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. [21] have proposed a mathematical abstraction of this task, the moral lineage tracing problem (MLTP), whose feasible solutions define both a segmentation of every image and a lineage forest of cells. Their branch-and-cut algorithm, however, is prone to many cuts and slow convergence for large instances. To address this problem, we make three contributions: (i) we devise the first efficient primal feasible local search algorithms for the MLTP, (ii) we improve the branch-and-cut algorithm by separating tighter cutting planes and by incorporating our primal algorithms, (iii) we show in experiments that our algorithms find accurate solutions on the problem instances of Jug et al. and scale to larger instances, leveraging moral lineage tracing to practical significance.

1. Introduction

Recent advances in microscopy have enabled biologists to observe organisms on a cellular level with higher spatio-temporal resolution than before [12, 16, 44]. Analysis of such microscopy sequences is key to several open questions in biology, including embryonic development of complex organisms [25, 26], tissue formation [17] or the understanding of metastatic behavior of tumor cells [50]. However, to get from a sequence 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 [2, 3, 13, 32, 33, 34].

Cell lineage tracing is typically considered a two step problem: In the first step, individual cells are detected and segmented in every image. Then, in the second step, individual cells are tracked over time and, in case of a cell division, linked to their ancestor cell, to finally arrive at the lineage forest of all cells (Fig. 1). The tracking 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. 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 [11, 42, 47, 20, 43] or reconstruction of tree-like structures [15, 39, 37, 45, 46]. It has been cast in the form of different optimization problems [22, 24, 36, 40, 41] that can deal with some of the mentioned difficulties, e.g., by selecting from multiple segmentation hypotheses [40, 41].

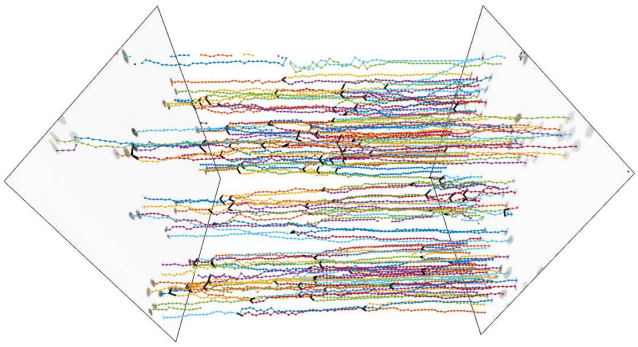

Figure 1. Depicted above is a lineage forest of cells from a sequence of microscopy images. The first image of the sequence is shown on the left. The last image is shown on the right. Cell divisions are depicted in black.

Jug et al. [21], on the other hand, have proposed a rigorous mathematical abstraction of the joint problem which they call the *moral lineage tracing problem (*MLTP*)*. It is a hybrid of the *minimum cost multicut problem (*MCMCP*)*, which has been studied extensively for image decompo-

*Authors contributed equally.

sition [4, 5, 6, 7, 8, 9, 10, 23, 28, 29, 48, 49], and the *minimum cost disjoint arborescence* problem, variations of which have been applied to reconstruct lineage forests in [22, 24, 36, 41, 40] or tree-like structures [15, 46, 45]. 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 (*c.f*. Fig. 2). Solving this optimization problem therefore tackles both subtasks – segmentation and tracking – simultaneously. While Jug et al. [21] demonstrate the advantages of their approach in terms of robustness, they also observe that their branch-and-cut algorithm (as well as the cutting-plane algorithm for the linear relaxation they study) is prone to a large number of cuts and exhibits slow convergence on large instances. That, unfortunately, prevents many applications of the MLTP in practice, since it would be too computationally expensive.

Contributions. In this paper, we make three contributions: Firstly, we devise two efficient heuristics for the MLTP, both of which are primal feasible local search algorithms inspired by the heuristics of [28, 31] for the MCMCP. We show that for fixed intra-frame decompositions, the resulting subproblem can be solved efficiently via bipartite matching.

Secondly, we improve the branch-and-cut algorithm [21] by separating tighter cutting planes and by employing our heuristics to extract feasible solutions.

Finally, we demonstrate the convergence of our algorithms on the problem instances of [21], solving two (previously unsolved) instances to optimality and obtaining accurate solutions orders of magnitude faster. We demonstrate the scalability of our algorithms on larger (previously inaccessible) instances.

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 [21] 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, *e.g*. 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. 2. 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 $t+1$.

