Global Constructive Optimization of Vascular Systems

Manfred Georg, Horst K. Hahn*, Tobias Preussner, and Heinz-Otto Peitgen

Abstract—We describe a method for creating realistic vascular systems based on optimality principles of theoretical physiology. Our model is initialized by a complete but simple and suboptimal tree that fills an organ at a given resolution, and subject to local and global optimization techniques, which we provide. The boundary conditions are given by the position and flow distribution for all vascular end points corresponding to the tree leaves, and by the position of the vascular hilum corresponding to the tree root. Optimization is driven by intravascular volume minimization, which is one of the major principles for the design of vascular systems discussed in literature. Our algorithm is novel in that it implements topological changes, which have proven essential to the optimization process. For global optimization, we additionally propose a multi-level strategy, and present initial results. The generated models are shown to be similar to real data acquired from corrosion casts of a human liver. The presented method achieves for the first time global optimization results. The generated models are shown to be similar to real vascular systems discussed in literature. Our algorithm is novel in that it implements topological changes, which have proven essential to the optimization process.

Index Terms—Blood vessels, computational modeling, liver surgery planning, optimality principles, vascular anatomy.

I. INTRODUCTION

We pose the problem, whether the geometry of a vascular system supplying or draining an organ, such as the human liver, can be solely characterized as the minimum of a mathematical optimization principle. To this end, we will first define an appropriate optimization problem and cost function. Second, we will present numerical solutions for this problem on a local scale. Third, we will discuss the global cost minimization problem and provide an algorithm toward its solution.

The medical motivation of our work stems from the field of surgical planning. In parenchymal organs, the identification of vascular territories is crucial for providing resection proposals as well as for preoperatively estimating resection volumes and patient outcome [2], [3]. Since the resolution of in vivo vascular imaging is limited, surgery planning would directly benefit from an accurate model of individual patient vascular systems [4]. Additionally, the simulation of vascular remodeling and growth, such as after destructive surgery, would be an asset in monitoring patient recovery. A compelling example is provided by living donor liver transplant (LDLT) surgery, where a complete recovery of liver function and volume can be observed for both graft and remnant within one year [5]. Moreover, computer vascular models have been used to create virtual CT scans, which in turn are used as a basis for verifying vascular image analysis methods [6].

A very natural approach for modeling a vascular system would be to simulate its growth through angiogenesis. Such a model was developed by Szczepan and Székely [7], whose motivation was to generate realistic looking vascular patterns with high graphical fidelity for surgical simulation. Although their work shows good results for capillary systems, because of its reliance on simulation of a detailed capillary bed, and for macroscopic structures, it is at present computationally too expensive to be applied to the three-dimensional vascular system of an entire organ. The same is true for earlier and more fundamental work, such as performed by H. Meinhardt or F. Nekka et al. While the latter builds upon modeling the phenomena of growth, branching, and anastomosis [8], the former aims toward explaining tree- and net-like structures from the basic principles of activation, inhibition, and elongation [9].

However, if we make the assumption that vascular systems, regardless of how they are formed, tend toward a state of optimality, then we can bypass vascular growth and directly model a fully formed vascular system. This approach is not novel; in fact, C. D. Murray’s famous remark in a 1926 paper [10], is to consider the work required to produce physiological systems. Twelve years earlier, the principle of minimum work in the context of hemodynamic research, which Murray later refined, had already been published by W.R. Hess [11]. Following Murray’s advice, we encounter much literature on the optimality principles of vascular systems. Kamiya and Togawa presented papers analyzing and extending the Hess-Murray law [12], [13]. As a specific result, they derived the minimization of intravascular volume as a general design principle of vascular systems. More generally, optimality in the context of a teleonomical principle was discussed in the seminal work of R. Rosen [14].

As an application of optimality principles, we find rule based approaches, such as the lung model produced by H. Kitaoka et al. with intriguing results [15]. A very similar algorithm was later described by M.H. Tawhai et al. [16]. In their works, a realistic model for the bronchial tree is constructed deterministically by a fixed set of rules, starting from a first few predefined bifurcations and the shape of the organ to be supplied. The rules steer volume partitioning, branching angles, radius and length ratios, and are applied recursively until a lower threshold cell size is reached. Tawhai
extends Kitaoka’s work by introducing average dimensions and order distributions in order to statistically steer the branching process. The advantage of such an approach lies in its computational efficiency and in the direct correspondence of the applied rules and the generated result. However, the produced model is only realistic in so far as the set of underlying rules explicitly create realism. We wish to avoid such limitation by only indirectly influencing the shape of a vascular system through a more generic use of optimality principles.

A less explicit but still computationally feasible approach called Constrained Constructive Optimization (CCO) has been proposed by Schreiner et al. [17], [18]. In this approach, a vascular tree is approximated by a collection of connected, straight, cylindrical tubes, where hydrodynamic resistance is modeled by Hagen-Poiseuille’s law assuming laminar flow of a perfect fluid. The model is built by iteratively adding terminal segments, one at a time, to a model following certain optimality criteria, which will be introduced in Sec. III-B. We have taken this model as our starting point and extended several of its aspects. For example, while CCO only locally optimizes the vascular tree after adding a node, we will try to optimize the entire tree as a whole.

