

Modeling the geometry and dynamics of the Endoplasmic Reticulum network

Congping Lin, Laurent Lemarchand, Reinhardt Euler, Imogen Sparkes

Abstract—The endoplasmic reticulum (ER) is an intricate network that pervades the entire cortex of plant cells and its geometric shape undergoes drastic changes. This paper proposes a mathematical model to reconstruct geometric network dynamics by combining the node movements within the network and topological changes engendered by these nodes. The network topology in the model is determined by a modified optimization procedure from the work (Lemarchand, et. al. 2014) which minimizes the total length taking into account both degree and angle constraints, beyond the conditions of connectedness and planarity. A novel feature for solving our optimization problem is the use of 'lifted' angle constraints, which allows one to considerably reduce the solution runtimes. Using this optimization technique and a Langevin approach for the branching node movement, the simulated network dynamics represent the ER network dynamics observed under latrunculin B treated condition and recaptures features such as the appearance/disappearance of loops within the ER under the native condition. The proposed modeling approach allows quantitative comparison of networks between the model and experimental data based on topological changes induced by node dynamics. An increased temporal resolution of experimental data will allow a more detailed comparison of network dynamics using this modeling approach.

Index Terms—Endoplasmic Reticulum, Network Dynamics, 0-1 Programming, Cutting Planes

1 INTRODUCTION

The endoplasmic reticulum (ER) is the largest membrane bound organelle in most eukaryotic cells, and forms a highly complicated interconnected network of tubules and flattened sacs (known as cisternae) [1], [2]. It serves important biological functions including protein and phospholipid synthesis, quality control and export, and calcium storage [3]. In addition to its complicated shape, the ER network is highly dynamic, transiting between tubules and cisternae and dynamically changing their polygonal network [1]. Figure 1 (left panels) shows an example of ER network dynamics. The dynamic ER shape is suggested to be adaptable to the cell's requirements for ER function [4]. Mechanisms governing the ER network changes and the biological significance of these changes remain unclear.

Tools for investigating the ER morphology have been developed. Sparkes *et al.* [5] constructed an image analysis method for pulling out persistent or static elements of the ER network in tobacco leaf epidermal cells. Persistent elements such as static points, may have important roles in anchoring the network to the plasma membrane [5], [6], [7], [8]. Lin *et al.* [9] developed an image processing method to abstract ER tubule networks into geometric graphs between persistent and non-persistent nodes from a series of ER images. In addition, Boucekhima and co-workers [10] have analyzed the network of ER

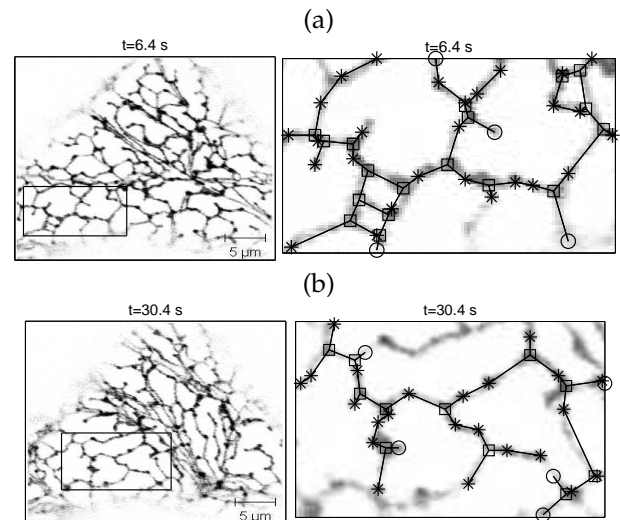


Fig. 1. Illustration of the ER networks at two different time points and the corresponding abstracted geometric graphs. The rectangle regions in left panels highlight a region with no ER cisternae and the right panels show their geometric graphs for the largest connected component. The abstracted graphs are obtained using the image processing method introduced in [9], where markers '*', '□' and 'o' represent persistent, non-persistent branching, non-persistent end nodes, respectively, and lines represent edges. The experimental ER images are taken from [5] (www.plantcell.org, Copyright American Society of Plant Biologists).

- Congping Lin is with Mathematics at the University of Exeter, UK
E-mail: c.lin@exeter.ac.uk
- Laurent Lemarchand, Reinhardt Euler are with Lab-STICC UMR 6285, UBO-Université Européenne de Bretagne, Brest, France
- Imogen Sparkes is with Biosciences at the University of Exeter, UK

cisternae.

In nature, networks (e.g., the skeletons of radiolarians or cell transport networks) are commonly structured

in an economic way, minimizing the length or surface; biophysical mechanisms (such as surface energies or interfacial tensions) can easily drive processes that result in minimal networks [11]. Indeed, the structure of tightly stacked ER sheets (cisternae) has been modeled as a minimum of elastic energy of sheet edges and surface [12]. In addition, the structure of the ER tubule networks in tobacco leaf epidermal cells is well modeled as a Euclidean Steiner network, a locally optimal network in which any local perturbation of non-terminal nodes would increase the total length of the network [9]. This is analogous to the notion of Euclidean Steiner trees [13] in which the non-terminals (called Steiner points) have degree 3 and each pair of edges at the Steiner points has an angle of degree 120° . Note, that the local optima might not give a unique network topology. In [14] we have proposed an optimization problem to construct an optimal network minimizing the total length with constraints on angles and degrees of nodes; such optimal networks are globally optimal and are close to the ER networks from experimental data.

To investigate ER network dynamics, the nodes were classified into *persistent* and *non-persistent* nodes [9]. The persistent nodes are those static ER points while the non-persistent nodes are either mobile ER junctions (referred to as *non-persistent branching nodes*) or ends of mobile tubules (referred to as *non-persistent end nodes*). In the following “non-persistent” is omitted in case of non-ambiguity. For the latrunculin B (treated) cells where the ER network was less dynamic, dynamics of branching nodes in a small network observed from experimental data were well modeled using a Langevin approach [9]; the network structure there was either fixed or changing among a small number of known topologies. The ER network in the native state is highly dynamic; dramatic remodeling (i.e., highly dynamic changes of network topology) and transitions between tubules and cisternae are observed. We have not attempted to model ER dynamics in the native state, but we prefer to extend the Langevin approach studied in [9] to model ER network dynamics by including the dynamics of end nodes and an optimization procedure for topology determination.

In this paper, we use live cell confocal microscopy data of ER networks in tobacco leaf epidermal cells taken from [5]. As the cortical ER in plant cells occupies a very thin, almost two-dimensional layer of cytoplasm beneath the plasma membrane, our study of the ER network is based on its 2D approximations. Figure 1 shows two instances of live cell images and corresponding abstracted geometric graphs from an ER recording of 50 frames with a time resolution of 1.6 s and a spatial resolution of $0.11\mu\text{m}$ per pixel; the geometric graph abstraction (including node positions of each type and edges) is obtained using the image processing method introduced in [9]. For the native state ER networks, due to the highly dynamic transition between tubule and cisternae, we select regions which do not contain cisternae for analysis.

