Efficient Algorithms for Moral Lineage Tracing

Markus Rempfler[1]*, Jan-Hendrik Lange[2]*, Florian Jug[3], Corinna Blasse[3], Eugene W. Myers[3], Bjoern H. Menze[1] and Bjoern Andres[2]

[1] *Institute for Advanced Study & Department of Informatics, Technical University of Munich*
[2] *Max Planck Institute for Informatics, Saarbrücken*
[3] *Max Planck Institute of Molecular Cell Biology and Genetics, Dresden*

Abstract

*Lineage tracing, the joint segmentation and tracking of living cells as they move and divide in a sequence of light microscopy images, is a challenging task. Jug et al. [20] have proposed a mathematical abstraction of this task, the moral lineage tracing problem (*MLTP*) whose feasible solutions define a segmentation of every image and a lineage forest of cells. Their branch-and-cut algorithm, however, is prone to many cuts and slow convergences for large instances. To address this problem, we make three contributions: Firstly, we improve the branch-and-cut algorithm by separating tighter cutting planes. Secondly, we define two primal feasible local search algorithms for the* MLTP*. Thirdly, we show in experiments that our algorithms decrease the runtime on the problem instances of Jug et al. considerably and find solutions on larger instances in reasonable time.*

1 Introduction

Recent advances in microscopy techniques have enabled biologists to observe organisms on a cellular level with higher spatio-temporal resolution than before [13, 17, 37]. Analysis of such microscopy sequences is key to several open questions in modern biology, including embryonic development of complex organisms [24, 25], tissue formation [18] or the understanding of metastatic behavior of tumor cells [43]. However, to get from vast amounts of raw microscopy images to biologically or clinically relevant quantities, such as cell motility, migration patterns and differentiation schedules, robust methods for *cell lineage tracing* are required and have therefore received considerable attention in the past [2, 3, 14, 29, 30, 31].

Typically, cell lineage tracing is considered a two step problem: In the first step, individual cells are detected and segmented in every frame. Then, in the second step, individual cells are tracked over time and, in case of a cell division, are linked to their ancestor cell, finally resulting in the lineage forest. The latter subproblem is complicated by cells that enter or leave the field of view, or low temporal resolution that allows large displacements or even multiple consecutive divisions within one time step.

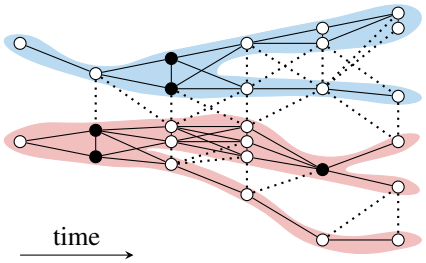

Figure 1 The *moral lineage tracing problem (*MLTP*)*[1]: Given a sequence of images decomposed into cell fragments (depicted as nodes in the figure), cluster fragments into cells in each frame and *simultaneously* associate cells into lineage forests over time. Solid edges indicate joint cells within images and descendant relations across images. Black nodes depict fragments of cells about to divide.

In addition to this, mistakes made in the first step, leading to over- or undersegmentation of the cells, propagate into the resulting lineage forest and cause spurious divisions or missing branches, respectively. The tracking subproblem is closely related to multi-target tracking [12, 36, 40] or reconstruction of tree-like structures [16, 33, 38, 39], with the most important difference that the objects can divide (only) into two. It is thus often cast as an optimization problem [21, 23, 32, 34, 35], dealing with some of the mentioned difficulties by being able to discard misdetections [23] or by providing and selecting from multiple detection hypotheses [34, 35].

Jug et al. [20], on the other hand, have proposed a rigorous mathematical abstraction for these two subtasks, the *moral lineage tracing problem (*MLTP*)*. It is a hybrid of the *minimum cost multicut problem (*MCMCP*)* problem, which has been extensively studied for image decomposition [5, 6, 7, 8, 9, 10, 11, 22, 27, 28, 41, 42], and the *minimum cost disjoint arborescence* problem, variations of which have been applied to reconstruct lineage forests in [21, 23, 32, 35, 34] or tree-like structures [16, 39, 38]. Feasible solutions to the MLTP define not only a valid cell lineage forest over time, but also a segmentation of the cells in every frame (*cf.* Fig. 1). Solving this optimization problem therefore tackles both subtasks – segmentation and

*contributed equally.

[1] The figure is a correction of the one displayed in [20], a template was kindly provided by the authors.

tracking – simultaneously. While Jug et al. [20] demonstrate the advantages of their approach in terms of robustness, they also observe that their branch-and-cut algorithm (as well as their cutting-plane algorithm of the corresponding relaxation) is prone to a large number of cuts and exhibits slow convergence on large instances. That, unfortunately, prevents many potential use cases of the MLTP in practice, since it would be too computationally expensive.

Contributions. In this paper, we address this disadvantage. We make three contributions: Firstly, we improve the branch-and-cut algorithm of [20] by separating tighter cutting planes. Secondly, we devise two heuristics for the MLTP, both of which are primal feasible local search algorithms inspired by the heuristics of [27] for the MCMCP. Finally, we evaluate our algorithms on the problem instances of [20], where we show that our algorithms compare favorably against the original branch-and-cut algorithm, and on two additional, larger instances, where we are able to find near-optimal solutions in reasonable time.

2 Background and Preliminaries

Consider a set of $\mathcal{T} = \{0, \dots, t_{\text{end}}\}$ consecutive frames of microscopy image data. In moral lineage tracing, we seek to jointly segment the frames into cells and track the latter and their descendants over time. This problem is formulated by [20] as an integer linear program (ILP) with binary variables for all edges in an undirected graph as follows.

For each time index $t \in \mathcal{T}$, the node set V_t comprises all cell fragments, *eg.* superpixels, in frame t. Each neighboring pair of cell fragments are connected by an edge. The collection of such edges is denoted by E_t. Between consecutive frames t and $t+1$, cell fragments that are sufficiently close to each other are connected by a (temporal) edge. The set of such inter frame edges is denoted by $E_{t,t+1}$. By convention, we set $V_{t_{\text{end}}+1} = E_{t_{\text{end}}+1} = E_{t_{\text{end}},t_{\text{end}}+1} = \emptyset$. The graph $G = (V, E)$ with $V = \bigcup_{t\in\mathcal{T}} V_t$ and $E = \bigcup_{t\in\mathcal{T}}(E_t \cup E_{t,t+1})$ is called *hypothesis graph* and illustrated in Fig. 1. For convenience, we further write $G_t = (V_t, E_t)$ for the subgraph corresponding to frame t and $G_t^+ = (V_t^+, E_t^+)$ with $V_t^+ = V_t \cup V_{t+1}$ and $E_t^+ = E_t \cup E_{t,t+1} \cup E_{t+1}$ for the subgraph corresponding to frames t and $t+1$.