A very similar technique to CCO was developed by Krętowski et al. to create two complementary vascular trees, which are joined to form a vascular system [6]. In this model, the problem of collisions between the two trees to avoid arterio-venous malformation is dealt with explicitly. Similarly, the capillary model by Szczerba and Székely also deals with collisions explicitly [7]. Surprisingly, our model ignores the problem of collisions as will be discussed in Sec. VI.

In this paper, we use much the same assumptions as are used in CCO [17]. That is, we also model a vascular system as a connected series of cylindrical tubes in which fluid can be modeled using Hagen-Poiseuille’s law. We further assume that optimality criteria govern the growth of vascular systems. Further assumptions will be detailed in Sec. III-A. It is important to note that, unlike Schreiner et al. [17] and Kitaoka et al. [15], during optimization we do not constrain any of the local bifurcation angles or asymmetry of the vascular tree.

Our general aim is to analyze whether individual vascular systems can be explained by fundamental optimality principles, and to show the extent to which this is possible. Our specific goal is to present a computationally efficient basic algorithm for the construction of a globally optimized vascular system and to show initial results.

The paper is organized as follows. We will first introduce the notation required for a precise algorithmic description. Then, we will define the optimization problem and its boundary conditions. Our optimization approach will be first described locally and then extended to the global problem using a multi-level strategy. A crucial ingredient to the effective local optimization will be topological changes, namely merging and splitting of nodes. Based on numerical and graphical results, we will discuss the properties of our algorithm and compare it to related work. The human liver, for which we have available high-resolution data from corrosion casts, serves as an example and gold standard (cf. Fig. 1). The last section describes possible extensions of our algorithm and future work. Finally, the fundamental question will be raised, whether the phenomenon of interdigitation can be explained by optimality principles alone.

II. NOTATION

In the following, we will introduce some useful notation for the exposition of the rest of this paper. We work in the two or three-dimensional space $\mathcal{V} := \mathbb{R}^d$ for $d = 2$ or $d = 3$ and we will model the vascular systems under consideration as trees or graphs. Let us start with the definition of the set of leaf nodes $\mathcal{L}$ and the (set of a) root node $\mathcal{R}$, which we assume to be fixed, prescribed by the user and such that $\mathcal{L} \cap \mathcal{R} = \emptyset$:

$$\mathcal{L} := \{Y_i \mid i = 1, \ldots, M\} \subset \mathcal{V}, \quad \mathcal{R} := \{X\} \subset \mathcal{V}.$$ 

Although the set $\mathcal{R}$ contains only one element, i.e. the root node, we introduce it for the consistency of the presentation.

Trees: For a set of vertices or nodes $\mathcal{N} := \{x_i \in \mathcal{V} \mid i = 1, \ldots, n\}$, which must contain the root-node $\mathcal{R}$ and the leaf-nodes $\mathcal{L}$, we define directed edges

$$e_j = (x_{j_1}, x_{j_2}) \quad \text{for} \quad x_{j_1}, x_{j_2} \in \mathcal{N}.$$ 

We always assume the edges to be directed toward the leaves. In particular, for an edge $e = (x_{j_1}, x_{j_2})$ the direction goes from node $x_{j_1}$ to node $x_{j_2}$. We collect the edges in a set

$$\mathcal{E} = \mathcal{E}(\mathcal{N}) := \{e_j \mid j = 1, \ldots, N\} \subset \mathcal{N} \times \mathcal{N}.$$
Now, we associate the following quantities with an edge \(e_j = (x_{j1}, x_{j2})\): The length of an edge will be the Euclidean distance between its vertices \(l_j := |x_{j1} - x_{j2}|\), and for the modeled vascular systems we denote the radius and the scalar flow of an edge \(e_j\) with \(r_j\) and \(f_j\), respectively. We denote the set of neighbor edges of a node \(x \in \mathcal{N}\) as \(\mathcal{B}_E(x) := \{e \in \mathcal{E} \mid e = (y, x) \text{ or } e = (x, y)\}\), i.e. the set of edges that share the node \(x\). Similarly the set of neighbor nodes of a node \(x \in \mathcal{N}\) is the set of nodes, which are connected with \(x\), i.e. \(\mathcal{B}_N(x) := \{y \in \mathcal{V} \mid (y, x) \text{ or } (x, y) \in \mathcal{E}\}\). The set of nodes and the set of edges define the directed graph \(G := (\mathcal{N}, \mathcal{E})\). We assume this graph to be a tree, i.e. connected and without any cycles. In the following, we will use the term branching point for the nodes \(x \in \mathcal{N} \setminus (\mathcal{L} \cup \mathcal{R})\) as opposed to the terminal nodes \(x \in \mathcal{L} \cup \mathcal{R}\).

Horton-Strahler order: For introducing a hierarchy on the modeled trees, we use an iterative, conditional, ascending ordering scheme developed originally by R.E. Horton and A.N. Strahler for the characterization of river systems [19], [20]. Applying this scheme to our situation, we initialize the leaves of the tree with a Horton-Strahler (HS) order \(i = 1\). For any other branch that splits into \(k\) branches, with HS orders \(i_1, \ldots, i_k\), its order is calculated as

\[
i = \begin{cases} 
\max_j \{i_j\} + 1 & \text{if there is } m \neq n \text{ such that } \max_j \{i_j\} = i_m = i_n, \\
\max_j \{i_j\} & \text{otherwise.}
\end{cases}
\]

A. Assumptions

Our model relies on several simplifications and assumptions. First, we assume that the flow of the fluid through the vessel system is laminar. Second, we assume blood as a perfect fluid with constant viscosity, which does not depend on the diameter of the vessel it is flowing through. The latter is in contradiction to the Fähræus-Lindqvist law, stating that blood has a lower effective viscosity when it flows through small capillaries than when it flows through larger vessels [21]. However, for blood vessels with a diameter larger than 0.3 millimeters the viscosity of blood can basically be regarded as constant, such that the assumption of constant viscosity is justified for our work, which is targeted toward the intermediate and large vessels. Additionally, we make the assumption that there are no dynamical pressure drops in the vascular system due to the angle of bifurcation or vessel curvature.