2 MODEL

We follow the node classification from [9]; the persistent nodes V_p have degree no more than 3, the non-persistent branching nodes V_b have degree 3 and the non-persistent end nodes V_e have degree 1. In [14], we have formulated an optimization problem to construct the network topology with degree constraints (either degree 3 or degree no more than 3) and angle constraints for given degree-3 nodes. Note that in the ER network dynamics, the set of degree-3 nodes may change as they may include some persistent nodes. We also aim to include the dynamics of end nodes. Thus, a degree constraint for end nodes might be necessary. In the following we illustrate a modified optimization problem from [14] for constructing the network topology and the dynamics for each type of nodes.

2.1 Topology optimization

Given a set of nodes V and a classification $V = V_p \cup V_b \cup V_e$, the problem is to find an undirected, connected, spanning and plane subgraph, whose nodes in V_b have degree 3, those in V_e have degree 1 and those in V_p have degree no more than 3, which minimizes the sum of the Euclidean distances associated with the connecting edges, and which satisfies an angle constraint for all degree-3 nodes (including persistent nodes if they are branching in the solution). More precisely, by an angle constraint we mean that, for a branching node $u \in V$, and its neighbours v_1, v_2, v_3 , any two angles at the node u formed from edges $uv_i (i = 1, 2, 3)$ have a sum no less than θ where θ is a given angle around 180° . The idea comes from the concern of force balance acting on the branching node. For each branching node (ER junction), as modeled in [9], each of the three ER filaments is assumed to apply a membrane tension force on this tubule junction. Thus, it is unlikely that the tension forces are in the same direction of a half plane, i.e., the sum of two angles of the branching node is less than 180° . This optimization problem is formulated as follows:

$$\text{minimize } \sum_{x_{uv} \in E} w_{uv} x_{uv} \quad (1)$$

$$\text{subject to } 1 \leq \sum_{v \neq u} x_{uv} \leq 3, \forall u \in V_p. \quad (2)$$

$$\sum_{v \neq u} x_{uv} = 1, \forall u \in V_e. \quad (3)$$

$$\sum_{v \neq u} x_{uv} = 3, \forall u \in V_b. \quad (4)$$

$$\delta(W) \geq 1, \forall W \subset V \quad (5)$$

$$|W| \geq 2. \quad (5)$$

$$x_{uv} \in \{0, 1\}, \forall uv \in E. \quad (6)$$

$$\begin{aligned}
x_{uv} + x_{uw} + x_{uz} &\leq 2, \forall u \in V \setminus V_e \\
&v, w, z \in V \setminus \{u\} \\
\widehat{vuw} + \widehat{wuz} &\leq \theta. \quad (7)
\end{aligned}$$

$$\begin{aligned}
x_{uv} + x_{wz} &\leq 1, \forall u, v, w, z \in V \\
&u \neq v \neq w \neq z \\
&uv \text{ and } wz \text{ are} \\
&\text{crossing.} \quad (8)
\end{aligned}$$

where

- the objective function (1) represents the total Euclidean distance of the connecting subgraph;
- (2) and (3,4) represent the degree constraints on nodes;
- (5) ensures the connectivity of the resulting subgraph, where $\delta(W) = \sum_{i \in W, j \notin W} x_{ij}$;
- (7) ensures that nodes of degree 3 in the resulting subgraph have edges that are spanned over an angle of more than θ degree;
- (8) ensures that the resulting graph is plane;
- x_{uv} and x_{vu} represent the same variable, and $x_{uv} = 1$ iff edge uv is selected in the solution.

We remark that in general for a fixed θ , the constraint (7) may lead to infeasibility, in particular for degree-3 nodes near the border of the graph.

2.2 Network dynamics

We extend the Langevin model in [9] which concerns the balance between the tension and Stokes drag forces and the Brownian forces for the non-persistent branching nodes. In particular, we construct a mathematical model including end-node movement and dynamic attachment/detachment of persistent nodes as well as elements of topological changes determined by optimization as described in Section 2.1, assuming no changes of node types and the total node number of nodes in the evolution of time. The node dynamics of each type is as follows:

- 1) branching node movements. By balancing the tension and Stokes drag forces with the Brownian forces, the perturbation for the non-persistent branching node is modeled as in [9] by

$$\frac{dx_i}{dt} = -a \nabla_{x_i} f(x_1, x_2, \dots, x_n) + \sqrt{2\sigma} \xi(t) \quad (9)$$

where f is the total length of the network between nodes (μm), ∇ is the gradient, a is the drift coefficient ($\mu m/s$), σ is the diffusion coefficient ($\mu m^2/s$), $\xi(t)$ is the white noise with zero mean and autocorrelation $\langle \xi(t), \xi(t') \rangle = \delta(t - t')$, and $x_i \in \mathcal{R}^2$ is the position of nodes (μm).

- 2) persistent node dynamics. Some of the persistent nodes are observed only to be visible for part of the time during remodeling [9]. We model this as a random process and assume the invisible persistent nodes appear with rate p_a and visible persistent

nodes disappear with rate p_d . The invisible persistent nodes are isolated from other nodes in the remaining network. We remark here that the set V_p in the optimization procedure is for the set of visible persistent nodes.

- 3) end node movements. The non-persistent end nodes (of degree one) can shrink or grow in the network [5]. We assume that growing and shrinking velocities are equal and denote them as $v \in R$; we assume that the end node can switch between growing and shrinking stochastically and denote the transition rate from shrinking to growing as p_s and that from growing to shrinking as p_g . More explicitly, the shrinking and growing process is modeled as

$$\frac{dx_i}{dt} = -v \nabla_{x_i} f(x_1, x_2, \dots, x_n) + \sqrt{2\sigma_e} \xi(t) \quad (10)$$

$$\frac{dx_i}{dt} = v \nabla_{x_i} f(x_1, x_2, \dots, x_n) + \sqrt{2\sigma_e} \xi(t) \quad (11)$$

where the first term accounts the velocity while the second term $\sqrt{2\sigma_e} \xi(t)$ represents the perturbation from the noise just as for the branching nodes in 1.

The parameters $v = 0, p_d = 0, p_a = \infty, \sigma_e = 0$ reduce to the Langevin approach in [9] modeling the non-persistent branching nodes. Considering the computing time, we choose a time step of $dt = 0.1s$. We use the parameters $a = 0.2\mu m/s$ and $\sigma = 0.008\mu m^2/s$ from the estimation in [9]. For other parameters, we have not yet obtained any estimations from live cell imaging data. Intuitively, we assume $p_a \gg p_d$ and set $p_d = 0.01/s, p_a = 0.7/s, p_s = p_g = 0.1/s, \sigma_e = \sigma$ and $v = 0.1\mu m/s$ if the corresponding dynamics is applied. Estimation of these parameters will be possible when high resolution of experimental data are available. We are not attempting to explore the parameter space for the dynamic modeling, but we do compare network dynamics with/without persistent node dynamics and/or end-node movements.

