Growing Digital Leaves by Vascular Network Optimization

Kameron Decker Harris

May 1, 2009

1 Leaves in Nature

1.1 Introduction

The explanation of structures observed in nature, like the leaf, has long been the goal of mathematicians. Growing life forms from first principles is a computing task outside the realm of supercomputers and will remain so into the foreseeable future. However, it is still possible to model them by incorporating what we understand of the principles that guide their formation. This paper builds on the work of Qinglan Xia [1], who was the first to present leaf growth in this cost-functional formulation and whose results I try to replicate.

1.2 What makes a leaf a transport system?

Leaves serve a simple purpose in the life of the tree. These sheddable mouths gobble up sunlight like little green solar panels. Sunlight yields an energy benefit that scales proportional to the leaf area. Cells in the leaf require a supply of water and soluble nutrients. This is passed upward from the roots in tissue called xylem. In return, photosynthesis produces sugars which the leaf returns to the tree through the phloem. The different vascular tissues are arranged like a coaxial cable, with xylem inside of the phloem. Because of this, the two are said to form vascular bundles. The tree trunk itself is one of these bundles, albeit with a heart-wood core of the dead xylem and a protective outer bark. The vascular system of the leaf forms its veins. The specific job of each leaf's vascular network is essentially the distribution and collection of water and nutrients. Evolution selects for a leaf that efficiently accomplishes that task. The network formed by the veins is then a type of optimal distribution network.

As water and solutes are transported through xylem, the *root*, where the leaf connects to the branch, can be thought of as a source and each cell as a sink. Photosynthesizing cells are likewise sources of sugar to be collected at the petiole. Because the xylem and phloem are bundled together, transport to and from the leaf is managed by the same network. The distribution and collection problems are equivalent.

Finding the best way to transport goods between sources and sinks given some cost is usually called the Monge-Kantorovich problem. The modeler must first define how the cost function is calculated for a given path. Depending on the cost function, different paths may be favorable. Consider *Matteuccia struthiopteris*, the edible fern commonly known as "fiddlehead." Many harvesters (sources) around the state gather up fiddleheads they find. To get their haul to City Market (sink), each individual could transport it in their vehicle. However, depending on the price of gas and the value of the harvester's time, it may be more efficient for the harvester to sell to the Ostrich Transport Company which will consolidate the shipments midway and take them their in a large truck. Combining shipments forms a branching or *ramified structure*. Ramified structures are often preferable when there is an economy of scale from shipping more items together [Fig. 1].

Furthermore, each path through the transport system has an associated weight representative of the how much material travels through it. In the circulatory system, for example, a blood vessel with

a high weight would be an artery. Large volumes pass through the arteries, requiring a larger radius in order to lower the impedance. A blood vessel with low weight would be a capillary, since they carry smaller volumes. In the leaf, veins vary in thickness in a similar way.

Suppose leaves actually utilize these optimal distribution networks. How do they form? As the leaf develops, it tries to maximize surface area while minimizing transport cost. An interesting thing happens if this cost scales more quickly than leaf area: Eventually, the transport cost becomes so constraining that adding any more cells offers no benefit. The leaf stops growing, rather than become suboptimal. Remarkably, this kind of model ends up giving us the leaf's final shape, something other leaf models have not been able to produce.

All the ingredients for a mathematical model of leaf growth are in place. Each leaf begins its life as the same "bud." To account for variation among species, different parameters are introduced to the cost function. At each time step, new cells are grown wherever the energy benefit exceeds the transport cost. Then the network of veins is rearranged locally to become an optimal transport network. When this lowers the total cost on the network, it can allow for the addition of more cells. If not, the leaf has finished growing. The resultant vascular network is space-filling and spans all the cells of the leaf. The boundary formed by those cells depends on the parameters of the cost function, and it can assume the shapes of many leaves found in nature.

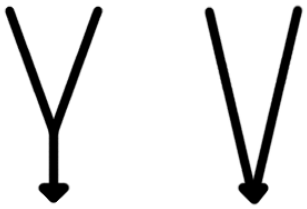

Figure 1: A y-shaped path may be more efficient than a v-shaped path.

2 Mathematical Leaves

2.1 Representation

Leaves are modeled in the computer as a weighted, directed graph $G = (V, E, w)$ on a finite square lattice in two dimensions (2D). This study disallows cycles to give a purely branching network or tree; real leaves in fact do have cycles although not usually for the highest order veins. Each grid point that contains a node represents a cell in the leaf. All cells have the same area equal to a^2 where a is the lattice spacing. Lattice points are denoted $\mathbf{r} = (x_i, y_j)$ where $i, j = 1, \ldots, N$. Subscripts are left out when unnecessary. The grid then contains N^2 points and a has a total area of $(aN)^2$. In order to not confuse empty sites and sites occupied by cells, symbols other r are used to denote vertices in the graph (e.g. $v_{ij} \in V$). Also, let the root be located at the origin $O = (0, 0)$.