Together, these assumptions mean that we can use Hagen-Poiseuille’s law to model flow and pressure drops in the vessel system. Finally, we assume that our flow is steady as opposed to pulsatile. This last assumption is motivated from literature, which suggests that there is not much difference between geometrically optimizing a vascular system with pulsatile flow and one with steady flow [22]. This will be detailed in the following section.

B. Physiological Considerations

We assume that in the course of vascular growth, certain characteristics of the vascular system are optimized [12], [13], [18]. Following the arguments of Murray, both the biochemical energy required to create and maintain blood, and the physical energy required to pump blood are costly. Consequently, it is natural to adopt the minimum work model developed by Hess and Murray wherein both the intravascular blood volume, and the energy lost to friction while pumping blood are minimized [11], [23]. The Hess-Murray law, which directly arises from this model, is described by

\[
f_j^l = \kappa (r_j^l)^3, \tag{1}
\]

relating the flow through a vessel to its radius. \(\kappa\) is a constant called flow coefficient (unit: \(s^{-1}\)) and specific for each organ and vascular system. It will be different, e.g., for arterial and venous systems, according to the different pressure drops available.

For flexibility in our model, we will generalize (1) as

\[
f_j^l = \kappa (r_j^l)\alpha, \tag{2}
\]

where \(\alpha\) is a parameter of the model called flow exponent. For \(\alpha = 3.0\) we obtain the condition under which shear stress on the vessel walls is uniform over the entire system [18]. A value
of α = 2.55 has been proposed to minimize wave reflection in pulsatile blood flow [22].

Additionally, when the flow is not assumed to be laminar, the optimal α is in the range between 2.33 and 3.0 [24], [25]. The value of α = 2.33 represents fully turbulent and α = 3.0 fully laminar flow; values between the two extremes represent turbulence of varying degree. Since this paper is mainly for theoretical purposes and results have already been published for α = 3.0 [10], [12], [18], we will adhere to this value.

Since flow must be conserved at a branching, for any edge $e_p$ we have

$$f_p = \sum_{e_j \in A_p} f_j.$$  \hfill (3)

In combination with (2), this creates a so-called allometric radius power law for branchings:

$$(r_p)^\alpha = \sum_{e_j \in A_p} (r_j)^\alpha$$  \hfill (4)

Note that this radius law is of a more general nature than the Hess-Murray law, since it allows for some freedom in the flow-radius relationship across a vascular system. In our model, we will adhere to the conservation of flow (3) and to the generalized Hess-Murray law (2); therefore, we will implicitly fulfill the allometric radius law (4).

C. Definition of costs of a tree

On a set of edges $\mathcal{E}$ we define a cost function $C$ as

$$C : \mathcal{E} \to \mathbb{R}_{0}^{+}, \quad C(e) = r_i^\beta f_j^\gamma 1_e,$$

for $\beta, \gamma \in \mathbb{R}$ chosen appropriately. The choice $\beta = 2, \gamma = 0$ leads to local costs, which are proportional to the intravascular volume of the segment modeled by the edge $e$. Other choices of $\beta$ and $\gamma$ were discussed in several papers as extensions to the Hess-Murray law, e.g. presenting cost models based on minimization of surface ($\beta = 1, \gamma = 0$), surface forces ($\beta = -2, \gamma = 1$), or work ($\beta = -4, \gamma = 2$) [25]–[27].

When the flow exponent $\alpha$ is held constant (cf. Sec. III-B), $\beta$ and $\gamma$ can be combined to a single exponent via (2). For the remainder of this paper, we will use $\beta = 2$ and $\gamma = 0$.

Now we can define the global costs (or energy) of a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ as the sum of the local costs of all its edges:

$$C_{\text{global}}(\mathcal{G}) = C_{\text{global}}(\mathcal{E}) = \sum_{e \in \mathcal{E}} C(e).$$

D. The global optimization problem

To optimize a given tree and thus the vascular system being modeled by the tree (cf. Introduction), we consider the following global optimization problem (cf. Fig. 3 left):

For a given root node $\mathcal{R}$, a given set of leaf nodes $\mathcal{L}$, given flow exponent $\alpha$ and coefficient $\kappa$, and prescribed flows at the leaf nodes, find the tree $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ from the set $\{\mathcal{G}\}$ of trees with leaves $\mathcal{L}$ and root $\mathcal{R}$ that fulfill (2) such that

$$C_{\text{global}}(\mathcal{G}) = \min\{C_{\text{global}}(\mathcal{G})\}$$

In our notation, this task is equivalent to the following one in which we are looking for a set of branching points and corresponding connections in form of edges such that the root $\mathcal{R}$ is connected with the leaves $\mathcal{L}$ (cf. Fig. 3 left):