In the simulation, we abstract locations and types of nodes in a self-contained ER network (where no ER junctions are connected to the nodes outside the network) using the image processing method in [9]. Considering this set of nodes and their types $V(0) = (V_p(0), V_b(0), V_e(0))$ as an initial state, we find its optimal graph $G(0)$ from the optimization problem formulated in Section 2.1. Then we apply the above node dynamics to each node according to its node type and find the corresponding optimal graph for nodes V . Following this way, we obtain a series of networks $G(t) = (V(t), E(t))$. We remark that degree θ in the angle constraint (7) for topology optimization is fixed during the dynamics, and at an infeasible step, we use the topology of the previous step; though this may avoid the angle constraint for some nodes, the sum of angles for those nodes are expected to be close to the given θ .

θ	BN only	Neighbours			lifted
	20	10	20	all	all
160°	11619	4582	37571	599357	1435
170°	12521	4902	39747	621560	1437
180°	13378	5235	41918	643168	1439

TABLE 1

Average number of angle constraints according to the number of neighbours considered and the angle threshold θ for a set of 50 real-life test cases. Column “BN only” corresponds to the number of constraints added in the full model of paper [14] when angle constraints are applied to branching nodes only, with a neighbourhood threshold of 20. The last column shows the number of equivalent constraints generated with the lifting technique of section 3.1

3 MANAGING THE MODEL

In [14], the constraints in minimizing the total length of the network are:

- branching nodes (either persistent or non-persistent) have degree 3;
- other nodes have degree 1 to 3;
- the network is connected;
- no edges cross (plane graph);
- incompatibility triplets of edges are not selected simultaneously. This set of incompatibility triplets is built according to an angle threshold (e.g. $\theta = 180^\circ$) and the number of closest neighbours to take into account (20 in the paper). This choice of 20 neighbours is sufficient to produce the optimal solution satisfying the angle constraint for the real-life test cases [14].

Comparatively, we extend the optimization problem in the following aspects:

- the angle constraints are applied to all nodes that have degree 3 in the solution, not only the initially determined branching nodes; moreover, angle constraints are checked for all possible neighbours of each degree-3 node.
- potential end nodes are included;

The introduction of end nodes only requires a modification of bounds on corresponding edge variables. On the contrary, the angle constraints in this paper lead to a great number of additional constraints. In fact, we have $C_{n-n_1}^3$ exclusion constraints to take into account where $n = |V|$ and $n_1 = |V_e|$. We take a set of real-life test cases as an example. Compared to the model in [14], the number of angle constraints can increase considerably (shown in Table 1): for the same neighborhood limit, the angle constraints number is roughly multiplied by 3 on average, because of taking into account the persistent nodes, and by more than 50 if all nodes are considered for the neighbourhood. This has also an impact on the runtimes, as shown in Section 3.3 devoted to tests.

Section 3.1 shows how to take account of all the neighbours of a node. Angle checking is managed, as

compared to the model and strategy used in [14], in a more complete and more efficient way. Section 3.3 shows the results in terms of computation runtimes for the different algorithms applied to a series of data sets.

3.1 Angle constraints lifting

To reduce the number of angle constraints and enhance the runtimes of the optimization process, lifting techniques can be used.

For any node v , neighbour of u , let

$$U(v) = \{w \in V | w \text{ is a neighbour of } u, \text{ clockwise } \widehat{vuw} \leq \theta\}.$$

Then, the inequality

$$\sum_{w \in U(v)} x_{uw} \leq 2 \quad (12)$$

is valid for all 0-1 solutions of our problem (BP), covering in particular all angle constraints (7) with all 3 nodes in $U(v)$. Observe, that starting from such an angle constraint, we can also obtain inequality (12) by sequential lifting, a well-known operation from polyhedral combinatorics [15]. It is used to strengthen valid inequalities both for a tight linear description of 0-1 solutions and to speed up solution times within a branch and cut-framework for instance. These effects have been very clearly confirmed by our computational experience. To generate all these “lifted angle constraints” (12) in an irredundant way, one could imagine the following procedure:

1. Let the neighbours of node u be $v(1), \dots, v(p)$ in clockwise order; set $i := 1$ and $q(0) := 0$;
2. calculate the maximum value $q(i)$ modulo p such that $v(i)uv(q(i)) \leq \theta$;
3. if $q(i) > q(i-1)$, add the corresponding constraint (12), (but discard the (redundant) inequality for $j = 1$ and $q(1)$ whenever $q(j) = q(1)$ for some $j > 1$); else discard the corresponding constraint (12), which will be redundant;
4. if $i = p$, stop; else set $i := i + 1$ and go to 2.

This procedure requires the neighbours of any node to be sorted. To save runtime we have implemented the generation of lifted angle constraints directly, considering each node u , each neighbour v of u , and the associated set $U(v)$ together with inequality (12). The number of inequalities can be further reduced by checking dominance¹ between angle constraints. Some tests (not presented here) show that dominance checking is costly in terms of runtimes and does not help to decrease the overall runtime of the optimization process.

3.2 Problem resolution

In [14] we have detailed a series of resolution techniques, which were based on either LP relaxation or BP for solving the topology optimization problem. The reference

1. Given two vectors $x, y \in \{0-1\}^n$, we say that x dominates y if $x_i \geq y_i$ for all $i = 1, \dots, n$ [15].

method is BP. It works as follows : The initial BP is given by the objective function (1) and constraints (2), (3), (4) and (6). It iteratively solves this problem, checking the resulting subgraph for crossing edges, a subset of angle constraints, and connectivity constraints. In the case of crossing or angle constraints not being satisfied, the corresponding inequalities (8) and (7) are added to the binary problem. Connectivity is ensured by computing connected components, and adding an inequality of type (5) for each of them. The problem is solved again, until the resulting subgraph is plane, connected and respects all of the considered angle constraints. The most efficient combination determined in [14] is based on a relaxation technique starting with the same initial problem except the binary constraints (6) that we replace by their linear relaxations $0 \leq x_{uv} \leq 1$. It is coupled with different separation procedures for the root node of the branch & cut search procedure. When separation procedures fail to find the cutting planes, the BP algorithm is applied, which itself relies on the branch & cut tool of the MIPS. The best combination for LP resolution uses recursive and parametric cuts, and is referred to as LP_{rk} in [14]. It corresponds to the *relax* option of the General Algorithm 1.