For any hypothesis graph $G = (V, E)$, a set $L \subseteq E$ is called a *lineage cut* of G and, correspondingly, the subgraph $(V, E \setminus L)$ is called a *lineage (sub)graph* of G if

1. For every $t \in \mathcal{T}$, the set $E_t \cap L$ is a multicut[2] of G_t.
2. For every $t \in \mathcal{T}$ and every $\{v, w\} \in E_{t,t+1} \cap L$, the nodes v and w are not path-connected in the graph $(V_t^+, E_t^+ \setminus L)$.
3. For every $t \in \mathcal{T}$ and nodes $v_t, w_t \in V_t$, $v_{t+1}, w_{t+1} \in V_{t+1}$ with $\{v_t, v_{t+1}\}, \{w_t, w_{t+1}\} \in E_{t,t+1} \setminus L$ and such that v_{t+1} and w_{t+1} are path-connected in $(V, E_{t+1} \setminus L)$, the nodes v_t and w_t are path-connected in $(V, E_t \setminus L)$.

[2] A multicut of $G_t = (V_t, E_t)$ is a subset $M \subseteq E_t$ such that for every cycle C in G_t it holds that $|M \cap C| \neq 1$, cf. [15].

For any lineage graph $(V, E \setminus L)$ and every $t \in \mathcal{T}$, the non-empty, maximal connected subgraphs of $(V_t, E_t \setminus L)$ are called *cells* at time index t. Furthermore, Jug et al. call a lineage cut, respectively lineage graph, *binary* if it additionally satisfies

4. For every $t \in \mathcal{T}$, every cell at time t is connected to at most two distinct cells at time $t+1$.

According to [20], any lineage graph well-defines a lineage forest of cells. Moreover, a lineage cut (and thus a lineage graph) can be encoded as a 01-labeling on the edges of the hypothesis graph.

Lemma 1 ([20])**.** For every hypothesis graph $G = (V, E)$ and every $x \in \{0,1\}^E$, the set $x^{-1}(1)$ of edges labeled 1 is a lineage cut of G iff x satisfies inequalities (1) – (3):

$$\forall t \in \mathcal{T} \forall C \in \text{cycles}(G_t) \forall e \in C: \quad x_e \leq \sum_{e' \in C \setminus \{e\}} x_{e'} \tag{1}$$

$$\forall t \in \mathcal{T} \forall \{v, w\} \in E_{t,t+1} \forall P \in vw\text{-paths}(G_t^+): \quad x_{vw} \leq \sum_{e \in P} x_e \tag{2}$$

$$\forall t \in \mathcal{T} \forall \{v_t, v_{t+1}\}, \{w_t, w_{t+1}\} \in E_{t,t+1} (\text{with } v_t, w_t \in V_t) \; \forall S \in v_t w_t\text{-cuts}(G_t) \forall P \in v_{t+1} w_{t+1}\text{-paths}(G_{t+1}): \quad 1 - \sum_{e \in S} (1 - x_e) \leq x_{v_t v_{t+1}} + x_{w_t w_{t+1}} + \sum_{e \in P} x_e \tag{3}$$

Jug et al. refer to (1) as *space cycle*, to (2) as *space-time cycle* and to (3) as *morality* constraints. We denote by X'_G the set of all $x \in \{0,1\}^E$ that satisfy (1) – (3). For the formulation of the additional *bifurcation* constraints, which guarantee that the associated lineage cut is binary, we refer to ([20, Eq. 4]). The set X_G collects all $x \in X'_G$ that also satisfy the bifurcation constraints.

Given cut costs $c : E \to \mathbb{R}$ on the edges as well as *birth* and *termination* costs $c^+, c^- : V \to \mathbb{R}_0^+$ on the vertices of the hypothesis graph, [20] defines the following *moral lineage tracing problem* (MLTP)

$$\min_{x, x^+, x^-} \quad \sum_{e \in E} c_e x_e + \sum_{v \in V} c_v^+ x_v^+ + \sum_{v \in V} c_v^- x_v^- \tag{4}$$

$$\text{subject to} \quad x \in X_G, \quad x^+, x^- \in \{0,1\}^V, \tag{5}$$

$$\forall t \in \mathcal{T} \forall v \in V_{t+1} \forall S \in V_t v\text{-cuts}(G_t^+): \quad 1 - x_v^+ \leq \sum_{e \in S} (1 - x_e), \tag{6}$$

$$\forall t \in \mathcal{T} \forall v \in V_t \forall S \in v V_{t+1}\text{-cuts}(G_t^+): \quad 1 - x_v^- \leq \sum_{e \in S} (1 - x_e). \tag{7}$$

The inequalities (6) and (7) are called *birth* and *termination* constraints, respectively.

3 Improved Branch-and-cut Algorithm

Jug et al. propose to solve the MLTP with a branch-and-cut algorithm, for which they design separation procedures

for (1) – (3), (6) – (7) and the bifurcation constraints. In the following, we propose several modifications of the constraints, which improve the performance of the optimization algorithm.

The authors of [20] pointed out that it is sufficient to consider only chordless cycles in (1) and, furthermore, it is well-known that chordless cycle inequalities are facet-defining for multicut polytopes (cf. [15] and [4]). However, this argument was not transferred to inequalities (2) and (3), which may be modified analogously.

Moreover, the inequalities of (3) where $\{v_t, w_t\} \in E_t$ is an edge of the hypothesis graph may be considerably strengthened by a less trivial, yet simple modification. Lemma 2 shows that with both results combined, we can equivalently replace (1) – (3) by the set of tighter inequalities (8) and (9). In relation to our improved version of the branch-and-cut algorithm, we refer to (8) as *cycle* and to (9) as *morality* constraints.