For a given root node $\mathcal{R}$, a given set of leaf nodes $\mathcal{L}$, given flow exponent $\alpha$ and coefficient $\kappa$, and prescribed flows at the leaf nodes, find a number $n \in \mathbb{N}$, a set of nodes (branching points) $\mathcal{N} := \{x_1, \ldots, x_n\} \subset \mathcal{V}$, and connections $\mathcal{E}$ between the nodes $\mathcal{N}$ from the set $\{(\mathcal{N}, \mathcal{E})\}$ of trees with leaves $\mathcal{L}$ and root $\mathcal{R}$ that fulfill (2) such that

$$C_{\text{global}}(\mathcal{E}) = \min\{C_{\text{global}}(\mathcal{E})\}$$

The global cost function $C_{\text{global}}$, which is minimized in this problem, is obviously non-convex. Due to this and the quite complex shape of the cost function $C_{\text{global}}$, the proof of the unique existence of a solution to this problem is still an open question. However, we know, since the cost functional is continuous and we can restrict the space of possible solutions to a bounded set in $\mathcal{V}$, that at least one minimum exists.

In the following sections we describe a strategy, which optimizes a given tree with respect to the global cost function $C_{\text{global}}$. Because we cannot expect to find the global minimum of $C_{\text{global}}$, we initially attempt to find a local minimum.

IV. LOCAL OPTIMIZATION

Assume that we are given an initial tree $\mathcal{G}^1 = (\mathcal{E}^1, \mathcal{N}^1)$ (cf. Sec. V-A) which is non-optimal in the sense of the cost function. We start the optimization with a local optimization strategy. Let us introduce the following local cost function

$$C_{\text{local}} : \mathcal{N} \to \mathbb{R}_{0}^{+}$$

with

$$C_{\text{local}}(x) := \sum_{e_i \in B_E(x)} C(e_i) = \sum_{e_i \in B_E(x)} r_i^\beta |y_i - x|,$$

which sums up the costs $C(e_i)$ of all neighbor edges $B_E(x)$. This local cost function can be seen as the restriction of the global costs to the local graph given by $B_E(x)$, i.e. $C_{\text{local}}(x) = C_{\text{global}}(B_E(x))$.

We emphasize that the local cost minimization problem is much easier to solve than the global problem. Indeed, the local

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig3}
\caption{Sketch for boundary conditions: terminal node positions and flow distribution (left). Modeling result for the shape of a flat square (right). Note that despite the symmetric shape, the optimal configuration evolves to be asymmetric.}
\end{figure}
cost function \( C_{\text{local}} \) is convex, since for fixed nodes \( y \in B_N(x) \) it is the sum of the convex functions \( C(e) \). This means that for fixed neighbor nodes we have a unique position of \( x \) that minimizes \( C_{\text{local}}(x) \).

### A. Local relaxation

To minimize \( C_{\text{local}}(x) \) with respect to the coordinates of \( x \in \mathcal{V} \), we compute the derivative of \( C_{\text{local}} \) w.r.t. its coordinates:

\[
\nabla C_{\text{local}}(x) = - \sum_{e_i \in B_E(x)} \omega_i \frac{y_i - x}{|y_i - x|},
\]

(5)

Note that the local cost derivative is independent of the edge’s length. Attributed to an edge for a given radius, it can thus be interpreted as a constant force with which the edge is pulling the node toward shortening the edge.

A critical point of the local cost function would be reached if \( \nabla C_{\text{local}}(x) = 0 \). In view of the local optimizations, which will be performed on neighboring nodes and their local costs, we employ an iterated steepest descent algorithm, i.e., we adjust the position of \( x \) in direction of the steepest descent with a step size \( \omega_x \):

\[
x_{\text{updated}} = x - \omega_x \nabla C_{\text{local}}(x).
\]

(6)

The step size is chosen separately for each node \( x \in \mathcal{N} \), because it is difficult to find an \( \omega \) which is optimal for every node. Moreover \( \omega_x \) can be increased or decreased for each node, as it will be detailed in the next section when discussing a global optimization strategy. We call one step of the gradient descent, performed on all non-terminal nodes, a relaxation of the tree. Note that the order by which the nodes are relaxed has a potential influence on the final configuration. We propose to store the node positions in a double buffer with a global update after each relaxation step.

During the iterated relaxation of all local configurations, the need eventually arises for topological changes. There are two ways in which the topology of the vascular system is altered, merging and splitting. We will explain these operations in the following sections.

### B. Merge

The process of merging involves joining two neighboring nodes and removing the edge that was between them. Merging should be done in a situation where successive relaxation would eventually place a node \( x_0 \in \mathcal{N} \) at the same location as one of its neighboring nodes \( x_1 \in B_N \). Before merging, the tree should be relaxed. Otherwise, a node might merge with a neighbor that would be in a drastically different position if the tree were allowed to relax.

The condition for merging could be tested explicitly according to the relative strength of the neighbor edge forces. An explicitly check, however, is computationally expensive and unnecessary since successive relaxation will bring the nodes together anyway. For this reason we prefer to simply test for an edge \( e^* \in B_E(x_0) \) whether the criterion

\[
\frac{l_{e^*}}{l_e} \leq \delta \quad \text{for all } e \in B_E(x_0) \setminus \{e^*\}
\]

(7)

is fulfilled for a given threshold \( 0 < \delta \ll 1 \). If this is the case, we remove the edge, assuming that the optimal position of \( x_0 \) coincides with \( x_1 \), i.e., we merge the nodes \( x_0 \) and \( x_1 \). Finally, the connectivity of all involved edges is updated, i.e., they are attributed to the newly created node (cf. Fig. 4).