The use of the LP solver implies the possibility of fractional optimal solutions. The basic procedure to "cut off" such fractional solutions is based on a min-cut algorithm. If an $s - t$ -cut leads to a cut value < 1 , the connectivity constraint is violated. This situation is detected using the Stoer-Wagner algorithm [16]. If V_1, V_2 is the corresponding partition of V , we add the constraint

$$\delta(V_1) \geq 1 \quad (13)$$

The separation procedures we use are extensions of this basic technique. They rely on a p -partition $P = \{V_1, V_2, \dots, V_p\}$ of the node set V . In such a case we add the constraint

$$\frac{1}{2} \sum \delta(V_i) \geq (p - 1) \quad (14)$$

The problem is now to find a partition of V , whose inequality is violated by the current optimal and fractional solution.

We have implemented 2 separation procedures for this type. The *r-cut* procedure is based on a recursive splitting of partitions using a min-cut algorithm, whereas the *k-cut* procedure is a 1-edge-like contraction procedure.

- The *r-cut* procedure for multi-cuts is inspired by [17]. Given a p -partition, find a minimum cut in each part of it. For the cut with minimum value among all these min-cuts, break the associated component to obtain a $(p + 1)$ -partition.
- The *k-cut* procedure for multi-cuts goes as follows. The partition is induced by the components of the graph $G' = (V, E' = \{e \in E | x_e \geq \alpha\})$, with $\alpha \in [0..1]$ as given by the current optimal solution. This separation procedure is applied for $\alpha = 0.8, 0.6, 0.4$.

In order to further reduce the resolution time, we can also apply these separation procedures to the inner nodes of the branch & cut search tree. This procedure corresponds to the *fullBC* option of the General Algorithm.²

Algorithm 1: General Algorithm

Data: a set of points $V = V_b \cup V_e \cup V_p$, a number of iterations I , an angle value θ , and options *relax* and/or *fullBC*

Result: a series of connected plane subgraphs

```

1 begin
2    $V_s := V$ 
3    $i := 1$ 
4   graph  $G := (V_s, E, w)$ , with  $E := V_s^2$ ,  $w_{uv}$  the
      Euclidean distance between  $u$  and  $v$ 
5   generate problem  $P$  with constraints (2), (3) and
      (4)
6   add lifted angle constraints of section 3.1
      obtained from (7) add constraints (6) to  $P$ 
7    $f := \text{solve}(P, \text{options})$ 
8   if  $G_f = \emptyset$  then
9     restore  $G_f$  from iteration  $i - 1$ 
10    compute objective function according to
        current node locations
11    goto 25
12  end
13  if  $G_f = (V, f)$  is connected then
14    if  $\text{checkPlanarity}(f) = OK$  then goto 25
15    else add corresponding constraints (8)
16  end
17  if  $G_f = (V, f)$  is not connected then
18    compute connected component set
         $W = \{W_1, \dots, W_p\}$ 
19    add corresponding constraints (5)
20  end
21  goto 7
22
23  if  $i = N + 1$  then goto 27
24  else  $i := i + 1$ 
25  apply dynamics rules
26  goto 4
27 end
```

The next section shows our results in terms of computation runtimes for the different algorithms applied to a series of data sets.

3.3 Tests

We apply the different versions of our General Algorithm for ER dynamics to a series of 50 data sets. We first compare the runtimes of the best approach presented

² In fact, only integral constraints are checked at each inner node of the B & C search tree because tests show that other separation procedures are too costly for both BP and LP_{rk} strategies in root node solving.

in [14], by using non-lifted angle constraints, to the runtimes obtained with lifted constraints for initial and optimal network determination of the 50 data sets. Then, we compare the runtimes of the different algorithms on the same 50 data sets for dynamics algorithm testing. For each of them, 10 iterations of the dynamics algorithm are run, and overall runtimes are reported. Missing data correspond to infeasible initial problems.

- 1) Incompatibility vs lifted angle constraints. The algorithm LP_{rk} of [14] considers angle incompatibilities generated by the 20 closest neighbours of a node. We consider here all of the node neighbours, with lifted angle inequalities. The same separation procedures are used (LP_{rk} algorithm from [14] and BP + relax option of the General Algorithm 1). Figure 2 shows the runtimes for the data set processing (one iteration). A limit of 5 minutes (300 secs) is set for the runtime.

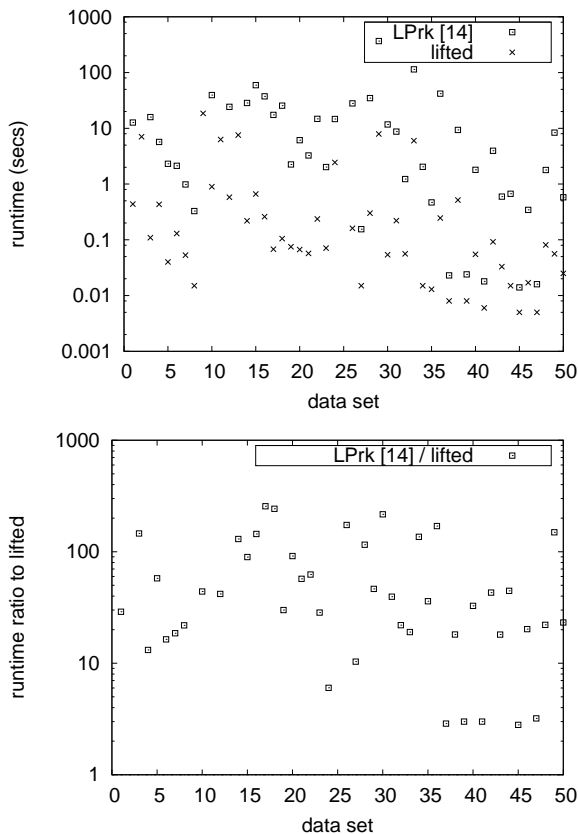


Fig. 2. (top) Runtimes for data sets with angle constraints on limited (20) neighbourhood from [14] and with lifted angle constraints of section 3.1. (bottom) ratio between the runtimes of the 2 approaches. Here, the angle $\theta = 170^\circ$ is used.

As expected, a huge reduction of inequalities in the BP and LP formulation (see table 1) leads to a large improvement of runtimes. And this is realized conjointly with the consideration of all nodes instead of just a number of closest neighbours as in [14].

- 2) Runtimes comparison for dynamics processing. We

apply the different versions of the General Algorithm for computing the dynamics of 50 data sets over 10 iterations for an angle constraint of $\theta = 180^\circ$. Tested versions are 1) BP, the general algorithm without options, 2) BP+BaC, with full branch & cut option, 3) LP, i.e., BP with relax option, and 4) LP+BaC, i.e., BP + both relax and full branch & cut options. Tests have been performed on a 24 core SMP linux system with 120 GB of memory. The average runtime ratio relative to the BP algorithm runtime for the feasible data sets is, respectively, 18.7% for BP+BaC, 16.9% for LP, and 95.7% for LP+BaC. Thus, both BP+BaC and LP are approximately 5 to 6 times faster than the direct BP solver.