Lemma 2. For every hypothesis graph $G = (V, E)$ it holds that $x \in X'_G$ iff $x \in \{0,1\}^E$ and x satisfies

$$\forall t \in \mathcal{T} \forall \{v, w\} \in E_t \cup E_{t,t+1}$$
$$\forall \text{ chordless } vw\text{-paths } P \text{ in } G_t^+ :$$
$$x_{vw} \leq \sum_{e \in P} x_e \tag{8}$$
$$\forall t \in \mathcal{T} \forall v', w' \in V_t \text{ such that } \{v', w'\} \notin E_t$$
$$\forall v'w'\text{-cuts } S \text{ in } G_t \forall \text{ chordless } v'w'\text{-paths } P \text{ in } G_t^+ :$$
$$1 - \sum_{e \in S}(1 - x_e) \leq \sum_{e \in P} x_e \tag{9}$$

Proof. We first show that any $x \in \{0,1\}^E$ satisfying all of (1) – (3) also satisfies (8) and (9) by contraposition. First, assume $x \in \{0,1\}^E$ violates an inequality of (8) for some $t \in \mathcal{T}$, $\{v, w\} \in E_t \cup E_{t,t+1}$ and chordless vw-path P. We distinguish the following cases: If $\{v, w\} \in E_t$ and P is a path in G_t, then the inequality is included in (1). If $\{v, w\} \in E_{t,t+1}$, then the inequality is included in (2). It remains to consider the case that $\{v, w\} \in E_t$ and P is not entirely contained in G_t. Let $\{v_t, v_{t+1}\}, \{w_t, w_{t+1}\} \in E_{t,t+1}$ with $v_t, w_t \in V_t$ be the first and the last inter frame edges in P, respectively. Furthermore, let $P_{v_{t+1}w_{t+1}}$ be the subpath of P between those edges. Now, either there is a v_tw_t-cut S in G_t such that $x_S = 1$ or there is a v_tw_t-path P' in G_t such that $x_{P'} = 0$. It is clear that P' can be extended to a vw-path of edges labeled 0, because $x_P = 0$. This yields either an inequality of (3) corresponding to $S, \{v_t, v_{t+1}\}, \{w_t, w_{t+1}\}$ and $P_{v_{t+1}w_{t+1}}$ or an inequality of (1) corresponding to $\{v, w\} \cup P'$ that is violated by x.

Next, suppose $x \in \{0,1\}^E$ violates an inequality of (9) for some $t \in \mathcal{T}$, $\{v', w'\} \in E_t$, a $v'w'$-cut S in G_t and a chordless $v'w'$-path P in G_t^+. Then $x_S = 1$ and $x_P = 0$. Clearly, x violates the inequality of (3) corresponding to $S, \{v_t, v_{t+1}\}, \{w_t, w_{t+1}\}$ and $P_{v_{t+1}w_{t+1}}$, where $\{v_t, v_{t+1}\}, \{w_t, w_{t+1}\}$ and $P_{v_{t+1}w_{t+1}}$ are defined similar to the last paragraph.

For the converse, we show that if $x \in \{0,1\}^E$ satisfies the inequalities (8) and (9), then it also satisfies (1) – (3). Any cycle in G_t^+ which is not chordless can be split into two cycles contained in G_t, G_t^+ or G_{t+1} which share exactly one edge. Therefore, any inequality of (1) – (2) is implied by a combination of inequalities from (8). This is a standard argument for multicut polytopes, cf., for instance, [4]. Moreover, for any $\{v_t, w_t\} \in E_t$ and any v_tw_t-cut S in G_t it holds that $\{v_t, w_t\} \in S$. Thus, reapplying the previous argument and the simple fact that

$$1 - \sum_{e \in S}(1 - x_e) \leq 1 - (1 - x_{v_tw_t}) = x_{v_tw_t},$$

we conclude that the inequalities (3) are implied by a combination of inequalities from (8) and (9). □

Remark. Suppose we introduce for every pair of non-neighboring nodes $v', w' \in V_t$ a variable $x_{v'w'}$ indicating whether v' and w' belong to the same cell ($x_{v'w'} = 0$) or not ($x_{v'w'} = 1$). Then any inequality of (9) is exactly the combination of a *cut* inequality $1 - x_{v'w'} \leq \sum_{e \in S}(1 - x_e)$ and a *path* inequality $x_{v'w'} \leq \sum_{e \in P} x_e$ in the sense of *lifted multicuts* ([4]). For neighboring nodes $v, w \in V_t$, i.e. $\{v, w\} \in E_t$, we have the variable x_{vw} at hand and can thus omit the cut part of the morality constraint, as the lemma shows.

3.1 Termination and Birth Constraints

We further suggest a strengthening of the birth and termination constraints in the MLTP. To this end, for any $v \in V_{t+1}$ let $V_t(v) = \{u \in V_t \mid \{u, v\} \in E_{t,t+1}\}$ be the set of neighboring nodes in frame t. Further, we denote by $E\big(V_t(v), V_{t+1} \setminus \{v\}\big)$ the set of inter frame edges that connect some node $u_t \in V_t(v)$ with some node $u_{t+1} \in V_{t+1}$ different from v.

Lemma 3. For every hypothesis graph $G = (V, E)$, the vectors $x \in X'_G$, $x^+, x^- \in \{0,1\}^V$ satisfy inequalities (6) iff the following inequalities hold:

$$\forall t \in \mathcal{T} \forall v \in V_{t+1} \forall S \in V_t v\text{-cuts}(G_t^+) :$$
$$1 - x_v^+ \leq \sum_{e \in S \setminus E(V_t(v), V_{t+1} \setminus \{v\})} (1 - x_e). \tag{10}$$

Similarly, $x \in X'_G$, $x^+, x^- \in \{0,1\}^V$ satisfy (7) iff

$$\forall t \in \mathcal{T} \forall v \in V_t \forall S \in vV_{t+1}\text{-cuts}(G_t^+) :$$
$$1 - x_v^- \leq \sum_{e \in S \setminus E(V_t \setminus \{v\}, V_{t+1}(v))} (1 - x_e) \tag{11}$$

hold true.

Proof. We show the claim only for birth constraints since the proof for termination constraints is analogous. Let $x \in X'_G$ and $x^+, x^- \in \{0,1\}^V$. Apparently, if (10) is satisfied, then

$$\sum_{e \in S \setminus E(V_t \setminus \{v\}, V_{t+1}(v))} (1 - x_e) \leq \sum_{e \in S} (1 - x_e)$$

implies that (6) also holds. Conversely, suppose (10) is violated. Then there exists some $t \in \mathcal{T}$ and $v \in V_{t+1}$,

$S \in V_t v$-cuts(G_t^+) such that $x_v^+ = 0$ and $x_e = 1$ for all $e \in S \setminus E\big(V_t(v), V_{t+1} \setminus \{v\}\big)$. Assume (6) is not violated, then there is a path P in G_t^+ from some node in V_t to v with $x_P = 0$. Then P must have non-empty intersection with $E\big(V_t(v), V_{t+1} \setminus \{v\}\big)$. Let $u \in V_t(v)$ and $v' \in V_{t+1} \setminus \{v\}$ be such that $\{u, v'\} \in P$. Since $x_{uv} = 1$ it follows that x violates the inequality

$$x_{uv} \leq \sum_{e \in P_{uv}} x_e$$

of (2) where P_{uv} is the subpath of P from u to v. This is a contradiction to $x \in X'_G$. □

3.2 Additional Odd Wheel Constraints

A *wheel* $W = (V(W), E(W))$ is a graph that consists of a cycle and a dedicated center node $w \in V(W)$ which is connected by an edge to every node in the cycle. Let E_C denote the edges of W in the cycle and E_w the remaining center edges. With a wheel subgraph $W = (V(W), E(W))$ of a graph G we may associate an inequality

$$\sum_{e \in E_C} x_e - \sum_{e \in E_w} x_e \leq \left\lfloor \frac{|V(W)| - 1}{2} \right\rfloor, \qquad (12)$$

which is valid for multicut polytopes ([15]). A wheel is called *odd* if $|V(W)| - 1$ is odd. It is known that wheel inequalities are facet-defining for multicut polytopes iff the associated wheel is odd ([15]).