---

**Fig. 4.** Result of merge operation at the example of a suboptimal configuration of two neighboring bifurcations. The two branchings are merged to form a trifurcation: pre-merge (left) and post-merge configuration (right).

When merging two nodes, special consideration must be made for merging with a terminal node (i.e., a leaf or root node). It is actually impossible to merge with a terminal node, since terminal nodes define the points, which our tree must supply; a resulting node would thus not be allowed to be moved by local relaxation. Instead, if the optimal position of a node is identical to a neighboring terminal node, we decrease the length of the connecting edge to zero. However, an edge of length zero still has a pull force (cf. next section). The pull force of this edge is always in the opposite direction as the pull force of the remaining edges in the branching. In this way, the zero length edge serves to anchor the branching node at the same position as the terminal node.

### C. Split

The splitting operation is, in a sense, the reverse of merging. Candidates for splitting are branchings, which involve four or more edges, i.e., trifurcations or higher-order nodes. Splitting is the backbone of the entire optimization process and is much more involved than its counterpart. The basic idea is to split a single branching into two smaller branchings (cf. Fig. 5) so that the most is gained in terms of cost. To this end, we introduce a measure called rupture strength \( \sigma \).

**Fig. 5.** Result of split operation at the example of a trifurcation. The suboptimal branching is split into two bifurcations: pre-split (left) and post-split configuration (right). The pre-split corresponds to the post-merge configuration of Fig. 4 right after additional relaxation.

Let \( S \subset B_E(x) \) be the set of edges, which are to be tested as candidates for the formation of a new node which will
be connected to the branching point \( x \in \mathcal{N} \). A new edge \( v = (x, x^*) \), which would be inserted connecting the original node \( x \) to the new node \( x^* \), receives the radius \( r_v \) according to the allometric radius law (4):

\[
 r_v = \left( \sum_{e \in S} r_e^\alpha \right)^{1/\alpha}.
\]

Where \( S' = S \) if \( S \) only contains child edges of \( x \), and \( S' = B_E \setminus S \) otherwise. Note that zero length edges (cf. Sec. IV-B) are never considered as candidates for splitting off; it can easily be shown that this is consistent with the condition of maximum rupture strength (cf. below).

Now, we define the pull force \( F_S \) of the edges in \( S \) as the sum of the forces of these edges (cf. Sec. IV-A)

\[
 F_S = - \sum_{e \subset (x, y) \in S} r_e^\beta \frac{y_i - x}{|y_i - x|}.
\]

This equation (8) is different from the local cost derivative (5) only in the set of edges for which the sum is computed. Since the node should be relaxed before splitting (\( \nabla C_{local}(x) = 0 \)), and since merging is performed before splitting (cf. Sec. IV-B), \( F_S \) would have the same length (but opposite direction) if \( S \) in (8) were replaced by \( B_E \setminus S \), except for nodes with a zero length edge, where the sum of forces is not zero.

The idea is to compare this pull force, which represents the tested configuration, to the force that would be present if another bifurcation node existed. We do this by introducing the rupture strength:

\[
 \sigma(S) = |F_S| - \left| r_v^\beta \frac{F_S}{|F_S|} \right| = |F_S| - r_v^\beta \quad \text{if } F_S \neq 0.
\]

\( \sigma(S) \) is an indicator for the gained cost per unit displacement of the new node \( x^* \) in the direction of \( F_S \). Accordingly, the set \( S \) is taken as a candidate for splitting if \( \sigma(S) > 0 \). Theoretically, we can try every combination of blood vessels to split off a node and select the choice with the largest rupture strength; however, this technique is too computationally expensive and in practice we use an iterative technique for an approximation of this search:

We start by searching for a split-off bifurcation, where the new node \( x^* \) receives exactly two of the original node’s edges while keeping all remaining edges attached to \( x \). We simply search all pairs of edges \( S_2 \subset B_E(x) \) for the split with the highest rupture strength \( \sigma(S) > 0 \), which creates a bifurcation. Even for very large branchings this is computationally feasible.

We further make the assumption that a trifurcation split \( S_3 \subset B_E(x) \) with the highest rupture strength will contain the two edges \( S_2 \subset S_3 \) of the bifurcation split with the highest rupture strength, computed before.

We can recursively extend this approach by assuming that the best split, which splits off \( N \) edges while keeping the remaining edges attached to the original node, will contain the \( N - 1 \) edges, which created the best ‘\( N - 1 \) split’. Following this assumption, we can quickly find the split with the absolute highest possible rupture strength. Although we do not give proof that this assumption holds, in practice this technique has been able to find the absolute best split for a branching in all cases where verification was possible.

After the optimal splitting set \( S \) was found, we initialize a new node \( x^* = x \) with \( \omega_{x^*} = \omega_x \) and update the connectivity by attaching all edges in the set \( S \) to the node \( x^* \) rather than to \( x \) and by creating the edge \( v = (x, x^*) \). We continue the optimization process by performing a relaxation step for the new node \( x^* \).