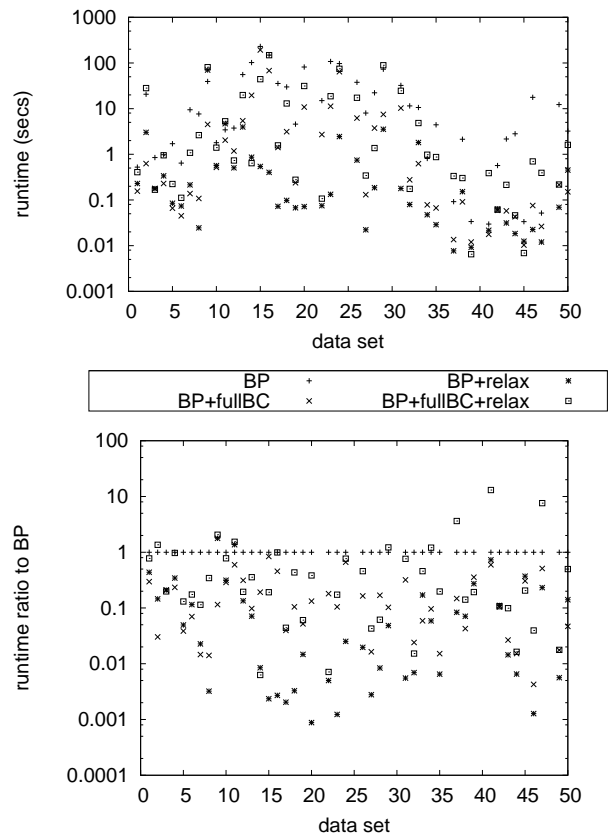


Fig. 3. (top) Runtimes in seconds for the data set with lifted angle constraints and the different option combinations from General Algorithm. (bottom) Ratio to the reference algorithm BP (without options). Here $\theta = 180^\circ$ is used.

The fastest is LP for 55% of the testcases, BP+BAC for 30% and LP+BAC for 8% of the testcases. The remaining 7% are infeasible ones. So even if it is not the fastest for every data set of the testbench, LP provides an overall good performance in terms of runtimes. The utility of the full branch & cut strategy is only relevant when the solver MIPS is solely used (BP+BaC algorithm). In addition, when LP is used, it implies inefficient cut searches. A possible reason is that most of them are

dominated by the ones found during the relaxed phase of the LP algorithm.

To conclude, the LP approach is efficient, and appears to be more performant in terms of runtimes, as compared to the second best candidate BP+BaC, for the considered testbench. It is also more robust, in the sense that no case implies excessive runtimes: the cumulated runtime for the whole testbench is 96 secs for LP and 419 secs for BP+BaC.

Concerning network optimization, the techniques developed in [14] are applicable, but they necessitate an adaptation in order to take into account complete angle constraints. The lifted angle formulation has shown to be effective for our data sets. Runtimes are low enough for the technique to be applied on series of successive networks, modified according to the ER network dynamics. At each step, the network is considered independently. However, solutions found by the solver in previous steps are not taken into account (except infeasible cases). This should be exploited to speed up resolution times. However, this is not straightforward, since crossing and angle constraints are not the same from one step to another. Furthermore, even if constraints are still satisfied, modifications of distances imply modifications of the objective value. A preliminary sensitivity analysis shows that most often, there is no slack in the solution values, thanks to the BP formulation of the problem.

4 ER NETWORK DYNAMICS

We now apply the proposed optimization technique to the real experimental data. We first solve the problem described in Section 2.1 for instantaneous ER networks from experiments in the native condition and then use our optimization procedure to model the ER network dynamics starting with an initial state of nodes being abstracted from the experimental data. Initially, infeasible cases are excluded for the whole analysis.

4.1 Instantaneous network topology

Figure 4 shows examples of optimal graphs for given sets of nodes together with the abstracted ER network for different values of θ in the angle constraints. To quantify the difference between these optimal graphs and the abstracted ER network we may use the concepts of *effective similarity*, *error correcting matching* and *angle distribution*.

We call effective similarity [14] $s(G_1, G_2)$ between two graphs G_1, G_2 , the average of two percentages: percentage of edges in G_1 that do appear in G_2 and percentage of edges in G_2 that do appear in G_1 . This measurement can be calculated via the adjacency matrices of the two graphs and it ranges from 0 to 1; it equals 0 if none of the edges coincides in the two graphs and it equals 1 if the connecting structures between the two graphs are the same. Meanwhile, in analogy to the notion of error correcting graph matching [18], we call normalized

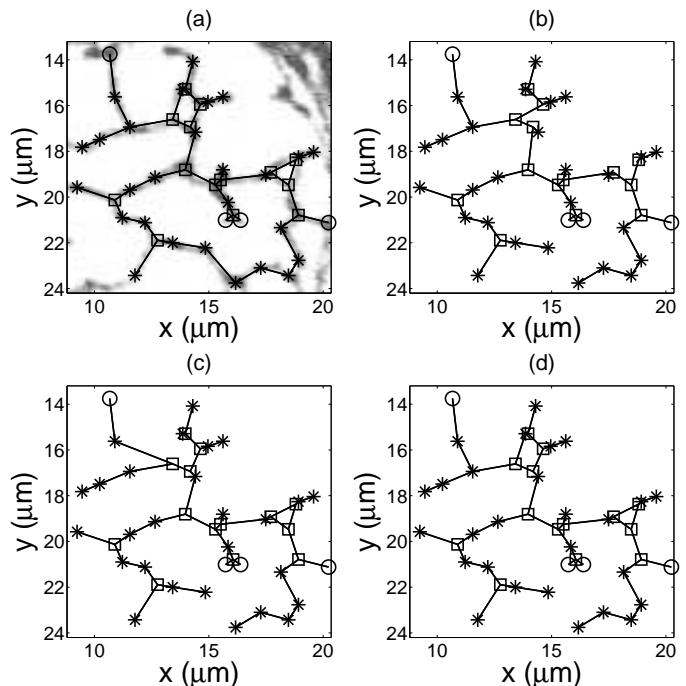


Fig. 4. A comparison between: an abstracted ER network (a), an optimal graph from the model with $\theta = 0^\circ$ (b), $\theta = 170^\circ$ (c) or $\theta = 180^\circ$ (d). Their total lengths of graphs are $49.73\mu\text{m}$, $54.64\mu\text{m}$, $47.54\mu\text{m}$ and $47.70\mu\text{m}$, respectively. The effective similarities of graphs in (b-d) with respect to the abstracted network in (a) are 0.88, 0.95 and 0.99, respectively, while the normalized error correcting matchings with respect to the abstracted network in (a) are 0.011, 0.004 and 0.001, respectively. The underlying ER image in (a) is from the imaging data in [5] while its geometric graph is abstracted using the image processing method in [9].

error correcting matching [14] $m(G_1, G_2)$ between two graphs G_1, G_2 over the same set of nodes, the ratio of the minimum number of edit operations (edge addition and edge deletion) necessary to transform one graph into the other, to the number of edges of the complete graph with the same set of nodes. This measurement can be calculated via the adjacency matrices as well.