We propose to add additional odd wheel inequalities to the MLTP in order to strengthen the corresponding LP relaxation. More precisely, we consider only wheels $W = (V(W), E(W)) \subset G$ such that $w \in V_{t+1}$ and $v \in V_t$ for all $v \in V(W) \setminus w$ and some $t \in \mathcal{T}$. This structure guarantees that for any $x \in X'_G$, the restriction $x_{E(W)}$ is the incidence vector of a multicut of W. Therefore, (12) holds with respect to x.

3.3 Optimization Procedure

For a subset of the constraints, we use the commercial branch-and-cut solver Gurobi [19] to solve the LP relaxation and find integer feasible solutions. Whenever Gurobi finds an integer feasible solution x, we check whether $x \in X_G$ and all birth and termination constraints are satisfied. If not, then we provide Gurobi with an additional batch of violated inequalities from (8) – (11) as well as violated bifurcation constraints and repeat. To this end, we adapt the separation procedures of [20] to account for our improvements in a straight-forward manner. We restrict ourselves to optionally add wheel inequalities for odd wheels with 4 nodes as described above (so-called *3-wheels*) to the starting LP relaxation.

4 Local Search Algorithms

In this section, we introduce two local search heuristics for the MLTP. The first is a greedy agglomeration heuristic that is based on GAEC [27], which constructs a lineage in a bottom-up fashion. The second is a Kernighan-Lin-type algorithm [26] and an extension of KLj [27] (a heuristic for the MCMCP), which tries to improve a feasible lineage by greedily moving vertices between components. We adapt both to account for birth and termination costs, and to treat temporal edges differently than spatial edges, thus allowing a cell to be linked to two different cells in the successive time point.

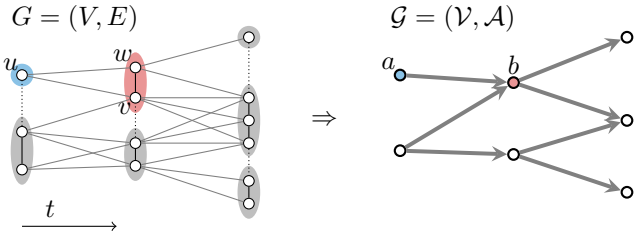

Figure 2 For a fixed decomposition of the frames (depicted with black solid/dashed cut edges), we associate a directed graph $\mathcal{G}$ over the components $\mathcal{V}$. The arcs $\mathcal{A}$ bundle all edges going from any node of one cell to any node of another cell in the successive frame. For example, the components $V_a = \{u\}$ and $V_b = \{v, w\}$ are linked by the arc ab which corresponds to the set of edges $E_{ab} = \{uv, uw\}$. Determining the optimal state of the temporal edges (grey) given a decomposition into cells boils down to finding an optimal branching in $\mathcal{G}$.

Both algorithms maintain a decomposition of the graph $(V, \bigcup_{t \in \mathcal{T}} E_t)$, *ie.* the components within each frame G_t represent the cells. Let $\mathcal{V}$ be the set of all cells. For each set of edges going from a component $a \in \mathcal{V}$ at time point t to a component b at $t + 1$, we associate an arc $ab \in \mathcal{A}$. This gives a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, as illustrated in Fig. 2. We write V_a for the set of vertices v in component $a \in \mathcal{V}$ and E_{ab} for the set of edges represented by arc $ab \in \mathcal{A}$. They further maintain a selection of the arcs $\mathcal{A}(y)$, where $y \in \{0, 1\}^{\mathcal{A}}$, to represent which temporal edges are cut.

4.1 Greedy Lineage Agglomeration (GLA)

This algorithm, shown in Alg. 1, takes an MLTP instance and constructs a feasible lineage in a bottom-up fashion. Starting from $\mathcal{V} = V$, it builds the lineage forest by either merging two components within an identical frame or selecting an arc $ab \in \mathcal{A}$. The final selection of arcs then determines which temporal edges are cut edges ($x_e = 1$). To do so, it maintains contracted edges between components within the same frame, denoted with $\mathcal{E}$, along with $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. The change in objective (4) caused by a particular move involving a and b is denoted with $\Delta_{ab}^{\text{move}}$. The three allowed moves, merge, setParent and changeParent, are depicted in Fig. 3. In order to determine the cost or reward of a particular move, we have to examine not only the edge between the involved components a and b, but also whether they have an associated parent or child cell already. For a merge, we have to consider arcs going to children or parents of either component, since they would be combined into an active arc and therefore change their state and affect the ob-

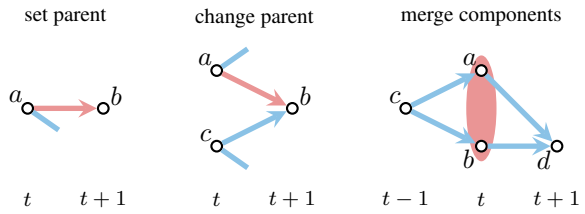

Figure 3 The three moves of GLA: set a as parent of b (**left**), change the parent of b from c to a (**middle**) or merge two components a and b into one (**right**). The major arc along which the move occurs is depicted in red, while arcs that affect the cost of the move are depicted in blue. When changing a parent, for example, the presence of other active arcs originating from a and c determine whether termination costs have to be paid. For a merge, we have to consider arcs to parents or children, which would be joined with an active arc and therefore change their state.