![Fig. 6. Result for shape of a real liver, similar to a portal (top) and a hepatic venous tree (bottom). Identical optimization parameters and organ shape were used for both images. The only differences are the initial root position (manually chosen) and leaf positions (top: square grid, bottom: random), and a slightly different viewpoint.](image)

V. TOWARD GLOBAL OPTIMIZATION

A. Initialization

So far, we have discussed local optimization procedures on a given tree. If, however, the prescribed volume of an organ is to be supplied with a new vascular tree, we have to consider an initialization procedure.

Initially, we generate the set of leaf nodes \( \mathcal{L} \) by either filling a grid or by picking uniformly random points within the volume. However, there is no reason another technique cannot be used; except for branchings, which have edges in the finest hierarchical level, we have found that the choice of the grid does not significantly influence the resulting branching structure (cf. Fig. 6). Additionally, we pick a single root node, which we will consider to be the anatomical hilum, i.e. the
entrance point of blood into the system. To initialize, we create edges that connect the root node $x \in \mathcal{R}$ directly to each of the leaf nodes:

$$\mathcal{E}^1 = \{(x, y) \mid x \in \mathcal{R}, y \in \mathcal{L}\}.$$  

In this manner, we have created the topology of a very crude vascular tree; all edges $\mathcal{E}^1$ are in the finest hierarchical level (cf. Fig. 7 left). The flows at each of the leaves is initialized to a constant value. In the simplest case, this value is identical for all leaves; however, any flow distribution, e.g. derived from a-priori knowledge or physiological data, can be chosen here. We set $\omega_x = 1$ for all nodes. Moreover, we choose $\kappa$ such that the radius $r_{\text{max}}$ of the root edge, which will evolve at a later stage, meets a fixed target value, e.g. $r_{\text{max}} = 5$ mm, for given $\kappa$ and total flow according to (2). With given $\alpha$, $\kappa$, and flow distribution, we use (2) to initialize all edge radii.

### B. Iterated local optimization

A first approach toward global optimization would be the iteration of the local relaxation described in the previous section. Many steps could be done on each node, before moving on to the local relaxation of the next node, but the change in position of neighboring nodes would render most of these adjustments irrelevant. Therefore we combine the local relaxation with a traversal of the tree and moreover consider a variable step size of $\omega_x$, adapted to the local steepness of the energy shape (cf. Algorithm 1).

The algorithm is similar to Armijo’s line search for steepest gradient descent and is known to converge to a local minimum [28]. It moves a node $x$ by a step of size $\omega_x$ in the direction of steepest local descent $\nabla C_{\text{local}}(x)$, as described by (6). If the cost does not decrease by such a step, the step size was too large, thus $\omega_x$ is halved and stored for the next iteration. In this case, no update is made. Otherwise, the algorithm evaluates the costs that would result from going a greater step of size $2\omega_x$, which is an extension of Armijo’s algorithm not influencing the convergence. If the greater step results in a further decrease of costs, the double value of $\omega_x$ is stored for the next iteration.

#### Algorithm 1 Adaptive local relaxation step

1: for all $x \in \mathcal{N} \setminus (\mathcal{R} \cup \mathcal{L})$ do
2: $\hat{x}_a = x - \omega_x \nabla C_{\text{local}}(x)$
3: if $C_{\text{local}}(\hat{x}_a) < C_{\text{local}}(x)$ then
4: $\hat{x}_b = x - 2\omega_x \nabla C_{\text{local}}(x)$
5: if $C_{\text{local}}(\hat{x}_b) < C_{\text{local}}(\hat{x}_a)$ then
6: $\omega_x = 2\omega_x$
7: $x = \hat{x}_b$
8: else
9: $x = \hat{x}_a$
10: end if
11: else
12: $\omega_x = \frac{1}{2}\omega_x$
13: end if
14: end for

Since for the cost computation in lines 3 and 5, the internode distances have already been computed, the test for node merging as described in (7) is computationally efficient. The third local operation, splitting, is always considered for trifurcations and higher-order nodes; only in very rare cases are such nodes optimal. The question whether to test a node for splitting or not is thus also very efficient. The actual computation of the optimal splitting configuration $S$, however, is the most computationally expensive part of our algorithm.

#### C. Convergence

The most important question we can ask of our algorithm is whether it will always provide an answer. Restated, will the algorithm always converge? In answer, the algorithm always finds a local minimum of cost. To prove this, note that each of the three operations of relaxation, merging, and splitting only performs an action if it will decrease cost. Furthermore, if the vascular tree is in equilibrium, this means that the tree can not be advantageously modified by local positional changes nor topological changes. Hence, since no other modifications are applied to the branchings in a vascular tree, it must be in a state of local minimum. Restated, because cost must always
be greater than zero, and cost in the course of the algorithm is always decreasing, the algorithm must converge to some local cost minimum.

D. A multi-level approach

The optimization process as discussed so far is not satisfactory, since the topology, which is induced by or created soon after the initialization, is subtly preserved throughout the iteration process. To alleviate this problem, we introduce a concept of hierarchical-iterations. In essence, we create more freedom for the optimization process to find a better local minimum, ideally, the global minimum. To understand where the problem lies, it is important to see that branchings rarely merge together after they have been split apart. The optimization process leads inevitably toward a tree which consists only of bifurcations, and occasionally, in rare cases, trifurcations.