Table 2 shows that overall the optimal graphs with θ close to 180° show a higher similarity and a lower error correcting matching than those with no angle constraint ($\theta = 0^\circ$). In addition, Figure 5 shows that the distribution of angles of degree-3 nodes in the model with both degree and angle constraints is not significantly different from that of the abstracted ER networks. Consistent with [14], these results suggest that the optimal graphs from the proposed optimization problem with θ around 180° is close to the corresponding ER networks, and that beyond degree constraints, angle constraints are necessary for understanding the principles governing the ER network geometry.

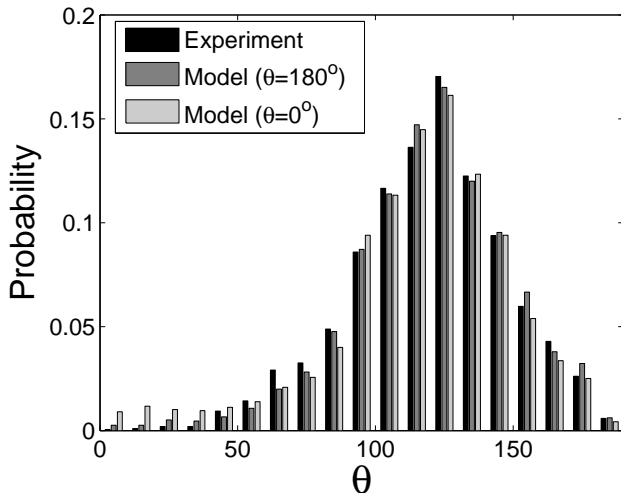


Fig. 5. Angle distribution of degree-3 nodes from the experimental data in the native state and from the optimal graphs with the instantaneous abstracted nodes. A two-sample Kolmogorov-Smirnov test suggests that the distribution from the experimental data is the same as that from the optimal graph with $\theta = 180^\circ$ ($p=0.0672$) and is different to that from the optimal graph with $\theta = 0^\circ$ ($p=0.9647$).

	$\theta = 0^\circ$	$\theta = 170^\circ$	$\theta = 180^\circ$
s	0.904 ± 0.0097	0.939 ± 0.0077	0.934 ± 0.0188
m	0.011 ± 0.0014	0.007 ± 0.0005	0.008 ± 0.0014

TABLE 2

A comparison of optimal solutions G_1 from different models to the abstracted ER network G in terms of similarity $s(G, G_1)$ and error correcting matching $m(G, G_1)$. Data indicate the mean \pm sem. T tests show there is a significant difference on $s(G, G_1), m(G, G_1)$ between the case where G_1 is the optimal graph for $\theta = 0$ and the case where G_1 is an optimal graph for $\theta = 170^\circ$ ($p = 0.0063, p = 0.0199$) and no significant difference between $\theta = 170$ and $\theta = 180^\circ$ ($p = 0.668, p = 0.504$); sample size $n = 46 - 50$ (infeasible problems are excluded).

4.2 Network dynamics

We first apply our dynamics model to a small ER network in latrunculin B (treated) cells where networks dynamics are independent of actin and myosin and are less dynamic and observations of ER tubule network dynamics are possible. In particular, we revisit a region with 6 persistent nodes and 2 non-persistent branching nodes, as shown in Figure 6 (top panels) in the treated condition. Note that the network topologies are changing among a set of Steiner tree topologies [9]. The dynamics in this region has been modeled in [9] by using a Langevin approach where network topology is determined simply by comparison with the possible

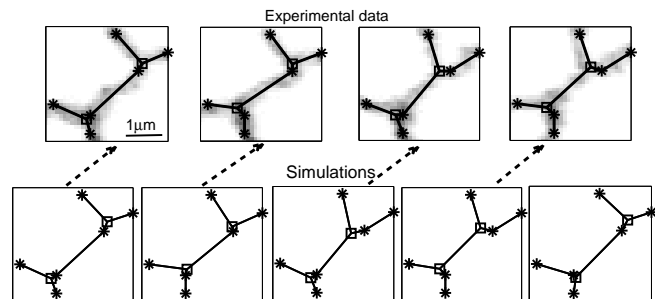


Fig. 6. Top: four different graph structures observed from treated ER network in a small region of experimental data. Bottom: five network topologies observed from stochastic simulations up to 200 s; four of them relate to those in experimental observed topologies (indicated by the arrows). The markers “*” and “□” are the same as in Figure 1. The top panels are modified from figures in [9].

Steiner trees. Here, we construct the network topology from the optimization problem proposed in Section 2.1 at each time step during the dynamics, and this proposed approach represents the network remodeling observed from experimental data as shown in Figure 6. The additional topology observed in simulation may be not necessarily observed in the experimental data due to the limited spatial resolution and the short time recording.

Next, we apply the simulation to larger regions of the ER networks. These regions were chosen from the experimental data in the native state; note, that larger regions of ER networks with no cisternae are not available in the treated experimental data. With the initial state of nodes determined from the experimental data at one frame in a region with no cisternae, we simulate the time evolution of the network by applying the dynamics of branching nodes. We find that the angles of the branching nodes from optimal graphs (with $\theta = 180^\circ$) in the dynamics statistically follow the same distribution as that in the experimental data (data not shown, with $p=0.9998$ in a two-sample Kolmogorov-Smirnov test). This suggests that the proposed dynamic modeling approach does exhibit quantitative agreement with the experimental data as far as angle distribution is concerned. In addition, by moving the branching nodes, the network topology changes accordingly, and represents features observed from experimental data, such as the loop closing/opening (i.e., appearing/disappearing of a cycle within the network) as shown in Figure 7.

Analogous to small network dynamics in Figure 6, for a larger set of nodes the network topology during dynamics also changes among a set of topologies; from [9] we know that these topologies are expected to be those from Euclidean Steiner networks. Figure 8 illustrates an example of network dynamics showing the oscillation of the total length of networks (in the geometric aspect) and its error correcting matching with the initial network topology (in the topological aspect) in the time evolution. We further characterize the dynamics of the

network by the number of topologies N_{top} that could be observed during the dynamics for a given initial state of the network nodes. Note, that $N_{top} = \lim_{t \rightarrow \infty} \{N_{top}(t)\}$ where $N_{top}(t)$ is the number of topologies observed up to time t . In the dynamics, the quantity $N_{top}(t)$ is a random variable. For a given initial state, it is clear that the topology number N_{top} is finite. In contrast to small networks, this quantity could be much larger for larger networks; see the bottom panel in Figure 8 (b) in comparison to that in Figure 6. In addition, comparing $N_{top}(t)$ and the matching in particular during the time $[10s - 25s]$ where number of topologies remain the same whereas the matching is oscillating; this again suggest changes of the network topologies.