Algorithm 1 Greedy Lineage Agglomeration (GLA)

while progress **do**
  $(a, b) \leftarrow \arg\min_{ab \in \mathcal{E} \cup \mathcal{A}} \Delta^{\text{move}}_{ab}$
  if $\Delta^{\text{move}}_{ab} < 0$ **then**
    $\text{applyMove}(\mathcal{G}, a, b)$ ▷ updates partitions of $\mathcal{G}$ and selects arcs $\mathcal{A}(y)$.
  else
    break
  end if
end while
return $\text{edgeLabels}(\mathcal{G})$ ▷ cut-edge labeling x^* from $\mathcal{V}$ and $\mathcal{A}(y)$.

jective. The detailed, incremental calculation of these move costs $\Delta^{\text{move}}_{ab}$ can be found in the appendix. We maintain feasibility at all times: two components with different parents cannot be merged (it would violate morality constraints (3)), and similarly, a merge of two partitions with a total of more than two active outgoing arcs is not considered (as it would violate bifurcation constraints). The algorithm stops as soon as no available move decreases the objective.

Implementation. We use a priority queue to efficiently retrieve the currently best move. After a move is applied, all affected moves are re-calculated and inserted into the queue. We invalidate previous editions of moves indirectly by keeping track of the most recent version for all $\mathcal{E}$. For each component, we actively maintain the number of children and its parent to represent the selected arcs $\mathcal{A}(y)$.

4.2 Kernighan-Lin with Optimal Branchings (KLB)

Algorithm 2 takes an MLTP instance and an initial decomposition, *eg.* the result of GLA, and attempts to decrease the objective function (4) in each step by changing the in-frame partitions in a KL-fashion. Like the algorithm proposed by [27] for the related MCMCP, it explores three different local moves to decrease the objective function maximally: **a)** apply a sequence of k node switches between two adjacent components a and b, **b)** a complete merge of two components, and **c)** splitting a component into two.

Algorithm 2 KL with Optimal Branchings (KLB)

while progress **do**
  for $a, b \in \mathcal{V}$ **do**
    if $\nexists uv \in E_t : u \in V_a \wedge v \in V_b$ **then**
      continue
    end if
    $\text{improveBipartition}(\mathcal{G}, a, b)$ ▷ move nodes across border or merge.
  end for
  for $a \in \mathcal{V}$ **do**
    $\text{splitPartition}(\mathcal{G}, a)$ ▷ split partition.
  end for
end while
return $\text{cutEdgeLabels}(\mathcal{G})$ ▷ cut-edge labeling x^* from $\mathcal{V}$ and $\mathcal{A}(y^*)$.

Operations that do not decrease the objective will be discarded. In contrast to the setting of a MCMCP, such local modifications affect the optimal parent-child relations between the (changed) components and thus, it becomes difficult to determine by how much the total objective (ideally) changes. To this end, we note that for a fixed intra-frame decomposition, determining the state of the temporal edges becomes a minimum cost branching problem (MCBP) on $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. Therefore, both $\text{improveBipartition}(\dots)$ and $\text{splitPartition}(\dots)$ repeatedly solve an MCBP to determine the objective after an elementary move.

Minimum Cost Branching on $\mathcal{G}$. For $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ over a fixed decomposition into cells $\mathcal{V}$, we formulate the MCBP with birth and termination costs and bifurcation constraint as an ILP:

$$\min_{y, y^-, y^+} \quad \sum_{ab \in \mathcal{A}} c_{ab} y_{ab} + \sum_{a \in \mathcal{V}} c^+_a y^+_a + \sum_{a \in \mathcal{V}} c^-_a y^-_a \tag{13}$$

$$\text{subject to} \quad \forall a \in \mathcal{V} \;:\; (1 - y^+_a) = \sum_{b \in \delta^-(a)} y_{ba} \tag{14}$$

$$\forall a \in \mathcal{V} \;:\; (1 - y^-_a) \leq \sum_{b \in \delta^+(a)} y_{ab} \leq 2 \tag{15}$$

$$y \in \{0,1\}^{\mathcal{A}}, \quad y^-, y^+ \in \{0,1\}^{\mathcal{V}} \;, \tag{16}$$

where y_{ab} indicates whether arc ab is active ($y_{ab} = 1$) or not ($y_{ab} = 0$). The equality constraint (14) ensures that at most one incoming arc is selected (preventing a violation of morality) and, if none is chosen, the birth indicator y^+_a is active. In the same sense, (15) enforces the penalty for termination if necessary, and its upper bound limits the number of children to 2, which enforces the bifurcation constraint. Note that $\mathcal{G}$ is acyclic by construction, and we therefore do not require cycle elimination constraints. Observing that $\forall e \in E_{ab} : 1 - y_{ab} = x_e$, *ie.* all edges in an arc need to have the same state to satisfy space-time constraints, we derive the weights $c_{ab} = -\sum_{e \in E_{ab}} c_e$. With a similar reasoning, all vertices of a component a need to be in the same birth/termination state, $\forall v \in V_a : y^+_a = x^+_v$, hence we derive $c^+_a = \sum_{v \in V_a} c^+_v$ (and analogous for termination costs

c_a^-). The MLTP objective f_{MLTP} (4) can be split into two parts and directly calculated from the decomposition into cells and the temporal cut edges induced from y^*, the optimal branching:

$$f_{\text{MLTP}}(x^*) = \sum_{e \in \bigcup_{t \in \mathcal{T}} E_t} c_e x_e^* \; + \sum_{e \in \bigcup_{t \in \mathcal{T}} E_{t,t+1}} c_e + f_{\text{MCBP}}(y^*) \;\; , \quad (17)$$

where we identify the first two sums as the costs for the fixed decomposition with all temporal edges cut, and $f_{\text{MCBP}}(y^*)$ corresponds to the achieved branching objective (13). The derivation is found in the supplements. This allows us to evaluate the change of objective of a move simply as the sum of partial changes from both decomposition and branching. For the latter, we note that for a given $\mathcal{V}$, modifying two of its cells a and b in frame t will have relatively localized effects on y^*. In particular, it only affects arcs that go from $t-1$ to t and from t to $t+1$, (their choice is independent from selected arcs in earlier or later frames when conditioned on $\mathcal{V}_{t-1}$ and $\mathcal{V}_{t+1}$). In practice, we find that the effect is often also spatially localized, hence we optionally restrict ourselves to only update the MCBP in a range of d_{MCB} (undirected) arc hops of from a and b. This parameter is instance dependent.

Implementation. The algorithm maintains the weighted $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, the objective corresponding to the intra-frame decomposition, *ie.* the first part of (17), and solves the MCBP on $\mathcal{G}$ by the branch-and-cut algorithm implemented in [19]. In order to reduce the number of overall calculations in later iterations, we mark components that have changed and then, in the next iteration, attempt to improve only those pairs of components which involve at least one changed component. To account for changes that affect moves in previous or subsequent frames, we propagate these "changed" flags to all potential parents or children of a changed component.