The predominance of bifurcations in optimized trees cannot be disputed; therefore, it is important to understand how this affects our procedure. A bifurcation can be optimal locally; but, when seen from the larger context of the whole tree, it becomes apparent that a leaf node supplied by the bifurcation could more efficiently be supplied by another part of the tree. To remedy this situation we must step back to a coarser, higher hierarchical level. We call this process of pulling back and looking at the basic structure of the tree pruning. To create a pruned tree at level $G^{l^*}$ from $G^1$, we define the following set of nodes and edges:

$$N^{l^*} = \{x \in N^1 \mid x \text{ has HS order } \geq l^*\},$$
$$E^{l^*} = \{e = (x_j, x_{j+1}) \in E^1 \mid x_j, x_{j+1} \in N^{l^*}\}.$$

This step creates a lot of freedom, as we are no longer dealing with all the minor bifurcations, which have a tendency to anchor the tree to a local minimum.

The next challenge is to use this freedom to reach a better local minimum. The most straightforward and effective way we encountered is simply to reconnect all the leaf nodes, which were chopped off in the pruning stage, to the nearest node in the pruned tree. The new tree $G^1 = (N^1, E^1)$ is created from the pruned tree $G^{l^*}$ by the following definition:

$$N^1 = N^{l^*} \cup \mathcal{L},$$
$$E^1 = E^{l^*} \cup \{e_{x^*} = (x^*, Y) \mid Y \in \mathcal{L} \text{ and } l_{e_{x^*}} \leq |x - Y| \ \forall x \in N^{l^*}\},$$

where $e_{x^*}$ is the shortest possible edge connecting a leaf $Y$ with a node $x^*$ of $N^{l^*}$. In this way, a new tree is created with several multifurcations, i.e. larger branchings than the original tree contained before pruning (cf. Fig. 8 right). The nearest neighbor reconnection is currently performed by an exhaustive search, which can also be replaced by some approximate search (e.g. Kd-tree). Furthermore, the search could be rendered more accurately by taking into account not only the nodes but also the edges of the pruned tree; however, this would be more computationally expensive and might have the effect of hampering the vital splitting operations, which are the reason for the effectiveness of the multi-level approach. In any case, this has not been investigated so far.

Fig. 8. A model, which has been optimized using local optimization only (left, cf. Fig. 11 center). Basic structure of this model when all edges of HS order 3 or less are removed (center). Initial state of the new model, which takes advantage of the old models structure (right).

Fig. 9. Pipeline for iterated global optimization (left) and scheme for hierarchical-iteration cycles with progressively finer levels (right). Note that in time, which goes from bottom to top, we do not smooth between pruning and reconnect.
this tree is optimized, it generally finds a lower cost minimum than the tree from which its structure was taken. In a sense, the process of hierarchical-iterations, i.e. successively pruning, reconnecting, and relaxing, can be thought of as restarting the algorithm from a higher hierarchical level. Indeed, in the extreme case of pruning to the very coarsest level, this technique is equivalent to reverting to the initialization stage, by connecting all leaves to the root node. To balance the effects of using the optimality that the tree already contains and of providing freedom to solve global optimization problems, we iterate the process on different hierarchical levels. In each hierarchical-iteration we use the information from progressively finer hierarchical levels.

For example, a globally optimal tree would be created by first optimizing the tree as originally described using relaxation, merging, and splitting. If the resulting tree has \( l_{\text{max}} \) hierarchical levels, by pruning away \( l_{\text{max}} - 1 \) levels we are left with the coarsest skeletal structure which is still beneficial to creating a more optimal tree. We then reconnect the leaf nodes to this skeletal structure, which only has one hierarchical level. By optimizing the reconnected tree, we arrive at a new local minimum, which, in our experience, is never more costly than the original tree.

![Fig. 10. The evolution of the global costs versus the number of relaxations is shown for the hierarchical optimization algorithm. The numbers in boxes are the cost peaks explain the levels to which the tree has been pruned. Moreover, it is obvious from the plot that the pruning and reconnection operations produce a big advantage for the global costs.](image)

By repeating this process on the reconnected tree (cf. Fig. 9), we continue to find lower local minima (Fig. 10). Specifically, we would now prune the newly reconnected tree by removing \( l_{\text{max}} - 2 \) hierarchical levels, again reconnecting, and again optimizing. Then \( l_{\text{max}} - 3 \) levels, and so on until we have no more hierarchical levels left to prune. The final tree produced in this manner we consider our globally optimized tree (cf. Fig. 8 and 11). The stopping criterion for the optimization procedure can be either an empirically chosen predefined number of iterations or a lower threshold on the relative cost decrease after a certain number of full iteration cycles.

We were able to completely optimize trees containing 1,000 nodes in approximately 3 minutes of computation time on a Pentium III, 800 MHz, with 256 MB of RAM. Trees with 12,000 nodes were still manageable on that machine. We used the same number of iterations for all trees shown in this paper \((t = 400 \) iterations per smoothing cycle\). However, more iterations are generally required for larger trees.

The splitting algorithm has a complexity of \( O(N^3) \), with \( N \) being the number of edges in a branching. Unfortunately, this time is significant with respect to relaxation iterations. Still, the relaxation iterations require the most amount of time.

VI. DISCUSSION AND FUTURE WORK

The modeling of a system is important both because it furthers general understanding of the system and because the model itself can be directly used in applied techniques. With this in mind, we have undertaken the modeling of vascular systems.