The complete leaf also requires a representation of the edges. Edges can only exist between neighboring vertices. Note that because the graph is a directed tree, every vertex has a unique parent $p(v)$ and a unique path to the root $P_v = \{v, p(v), p(p(v)), \ldots, O\}$. The edge that whose tail is the vertex v is labeled e_v and can be written as the vector $\mathbf{e}_v = \mathbf{v} - \mathbf{p}(\mathbf{v})$. With this lattice, the parent and children of vertex v are restricted to the 8 sites in its range 1 Moore neighborhood, denoted $\mathcal{N}_v$. The set of children for vertex v is denoted C_v.

2.2 Weight Function

The weights on the edges of the graph must reflect the flow of nutrients into and out of the cell. Obviously, the highest weight edge will extend from the root since it is the source for the rest of the leaf. Each cell can be expected to consume roughly the same amount of water which should be roughly proportional to its area. The weight on the edge e_v which supplies v will then be a sum of all the weights "downstream" from or descended from v plus the consumption of the cell at v. We take the consumption of one cell to be equal to its area a^2. The balancing equation for the weights is then

$$w(e_v) = a^2 + \sum_{x \in C_v} w(e_x)\,. \tag{1}$$

This equation is reminiscent of Kirchoff's law for circuits, which maintains that the current flowing into a vertex must be the same as the current flowing out of it, except in this case each vertex adds a factor of a^2. Given a set of vertices and edges, the weight function is uniquely determined. Instead of recalculating the weights every time an edge is added, they are stored in an array. When an edge e_v is added, the weights only need to be updated along the path P_v.

2.3 Cost Functional

Define the total cost of the network F as the sum over all vertices of the graph of the cost f at each vertex

$$F[f] \doteq \sum_{v \in V \setminus O} f(v) \tag{2}$$

$$f(v) \doteq (w(e_v))^\alpha \times ||e_v|| \times m_\beta(v) \tag{3}$$

where $\alpha \in [0, 1)$ and $\beta > 0$ are parameters. The cost functional is just the sum of the cost function over all vertices of G. The length of edge e_v is denoted $||e_v|| = ||\mathbf{e}_v||_2$. The function m_β is defined recursively by

$$m_\beta(v) \doteq m_\beta(p(v)) \times h_\beta(e_v, e_p(v)) \tag{4}$$

where

$$h_\beta(e_v, e_p(v)) \doteq \begin{cases} \left|\hat{\mathbf{e}}_v \cdot \hat{\mathbf{e}}_{p(v)}\right|^{-\beta}, & \text{if } \hat{\mathbf{e}}_v \cdot \hat{\mathbf{e}}_{p(v)} > 0 \\ +\infty, & \text{otherwise} \end{cases} \tag{5}$$

where the hats denote Cartesian unit vectors of the associated edges. Let $m_\beta(O) = 1$. Since G contains no cycles, such "upstream" recursive definitions are well defined.

The fixed parameters α and β control different kinds of cost. Because $\alpha < 1$, cost scales sub-linearly with weight. This allows for tuning the efficiency of transporting two items together (an edge with twice as much weight) rather than separately (two edges). Very small α lead to pointy leaves with high weights, while larger α grow rounder leaves. The other parameter β tunes how much cost comes from changing the direction of flow. Because of the dot product, $m_\beta(v) \propto |\cos\theta|^{-\beta}$ where θ is the angle formed between incoming and outgoing edges of a vertex $u \in P_v$. As θ grows, the denominator shrinks, and cost increases. No branches will form at right angles or any larger angles, since the cost is then infinite.

3 Growing Algorithms

3.1 Initialization

At $t = 0$, every leaf is the same. All data structures are initialized to zero. A node is added at the root, and its parent is designated such that $\hat{\mathbf{e}}_O = (0, 1)$. This is sets the direction of growth.

3.2 Selection Principle

Where does the leaf grow new cells? They grow only where the change in total cost is less than the revenue they provide. This cost change is the incremental cost of having the new cell as part of the network. The revenue, on the other hand, is a function of the energy provided by photosynthesis. Like energy consumption, this is taken to be proportional to the cell's area. The constant of proportionality is $\epsilon > 0$, a parameter that tunes what can be thought of as the efficiency of photosynthesis. The *selection principle*, as Xia calls it, picks candidate new cells whose incremental cost is less than ϵa^2.

