Wiley, pp 183–201

Prusinkiewicz P, Lindenmayer A (2012) The algorithmic beauty of plants. Springer Science & Business Media

Runions A, Fuhrer M, Lane B, Federl P, Rolland-Lagan AG, Prusinkiewicz P (2005) Modeling and visualization of leaf venation patterns. ACM Trans Graph (TOG) 24(3):702–711

Sabbatini E, Revel GM, Kobayashi MH (2015) Vibration reduction using biologically inspired topology optimization method: optimal stiffeners distribution on an acoustically excited plate. J Vib Control 21(7):1398–1412

Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. Struct Multidiscip Optim 43(5):589–596

Stanley KO, Miikkulainen R (2003) A taxonomy for artificial embryogeny. Artif Life 9(2):93–130

Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 24(2):359–373

Xu X, Liang X, Ren J (2007) Optimization of heat conduction using combinatorial optimization algorithms. Int J Heat Mass Transfer 50(9):1675–1682

Yan S, Wang F, Sigmund O (2018) On the non-optimality of tree structures for heat conduction. Int J Heat Mass Transf 122:660–680