5 Experiments & Results

Instances and Setup. We evaluate our algorithms on the two large instances of [20]: *Flywing-epithelium* and *N2DL-HeLa-full*. The hypothesis graph of the former instance consists of 5026 nodes and 19011 edges, while the latter consists of 10882 nodes and 19807 edges. In addition to this, we report experiments on two more sequences of a flywing epithelium time-lapse microscopy with a wider field of view. The corresponding hypothesis graphs consist of 10641 nodes and 42236 edges, respectively 76747 edges. We denote the data sets with *Flywing-wide I* and *II*. These instances are preprocessed with the same pipeline as *Flywing-epithelium*. For details on the preprocessing, we refer to [20].

Our choice of birth and termination costs follows [20], *ie.* we set $c^+ = c^- = 5$ for all instances. We initialize the KLB heuristic with the solution of GLA to decrease the number of outer iterations. We benchmark three versions of KLB: The first one solves the full MCBP in each turn and is denoted with KLB. The second, KLB-d=inf, exploits the described temporal locality and solves the MCBP within the (reachable) subgraph of $t \pm 1$, while the third, KLB-d=10, additionally exploits spatial locality ($d_{\text{MCB}} = 10$).

We apply our improved branch-and-cut algorithm with and without additional 3-wheel constraints in order to evaluate the effect of the added inequalities. For the more extensive data sets *Flywing-wide I* and *II* we provide Gurobi with the objective value of the (quickly available) GLA solution to avoid separating poor integral points and accelerate fathoming within the branch-and-bound tree.

Convergence Analysis. The convergence of our algorithms in comparison to the branch-and-cut algorithm of [20] is reported in Fig. 4 and Table 1. We find that GLA is the fastest in all instances, but only reaches a local optimum with a gap of about $1.95\,\%$ and $3.69\,\%$, respectively. This solution is improved by KLB in terms of objective, up to a gap of $0.76\,\%$ and $1.86\,\%$. All three variants of KLB obtain the same solution in terms of cut-edge labeling, but those which exploit temporal locality, *ie.* KLB-d=inf and KLB-d=10, do so considerably faster. Considering spatial locality, on the other hand, does not improve runtime on these two instances. We find that KLB spends most of the time in the first outer iteration, where it has to check a large number of bipartitions that do not improve and will therefore not be considered in the next iteration. Our implementation of KLB currently only uses a single core, but could potentially be sped up by parallelization.

The improved branch-and-cut algorithm, denoted with tight-ILP and tight-ILP+3W (with additional 3-wheel constraints), retrieves feasible solutions considerably faster than the algorithm of [20] (original-ILP). For *Flywing-epithelium*, they reach the optimal solution in $74.5\,s$ and $71\,s$, respectively, while the original algorithm took $4098\,s$. On *N2DL-HeLa*, the original-ILP found the optimal solution after $83929\,s$, and tight-ILP or tight-ILP+3W found it in $610\,s$ and $827.5\,s$. In both cases, they provide tighter bounds, and the variant with 3-wheel constraints is even able to prove optimality of the solutions found for both instances. On a separate experiment with disabled bifurcation constraints, we found that tight-ILP+3W was able to prove optimality on *Flywing-epithelium* within $296\,s$ and its solution does not violate any bifurcation constraint. As shown in Fig. 5, we observe that our modifications of the branch-and-cut greatly reduce the number of morality cuts.

On the larger instances *Flywing-wide I* and *II*, we present our results in Fig. 6. The branch-and-cut algorithms did not determine a feasible solution, but provide lower bounds. For the variant with 3-wheel constraints, these bounds are slightly tighter in both instances. Consequently, we are able to determine the maximal optimality gaps for GLA to be $2.9\,\%$ (*I*) and $2.1\,\%$ (*II*), and $1.3\,\%$ (*I*) and $0.9\,\%$ (*II*) for KLB. Again, all variants of KLB obtain identical solutions. On these instances, not only temporal locality helps to speed up the algorithm, but also spatial locality: KLB-d=inf reduces runtime from $9154.61\,s$ to $1781.48\,s$ (*I*) and from $28625.40\,s$ to $10388.60\,s$ (*II*), while KLB-d=10 re-

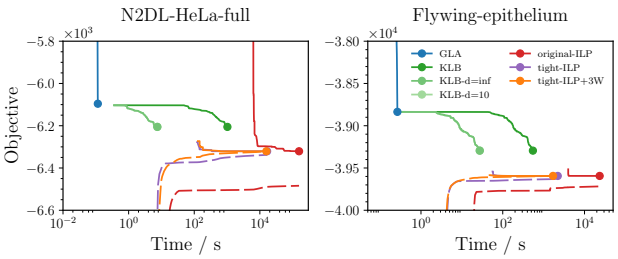

Figure 4 Comparison of algorithms for the MLTP in terms of runtime, objective (solid) and bounds (dashed) on the large instances of [20]. Our heuristics are able to determine feasible solutions quickly, while our branch-and-cut algorithms (tight-ILP, tight-ILP+3W) converge to the optimal solution in up to one hundredth of the time of the original branch-and-cut algorithm (original-ILP) and provide tighter bounds.

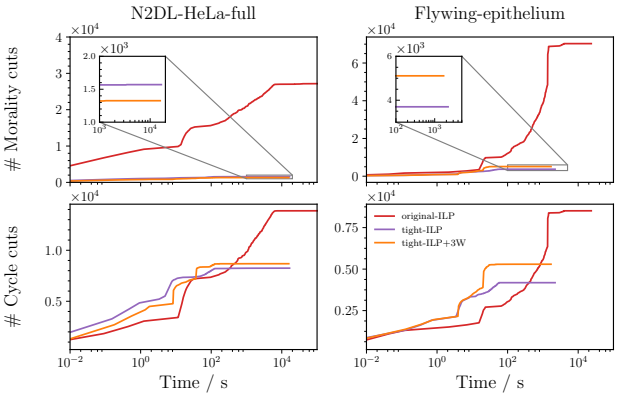

Figure 5 Number of morality cuts (**top**), *ie.* (3) or (9), and cycle cuts (**bottom**), *ie.* (1) and (2) or (8), separated in the different branch-and-cut algorithms. We observe that our algorithms, tight-ILP and tight-ILP+3W, require considerably fewer morality cuts, while the number of cycle cuts (including both space-cycles and space-time-cycles) is in the same order of magnitude.