New cells can only grow adjacent to already existing cells. Boundary points are unoccupied points b with at least one occupied neighboring site $n \in \mathcal{N}_b$. Given multiple sites which can supply the new node, we want to choose the one with the least cost to be its parent. To find the best parent n^* for a given b requires the incremental cost $C(b, n)$ of adding a new node b with parent n. This C will consist of two parts. First, there is the cost of the new node itself, $f(b)$. Second, there is the effect of attaching a new weight a^2 to b. This weight will increment the weights of all $u \in P_b$. Xia calls this second contribution the *potential function*

$$\Pi(v, w_0) = \sum_{x \in P_v \setminus O} ((w(e_x) + w_0)^\alpha - (w(e_x))^\alpha) \times ||e_x|| \times m_\beta(x)\,. \tag{6}$$

which is the change to the total cost of adding an arbitrary weight w_0 to e_v. The change in total cost is

$$C(b, n) = f(b) + \Pi(n, a^2)\,. \tag{7}$$

With the incremental cost in hand, adding new cells is easy. At step $t = i + 1$, find the best parent n^* for candidate cell b. If $C(b, n^*) \leq \epsilon a^2$, add vertex b and edge $\mathbf{e}_b = (n^*, b)$ to G_i. Do this for all candidate points in the boundary of the leaf G_i. Denote the new graph as $\tilde{G}_{i+1}$.

3.3 Optimization

The new leaf $\tilde{G}_{i+1}$ is most likely not an optimal transport system, so that the optimal network $G_{i+1} \neq \tilde{G}_{i+1}$. However, given that G_i was optimal, we require only a local search to get from $\tilde{G}_{i+1}$ to G_{i+1}.

Xia fixes the vertex set V then modifies the edge set E by looking for a better parent for each vertex. To do this, another incremental cost function is needed for the change in cost of changing a given vertex's parent.

Let v be the vertex in question, $p = p(v)$, and u be the candidate parent. The incremental cost of changing v's parent can be broken into three parts: the upstream change in cost, the change to $f(v)$, and the downstream change in cost.

If $w_0 = w(e_v)$, changing v's parent from p to u will lower the weights along P_p and raise them along P_u by amount w_0, since weight accumulates upstream. If p and u share a common ancestor p^*, the weight of every vertex $v \in P_{p^*}$ is unchanged. Thus, the change in cost will be $\Theta(p, p^*, -w_0) + \Theta(u, p^*, w_0)$ where

$$\Theta(v, p^*, w_0) = \sum_{x \in P_v \setminus P_{p^*}} ((w(e_x) - w_0)^\alpha - w(e_x)^\alpha) \times ||e_x|| \times m_\beta(x) \tag{8}$$

is a function that measures the change to the total cost of adding weight w_0 to every edge between v and an ancestor p^*. Note that $\Theta(v, O, w_0) = \Pi(v, w_0)$.

Let $f'(v)$ be the cost at vertex v with new parent u, and let $e'_v = (u, v)$. The change in cost at v is $f'(v) = w(e_v)^\alpha\, ||e'_v||\, m_\beta(u)\, h_\beta\,(e'_v, e_u)$ minus the old cost $f(v)$.

The dowstream cost comes from the change in $m_\beta(x)$ for all x descended from v; call the set of descendents D_v. These new m_β's are the same as before except multiplied by the ratio $m'_\beta(v)/m_\beta(v)$. The downstream change to the cost can be written as

$$\left(\frac{m_\beta(u)\, h_\beta\,(e'_v, e_u)}{m_\beta(v)} - 1\right) \sum_{x \in D_v} f(x) \tag{9}$$

and we can also write

$$D_v = \left(\bigcup_{x \in C_v} D(x)\right) \cup C_v \tag{10}$$

which is a recursive definition for the set of descendants.

Given a network, the incremental cost of changing v's parent is the sum of these three parts

$$\begin{aligned}\Delta F \;=\; & \Theta(p, p^*, -w_0) + \Theta(u, p^*, w_0) \\ & + w(e_v)^\alpha \, ||e'_v|| \; m_\beta(u) \; h_\beta \left(e'_v, e_u\right) - f(v) \\ & + \left(\frac{m_\beta(u) \; h_\beta \left(e'_v, e_u\right)}{m_\beta(v)} - 1 \right) \sum_{x \in D_v} f(x) \; . \end{aligned} \quad (11)$$