To our knowledge, our approach is the first to be able to globally optimize a complex vascular system based on geometric or physiological cost principles. Note that in the presented optimization approach, we do not introduce rules on branching angles, such as for example in [15]. Rather, these emerge to be optimal from the iterative local optimization strategy. The multi-level global optimization has become necessary due to the memory effect of early large node splitting (cf. Fig. 11 center). Yet, we understand the presented method as a starting point for a more elaborate global optimization methodology.

In the future, we plan to allow at least two complementary trees to optimize contemporaneously. The addition of another root node, as the drainage point of the system, significantly complicates matters since the consistency of pressure drops in the system must be considered. Still, one can show that the two-tree system is well defined and computationally feasible.

Furthermore, we conjecture that the positional and topological optimization with respect to pressure-flow effectiveness cause opposing vascular trees to naturally separate to a large extent, a phenomenon, which we call interdigitation. To understand this, note that any real vascular system is asymmetric to a certain extent. The path distance and the hydrodynamic pressure drops from the root to the vascular endpoints exhibit a wide distribution [18]. Given this, shorter perfusion paths can afford longer drainage paths and vice versa, in order to meet the pressure conditions at the entry and exit points of the system.

We can thus argue that an optimal vascular configuration, in the sense described in this paper, must show interdigitation. However, the current implementation of hierarchical-iterations is not able to globally optimize the two-tree system in this sense. We hope that consistent local optimization of the tree on all hierarchy levels—not only on the finest—, will render this possible.

Note that a perfectly symmetrical object, such as the flat square in Fig. 3 right, is optimally supplied by an asymmetric vascular system. Asymmetry of vascular systems, which emerges automatically within the described optimization scheme, was pronounced and a comprehensive scheme for its quantitative characterization presented in the work of D. L. Turcotte et al. [29].

The most important difference between our method and CCO [17] is the ability to perform local and global topological
changes during optimization. We argue that these are the major ingredients toward a fully optimized vascular system. The challenge, however, has been to find a computationally feasible algorithm to do so.

Another vascular model for the human liver was developed by Kretowski et al. [6], taking into account two complementary vascular trees. Still, even if collisions are explicitly avoided within their model, the trees are not truly optimized together. It is quite similar to the work by Schreiner et al. [17] since it successively adds new leaves and locally optimizes the newly created branching.

Our generated trees are fractal in the sense that they are volume filling and scale invariant [30]. They are volume filling since the leaf positions are chosen such that a given volume is filled at a given resolution; and they are scale invariant since the optimization principles only take into account the relation of local geometric properties of the tree. The significance of fractal geometry as a biological design principle has been pronounced by E.R. Weibel [31]. One of his aims was to explain how, e.g., “the large internal gas exchange surfaces of the lung can be homogeneously and efficiently ventilated and perfused at low energy cost.”

The optimality of the bronchial tree has been questioned by B. Mauroy et al. [32], showing that physical optimization is critical for the physiological robustness of the bronchial system. Following their arguments, “the design of bronchial trees must be provided with a safety factor and the capacity for regulating airway calibre”, without risking very large variations in air supply. It is a fundamental question, whether a corresponding safety argument applies to blood vessels.

To discuss the implications of our work for surgery planning and risk analysis, it will be crucial to study the differences of real and modeled systems. Comparing the generated models for the shape of the human liver with vascular information obtained from corrosion casts [1], reveals a striking similarity in terms of gross vascular anatomy, branching patterns, and asymmetry (cf. Fig. 6). Preliminary quantitative results on scaling and branching properties, such as radius and length ratios, show a good correspondence. We plan to publish these results in the near future together with a comparison to in vivo data.

A potentially valuable approach to evaluate our model is to use in vivo data as an initialization for the global optimization procedure and to quantify the differences between initial and final configuration. With other words, we want to know whether an individual vascular system can be considered as the minimum of the described mathematical optimization problem. Of course, such comparison will be incomplete until we include both the perfusing and the draining vessels in the model. If successful, this will lead to an estimation of the predictive power of optimal vascular systems. We will then be able to model detailed vascular anatomy based on patient images acquired in vivo, e.g., by CT or MRI.

As one of the next steps, we will extend the presented algorithm to work with at least two complementary trees simultaneously. We will also consider a physiological simulation of flows and pressure drops in different flow regimes. By combining these two points we hope to develop an algorithm that displays interdigitation as a consequence of optimality criteria. Furthermore, we will conduct an in-depth comparison with CCO, which is the most closely related work, and also extend our considerations on anatomical optimality to other parenchymatous organs such as the lung and kidney.

ACKNOWLEDGMENTS

We wish to thank Jean H.D. Fasel for providing vascular corrosion casts of the human liver and corresponding CT images, which were an indispensable source of inspiration and anatomical information for our work. We acknowledge our colleagues Olaf Konrad-Verse for developing and providing vascular visualization tools, Richard Rascher-Friesenhausen for his aid on optimization strategies, Florian Link for his groundbreaking work on the image analysis and modeling platform MeVisLab, on which the presented work was built, as well as Anja Hennemuth and Holger Bourquain for providing fascinating insight into the resection, transplantation, and regeneration of the human liver. We thank Hiroko Kitaoka, Ewald R. Weibel, and Carl J.G. Evertsz for their
advice and profound discussions. Manfred Georg particularly acknowledges Karin Peitgen and Heinz-Otto Peitgen for their hospitality during his stay at MeVis.

REFERENCES