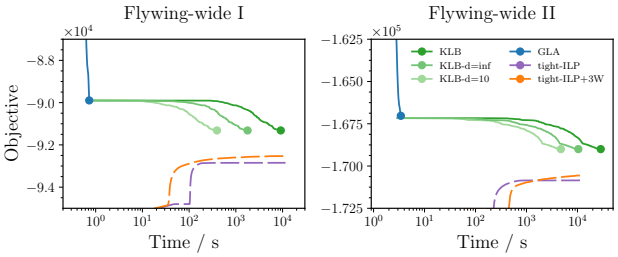

Figure 6 Results on the more extensive instances *Flywing-wide I* and *II*. The branch-and-cut algorithm with 3-wheel constraints provides slightly tighter bounds, with which we determine the gaps for GLA to be 2.9 % (*I*) and 2.1 % (*II*), and 1.3 % (*I*) and 0.9 % (*II*) for KLB. Exploiting temporal and spatial locality when re-solving the MCBP considerably reduces runtime of KLB.

duces this further to 394.20 s and 4724.94 s.

Solution Quality. We compare the solution quality of our two heuristics by segmentation (SEG) and tracking (TRA) metrics as used in [30] for *Flywing-epithelium*. The results

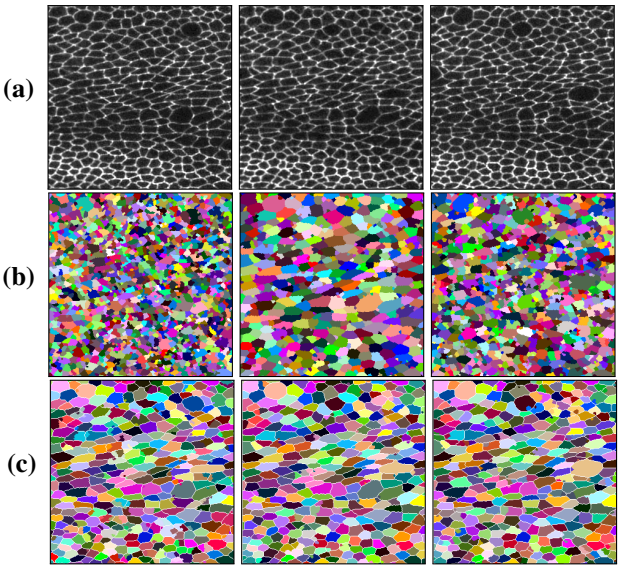

Figure 7 Depicted above are **(a)** three representative and consecutive time points of *Flywing-wide I*, **(b)** corresponding decompositions of these images into cell fragments, and **(c)** decompositions of the images defined by a feasible solution of the moral lineage tracing problem. The white edges in **(c)** are the outlines of gold standard human lineage annotation, added here for simplified comparison.

are reported in Table 2. We observe that KLB improves the scores of GLA slightly (up to an additional 1.2 % and 0.81 % for SEG and TRA, respectively). The (optimal) solutions of the original and the improved branch-and-cut algorithm are identical, and achieve slightly better scores in both measures than the heuristics. All presented algorithms outperform the baseline, the *packing analyzer* [1], a typical cell tracking tool used by biologists whose scores were originally reported in [20].

6 Conclusion

In this work, we addressed efficient algorithms for the recently introduced MLTP [20], a mathematical framework for cell lineage reconstruction, which treats both subproblems, image decomposition and tracking, jointly. We have improved the branch-and-cut algorithm of [20] by separating tighter cutting planes. Furthermore, we proposed two heuristics for MLTP: a fast agglomerative procedure called GLA that constructs a feasible lineage in a bottom-up fashion, and a variant of the KL-algorithm which attempts to improve a given lineage by switching nodes between components, merging or splitting them, and repeatedly solves a MCBP conditioned on fixed partitions. Our algorithms find exact solutions for previous instances up to two orders of magnitude faster and produce near-optimal solutions for wider instances in reasonable time. This empirically demonstrates that our methods alleviate runtime issues with MLTP instances and improve the applicability of *moral lineage tracing* in practice.

Table 1 Detailed quantitative comparison of algorithms for the MLTP. BestGap is calculated using the tightest bound of any algorithm, while Gap is based on the bound established by each particular algorithm. We limit the runtime of tight-ILP to approximately the time it took tight-ILP+3W to prove optimality. KLB solves the full branching problem in every turn, while KLB-d=inf utilizes temporal locality and KLB-d=10 additionally uses spatial locality with $d_{\mathrm{MCBP}} = 10$.

	Flywing-epithelium					N2DL-HeLa-full				
Method	Time / s	objBest	objBound	Gap	BestGap	Time / s	objBest	objBound	Gap	BestGap
GLA	0.26	-38835.90			0.0195	0.12	-6095.85			0.0369
KLB	546.32	-39294.65			0.0076	1025.39	-6205.54			0.0186
KLB-d=inf	27.45	-39294.65			0.0076	7.51	-6205.54			0.0186
KLB-d=10	27.28	-39294.65			0.0076	7.50	-6205.54			0.0186
tight-ILP	2214.06	-39593.90	-39630.50	0.0009	0.0000	16627.30	-6320.81	-6337.98	0.0027	0.0000
tight-ILP+3W	1696.80	-39593.90	-39593.90	0.0000	0.0000	15780.50	-6320.81	-6320.81	0.0000	0.0000
original-ILP [20]	23460.80	-39593.90	-39717.80	0.0031	0.0000	156542.00	-6320.81	-6484.02	0.0258	0.0000

Table 2 Comparison of the distance from ground truth of segmentation (SEG) and traced lineage forest (TRA) on *Flywing-epithelium*. ILP denotes the result of the branch-and-cut algorithm, while PA [1] is a common tracking tool used by biologists.

Algorithm	SEG	TRA
GLA	0.9363	0.9640
KLB	0.9485	0.9721
ILP	0.9722	0.9813
PA (auto)	0.7980	0.9206

References

[1] B. Aigouy, R. Farhadifar, D. B. Staple, A. Sagner, J.-C. Röper, F. Jülicher, and S. Eaton. Cell flow reorients the axis of planar polarity in the wing epithelium of drosophila. *Cell*, 142(5):773–786, 2010. 7, 8

[2] F. Amat, W. Lemon, D. P. Mossing, K. McDole, Y. Wan, K. Branson, E. W. Myers, and P. J. Keller. Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nature methods*, 2014. 1

[3] F. Amat, E. W. Myers, and P. J. Keller. Fast and robust optical flow for time-lapse microscopy using super-voxels. *Bioinformatics*, 29(3):373–380, 2013. 1

[4] B. Andres, A. Hornakova, and J.-H. Lange. Lifting of multicuts. *CoRR*, abs/1503.03791, 2015. 3

[5] B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *ICCV*, 2011. 1

[6] B. Andres, T. Kröger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Köthe, and F. A. Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *ECCV*, 2012. 1

[7] B. Andres, J. Yarkony, B. S. Manjunath, S. Kirchhoff, E. Türetken, C. C. Fowlkes, and H. Pfister. Segmenting planar superpixel adjacency graphs w.r.t. non-planar superpixel affinity graphs. In *EMMCVPR*, 2013. 1