4 Results

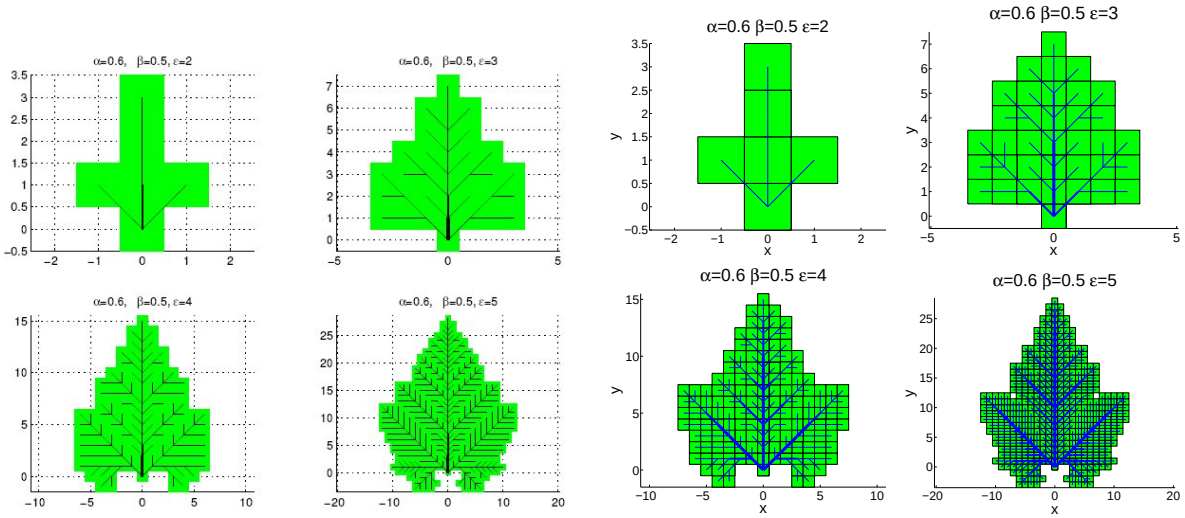

Figure 2: (a) renderings by Xia for $\alpha = 0.6$, $\beta = 0.5$, $\epsilon = 2, 3, 4, 5$. (b) what was found for those same parameters. All but the smallest leaves are different. This could be due to differences in the number of cells added and the details of the optimization. Xia's expression for the cost change when changing a parent is different from Eqn. (11)

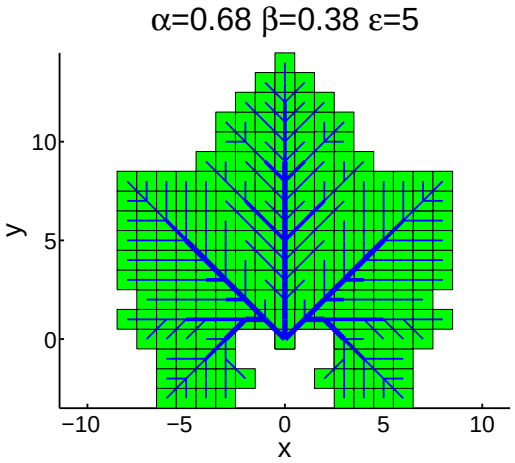

Figure 3: A parameter set which reveals a shape like a maple leaf.

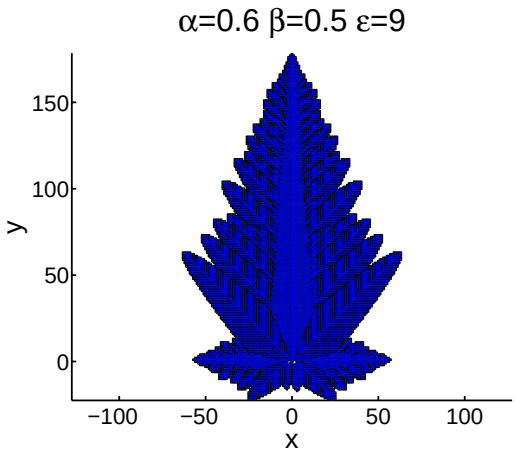

Figure 4: Turning up ϵ makes a larger leaf, although it does not look like any example from nature.

5 Conclusion

A mathematical description of leaf shape and vascular network structure is an enticing goal. The cost functional approach makes sense from an evolutionary perspective, and the study of optimal transport networks has evident applications to all sorts of problems. The difficulty with this approach comes from measuring the model parameters. To verify this model, there need to be data on real leaves. Also, not all leaves can be produced by such a model. Classification of the leaves that do fit the model versus those that do not could elucidate other mechanisms that play into the formation of leaves. These could then be put into a more realistic or completely different model for natural leaves not found in the parameter space of Xia's model.

The easiest way to modify the model is with a different cost function. An obvious way would be to raise the lengths to a power. Other lattices, among them hexagonal, might be tried. Also, less leafy yet optimal networks could be put together in three and higher dimensions.

Before any of these exciting ideas can be tried, we need to work out the details of what is already there. Xia's incremental cost function needs to be reconciled with Eqn. (11), otherwise we are calculating different costs. Also, it needs to be determined whether to add cells one at a time or as many as possible before optimization. When optimizing, we should know the difference between optimizing cells in a systematic way versus randomly. Xia also does not mention recent work with other physics-type models, so a thorough literature review is in order.

Dicrepancies aside, it is nice to see lifelike shapes emerge from a simple model.

References

[1] Qinglan Xia. The formation of a tree leaf. *ESAIM Control Optim. Calc. Var.*, 2:359–377, 2007.