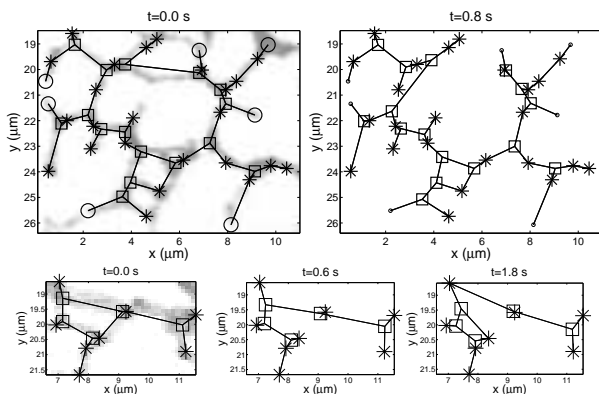


Fig. 7. Two examples of loop closing/opening from the network dynamics by moving non-persistent branching nodes in the simulations. The imaging data for the initial states of network nodes is taken from [5].

We now use the quantity $N_{top}(t)$ to compare network dynamics when applying different dynamics. In particular, we consider time evolution of networks with/without dynamics of persistent nodes and/or dynamics of end nodes in addition to the movement of branching nodes. Figure 9 shows an example of the comparison where the number of topologies is averaged over three simulations for a given initial state. This example illustrates an increase in the number of network topologies when adding dynamics of persistent nodes or end nodes. We calculate the averaged number of $N_{top}(t)$ for each real data set and found that about 62% cases, 73% and 77% of the initially feasible cases have an increase in the number of topologies when adding dynamics of persistent nodes (case I), dynamics of end nodes (case II), or dynamics of both persistent and end nodes (case III), up to time $t = 10s$. In addition, we calculate the differences between each of the three cases and that with dynamics of branching nodes only on the quantity $N_{top}(t)$ and find that the average differences are 0.6, 4.4 and 4.2 topologies up to time $t = 10s$ and are 4.4, 15.1 and 20.2 topologies up to time $t = 50s$, respectively. These suggest that for the abstracted ER networks in the native state, with additional dynamics of end nodes or persistent nodes, the network remodeling

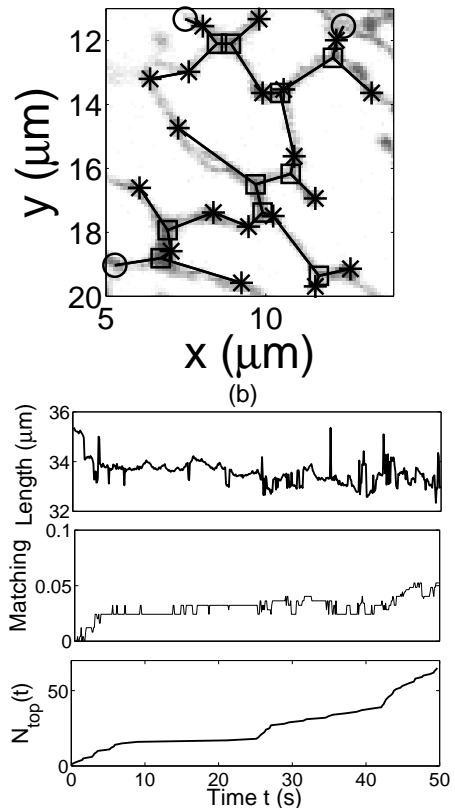


Fig. 8. Illustration of network dynamics from simulations when applying movement of non-persistent branching nodes only. (a) shows the initial network together with the experimental data whose total length is $37.3\mu m$; (b) shows the oscillation pattern in terms of the dynamics of the total length (top), that of the error correcting matching $m(G(t), G(0))$ (middle) and the number of topologies (bottom) up to the time t where $G(t)$ is the optimal graph at time t from simulations. The underlying imaging data is taken from [5].

could be enhanced, even without taking into account active components (such as actin and myosin motors) that drives the network dynamics.

5 DISCUSSION

ER network remodeling and the dynamic nature of this process have not been adequately described in a quantitative manner except for small networks in latrunculin B (treated) conditions. Questions relating to whether the entire ER network maintains an optimal configuration/topology in terms of length minimization remain unanswered. Understanding these basic properties of the ER provides a platform from which to determine and measure the effects of the molecular components (such as actin, myosin and reticulons) controlling network modeling. In this article, we focus on ER networks formed by tubules and introduce a computationally manageable approach to construct the ER network dynamics in a considerable large domain. In particular, we consider

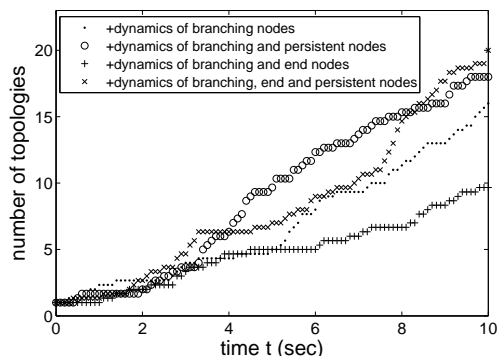


Fig. 9. Illustration of dynamic variability characterized by the number of topologies applying different dynamics. Each data point is the average from three simulations.

both the dynamics of network movement and changes of network topologies. The network movement extends the Langevin approach used in [9] by including dynamics of end nodes and persistent nodes while network topologies are determined by minimizing the network length with angle and degree constraints, extending the optimization procedure in [14]. Note that dynamics of the nodes within the network generate the remodeling of the network.

The proposed optimization problem for network topology gives optimal solutions close to the actual network geometry in single snap shots (i.e., time points) of the experimental data. Differences between the actual network and the optimal graphs (if any) will allow future investigation into new features governing network dynamics. In addition, the simulation of network dynamics recaptures ER network features observed from the native experimental data such as the angle distribution and loop closing/opening. Quantities have been developed to analyze the network dynamics by calculating the number of topologies during network remodeling and have been compared between different dynamics. In particular, we show that network remodeling could be enhanced by including dynamics of end or/and persistent nodes in addition to the dynamics of branching nodes.