[8] S. Bagon and M. Galun. Large scale correlation clustering optimization. *CoRR*, abs/1112.2903, 2011. 1

[9] T. Beier, B. Andres, U. Köthe, and F. A. Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. In *ECCV*, 2016. 1

[10] T. Beier, F. A. Hamprecht, and J. H. Kappes. Fusion moves for correlation clustering. In *CVPR*, pages 3507–3516, 2015. 1

[11] T. Beier, T. Kroeger, J. H. Kappes, U. Köthe, and F. A. Hamprecht. Cut, glue, & cut: A fast, approximate solver for multicut partitioning. In *CVPR*, pages 73–80, 2014. 1

[12] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011. 1

[13] B.-C. Chen, W. R. Legant, K. Wang, L. Shao, D. E. Milkie, M. W. Davidson, C. Janetopoulos, X. S. Wu, J. A. Hammer, Z. Liu, et al. Lattice light-sheet microscopy: Imaging molecules to embryos at high spatiotemporal resolution. *Science*, 346(6208):1257998, 2014. 1

[14] N. Chenouard, I. Smal, F. De Chaumont, M. Maska, I. F. Sbalzarini, Y. Gon, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. Magnusson, J. Jalden, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. Van Wezel, H.-W. Dan, Y.-S. Tsai, C. Ortiz De Solorzano, J.-C. Olivo-Marin, and E. Meijering. Objective comparison of particle tracking methods. *Nature Methods*, 11(3):281–289, 2014. 1

[15] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1):87–115, 1993. 2, 3, 4

[16] J. Funke, B. Andres, F. A. Hamprecht, A. Cardona, and M. Cook. Efficient automatic 3d-reconstruction of branching neurons from em data. In *CVPR*, pages 1004–1011. IEEE, 2012. 1

[17] A. Greenbaum, W. Luo, T.-W. Su, Z. Göröcs, L. Xue, S. O. Isikman, A. F. Coskun, O. Mudanyali, and A. Ozcan. Imaging without lenses: achievements and remaining challenges of wide-field on-chip microscopy. *Nature methods*, 9(9):889–895, 2012. 1

[18] C. Guillot and T. Lecuit. Mechanics of epithelial tissue homeostasis and morphogenesis. *Science*, 340(6137):1185–1189, 2013. 1

[19] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016. 4, 6

[20] F. Jug, E. Levinkov, C. Blasse, E. W. Myers, and B. Andres. Moral lineage tracing. In *CVPR*, 2016. 1, 2, 3, 4, 6, 7, 8

[21] F. Jug, T. Pietzsch, D. Kainmüller, J. Funke, M. Kaiser, E. van Nimwegen, C. Rother, and G. Myers. Optimal joint segmentation and tracking of escherichia coli in the mother

machine. In *Bayesian and grAphical Models for Biomedical Imaging*, pages 25–36. Springer, 2014. 1

[22] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnörr. Higher-order segmentation via multicuts. *Computer Vision and Image Understanding*, 143:104–119, 2016. 1

[23] B. X. Kausler, M. Schiegg, B. Andres, M. Lindner, U. Koethe, H. Leitte, J. Wittbrodt, L. Hufnagel, and F. A. Hamprecht. A discrete chain graph model for 3d+ t cell tracking with high misdetection robustness. In *ECCV*, pages 144–157. Springer, 2012. 1

[24] P. J. Keller, A. D. Schmidt, A. Santella, K. Khairy, Z. Bao, J. Wittbrodt, and E. H. Stelzer. Fast, high-contrast imaging of animal development with scanned light sheet-based structured-illumination microscopy. *Nature methods*, 7(8):637–642, 2010. 1

[25] P. J. Keller, A. D. Schmidt, J. Wittbrodt, and E. H. Stelzer. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322(5904):1065–1069, 2008. 1

[26] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970. 4

[27] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *ICCV*, pages 1751–1759, 2015. 1, 2, 4, 5

[28] S. Kim, C. D. Yoo, S. Nowozin, and P. Kohli. Image segmentation usinghigher-order correlation clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1761–1774, 2014. 1

[29] K. Li, E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, and P. G. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical image analysis*, 12(5):546–566, 2008. 1

[30] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolcková, M. Streitová, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G. Magnusson, J. Jaldn, H. M. Blau, O. Dzyubachyk, P. Křížek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Muñoz-Barrutia, E. Meijering, M. Kozubek, and C. Ortiz-de Solorzano. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014. 1, 7

[31] E. Meijering, O. Dzyubachyk, and I. Smal. Methods for cell and particle tracking. *Methods in Enzymology*, 504(9):183–200, 2012. 1

[32] D. Padfield, J. Rittscher, and B. Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical image analysis*, 15(4):650–668, 2011. 1

[33] M. Rempfler, B. Andres, and B. Menze. The minimum cost connected subgraph problem in medical image analysis. In *MICCAI*, 2016. 1

[34] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F. A. Hamprecht. Graphical model for joint segmentation and tracking of multiple dividing cells. *Bioinformatics*, 31:948–956, 2015. 1

[35] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht. Conservation tracking. In *ICCV*, pages 2928–2935, 2013. 1

[36] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph decomposition for multi-target tracking. In *CVPR*, pages 5033–5041, 2015. 1

[37] R. Tomer, K. Khairy, F. Amat, and P. J. Keller. Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. *Nature methods*, 9(7):755–763, 2012. 1

[38] E. Türetken, F. Benmansour, B. Andres, P. Glowacki, H. Pfister, and P. Fua. Reconstructing curvilinear networks using path classifiers and integer programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2515–2530, 2016. 1

[39] E. Türetken, G. González, C. Blum, and P. Fua. Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9(2-3):279–302, 2011. 1

[40] X. Wang, E. Türetken, F. Fleuret, and P. Fua. Tracking interacting objects optimally using integer programming. In *ECCV*, pages 17–32. Springer, 2014. 1

[41] J. Yarkony, A. Ihler, and C. C. Fowlkes. Fast planar correlation clustering for image segmentation. In *ECCV*, pages 568–581, 2012. 1

[42] J. E. Yarkony and C. Fowlkes. Planar ultrametrics for image segmentation. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *NIPS*, pages 64–72. Curran Associates, Inc., 2015. 1

[43] I. K. Zervantonakis, S. K. Hughes-Alford, J. L. Charest, J. S. Condeelis, F. B. Gertler, and R. D. Kamm. Three-dimensional microfluidic model for tumor cell intravasation and endothelial barrier function. *Proceedings of the National Academy of Sciences*, 109(34):13515–13520, 2012. 1