We have not yet been able to relate this type of quantification of network dynamics to the dynamics of the native experimental data. This is due to the highly motile nature of the native ER network and the current limited time resolution in the experimental data preventing detailed observations of node dynamics. Also, our current type of analysis is based on tubule dynamics, whereas in fact the native network can readily change to cisternal forms preventing such modeling to be undertaken. The highly dynamic ER is also expected to be influenced by the actin and myosin dependent processes as well as cytoplasmic streaming [5], [8]. Under the treated condition the role of myosin, motors that traverse the actin, and actin dynamics are absent. Modeling elements related

with the actin and myosin activities are beyond the scope of this work as the specific molecular components involved in these processes and the mechanochemical characteristics of the myosin motors (e.g., step length, processivity, directionality, force generation) that drive the motion of the ER are unknown ([5], [19], [20] and references therein). However, it does go beyond our current understanding and provides an approach to model the ER which solely contains tubules and does not undergo drastic remodeling (e.g. keeping the same set of nodes). Using an increased time resolution for recording ER images (when available), a detailed quantitative comparison at a local time of the actual tubule network dynamics and simulated network dynamics would then be possible.

In addition, this model could be generalized to incorporate a number of features that may be important for the ER network dynamics in the native state. This includes: 1) deterministic cytoplasmic streaming, 2) Brownian forces from the cytoplasm (which might be heterogeneous); 3) creation/destruction of nodes (besides the persistent nodes); 4) active molecular motors (e.g., myosin); 5) dramatic transition between tubules and sheets. By including the above elements when necessary, a more robust model of ER network dynamics will be generated. This can then be used to understand how small genetic perturbations can drastically alter the network. Since the ER network in plants plays pivotal roles in processes which may be affected by ER network morphology and remodeling (such as protein synthesis and viral trafficking), a greater understanding and long term quantitative measures of the network are essential.

REFERENCES

- [1] T. Levine and C. Rabouille, Endoplasmic reticulum: one continuous network compartmentalized by extrinsic cues, *Curr. Opin. Cell. Biol.* 17:362-368. 2005.
- [2] U. Goyal and C. Blackstone, Untangling the web: Mechanisms underlying ER network formation, *Biochim. Biophys. Acta* 1833:2492-2498. 2013.
- [3] E. M. Lynes, and T. Simmen, Urban planning of the endoplasmic reticulum (ER): how diverse mechanisms segregate the many functions of the ER. *Biochim. Biophys. Acta.* 1813:1893-1905. 2011.
- [4] I. A. Sparkes, T. Ketelaar, N. C. de Ruijter, and C. Hawes. Grab a Golgi: laser trapping of Golgi bodies reveals in vivo interactions with the endoplasmic reticulum. *Traffic.* 10:567-571. 2009.
- [5] I. A. Sparkes, J. Runions, C. Hawes, and L. Griffing, Movement and remodeling of the Endoplasmic Reticulum in nondividing cells of Tobacco leaves, *Plant Cell* 21:3937-3949. 2009.
- [6] P. Wang, T. J. Hawkins, C. Richardson, I. Cummins, M. J. Deeks, I. Sparkes, C. Hawes, and P. J. Hussey. The plant cytoskeleton, NET3C, and VAP27 mediate the link between the plasma membrane and endoplasmic reticulum. *Curr. Biol* 24: 1397. 2014.
- [7] L. R. Griffing. Networking in the endoplasmic reticulum. *Biochem. Soc. Trans.* 38:747-753. 2010.
- [8] I. A. Sparkes, L. Frigerio, N. Tolley, and C. Hawes, The plant endoplasmic reticulum: a cell-wide web. *Biochem. J.* 423:145-155. 2009.
- [9] C. Lin, Y. Zhang, I. A. Sparkes, and P. Ashwin, Structure and Dynamics of ER: Minimal Networks and Biophysical Constraints, *Biophys. J.* 107: 763-772. 2014.

- [10] A. N. Boučekhima, L. Frigerio, and M. Kirkilionis, Geometric quantification of the plant endoplasmic reticulum. *J. Microsc.* 234:158-172. 2009.
- [11] A. T. Fomenko, *The Plateau Problem: Historical Survey*. Gordon & Breach, Williston, VT. 1989.
- [12] M. Terasaki, T. Shemesh, N. Kasthuri, R. W. Klemm, R. Schalek, K. J. Hayworth, A. R. Hand, M. Yankova, G. Huber, J. W. Lichtman, T. A. Rapoport, and M. M. Kozlov, Stacked endoplasmic reticulum sheets are connected by helicoidal membrane motifs. *Cell.* 154:285-296. 2013.
- [13] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner tree problem*, *Ann. Discrete Math.* Vol. 53, Elsevier Science Publishers B.V., North-Holland, Amsterdam. 1992.
- [14] L. Lemarchand, R. Euler, C. Lin, and I. Sparkes. Modeling the Geometry of the Endoplasmic Reticulum Network. *Algorithms for Computational Biology Lecture Notes in Computer Science 8542*: 131-145. 2014.
- [15] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, Wiley-Interscience, 1988.
- [16] M. Stoer and F. Wagner, A simple min-cut algorithm, *J. ACM* 44:585-591. 1997.
- [17] F. Barahona, On the k-cut problem, *Oper. Res. Lett.* 26:99-105. 2000.
- [18] H. Bunke, Error correcting graph matching: on the influence of the underlying cost function, *IEEE Trans. Pattern Anal. Mach. Intell.* 21:917-922. 1999.
- [19] L. R. Griffing, H. T. Gao, I. Sparkes. ER network dynamics are differentially controlled by myosins XI-K, XI-C, XI-E, XI-I, XI-1, and XI-2. *Front Plant Sci.* 5:218. 2014.
- [20] I. Sparkes. Recent Advances in Understanding Plant Myosin Function: Life in the Fast Lane. *Mol. Plant* 5:805-812. 2011.



Imogen Sparkes received her PhD in plant biology in 2003 from the University of Leeds, UK. Between 2003 and 2012 she tackled numerous plant cell biology topics relating to the biogenesis and dynamics of organelles including the ER. She joined the University of Exeter as a cell biology lecturer in 2012. Her research group is focused on plant organelle dynamics.



Congping Lin received her M.Sc degree in Mathematics in 2009 at Zhejiang Normal University, China and PhD in System biology in 2012 at University of Exeter, UK. She is currently an Associate Research Fellow at the University of Exeter. Her research interest lies in mathematical biology and nonlinear dynamics, focusing on mathematical modeling of the complex dynamics in living cells.



Laurent Lemarchand received his PhD in computer science in 1998 at the University of Rennes, France. Since 1999, he is associate professor at the Univ. Bretagne Occidentale, Brest, France. He is currently member of the Lab-STICC laboratory. His research topics include linear programming, meta-heuristics optimization and silicon compilation.



Reinhardt Euler is Professor of Computer Science at Lab-STICC laboratory (CNRS 6285) of the University of Brest. He obtained a PhD in Mathematics at the University of Cologne and has held visiting professorships at Grenoble University, the University of British Columbia in Vancouver and Carnegie Mellon University in Pittsburgh. His research interests include combinatorial algorithms and optimization, graph theory and the efficient solution of large scale real life problem instances.