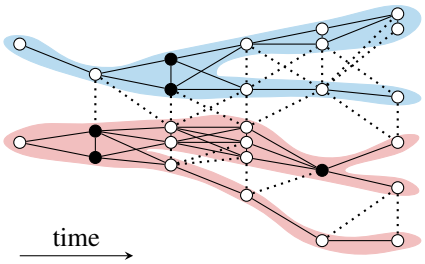

Figure 2. The *moral lineage tracing problem (*MLTP*)*[1]: Given a sequence of images decomposed into cell fragments (depicted as nodes in the figure), cluster fragments into cells in each frame and *simultaneously* associate cells into lineage forests over time. Solid edges indicate joint cells within images and descendant relations across images. Black nodes depict fragments of cells about to divide.

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 $(V, E_t \setminus L)$.

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 [21], 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 ([21])**.** For every hypothesis graph $G = (V, E)$ and every $x \in \{0, 1\}^E$, the set $x^{-1}(1)$ of edges labeled 1 is

[1] The figure is a correction of the one displayed in [21].

[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. [19].

a lineage cut of G iff x satisfies inequalities (1) – (3):

$$\forall t \in \mathcal{T} \forall C \in \text{cycles}(G_t) \forall e \in C:$$
$$x_e \leq \sum_{e' \in C \setminus \{e\}} x_{e'} \tag{1}$$
$$\forall t \in \mathcal{T} \forall \{v, w\} \in E_{t,t+1} \forall P \in vw\text{-paths}(G_t^+):$$
$$x_{vw} \leq \sum_{e \in P} x_e \tag{2}$$

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

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 [21, 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, [21] defines the following *moral lineage tracing problem* (MLTP)

$$\min_{x, x^+, x^-} \quad \sum_{e \in E} c_e x_e + \sum_{v \in V} c_v^+ x_v^+ + \sum_{v \in V} c_v^- x_v^- \tag{4}$$
$$\text{subject to} \quad x \in X_G, \quad x^+, x^- \in \{0,1\}^V, \tag{5}$$
$$\forall t \in \mathcal{T} \forall v \in V_{t+1} \forall S \in V_t v\text{-cuts}(G_t^+):$$
$$1 - x_v^+ \leq \sum_{e \in S} (1 - x_e), \tag{6}$$
$$\forall t \in \mathcal{T} \forall v \in V_t \forall S \in v V_{t+1}\text{-cuts}(G_t^+):$$
$$1 - x_v^- \leq \sum_{e \in S} (1 - x_e). \tag{7}$$

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

3. Local Search Algorithms

In this section, we introduce two local search heuristics for the MLTP. The first builds a lineage bottom-up in a greedy fashion, while the second applies Kernighan-Lin [27] updates to the intra-frame components. The latter requires repeatedly optimizing a branching problem, given a fixed intra-frame decomposition, for which we discuss an efficient combinatorial minimizer.

Both algorithms maintain a decomposition of the graph $(V, \bigcup_{t \in \mathcal{T}} E_t)$, *i.e.* the components within each frame G_t that represent the cells. We denote the set of all cells with

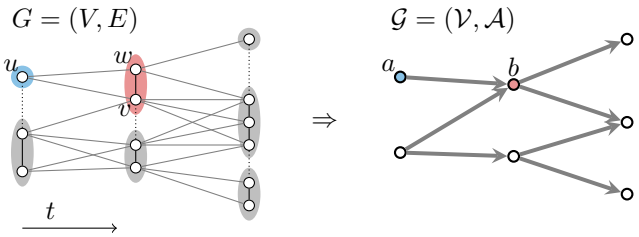

Figure 3. For a fixed decomposition of the frames (depicted with black solid/dashed cut edges), we associate a directed graph $\mathcal{G}$ over the components $\mathcal{V}$. The arcs $\mathcal{A}$ bundle all edges going from any node of one cell to any node of another cell in the successive frame. For 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 $\mathcal{G}$.

Algorithm 1 Greedy Lineage Agglomeration (GLA)

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

$\mathcal{V}$. 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. 3. 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.

3.1. Greedy Lineage Agglomeration (GLA)

The first algorithm takes an MLTP instance and constructs a feasible lineage in a bottom-up fashion. It is described in Alg. 1 and follows a similar scheme as the GAEC [28] heuristic for the MCMCP in the sense that it always takes the currently best possible transformation, starting from $\mathcal{V} = V$. It applies three different types of transformations: 1) a merge contracts all edges between two components of the same time point t, combining them into one single component. 2) setParent selects an arc $ab \in \mathcal{A}$ and thereby sets a of $\mathcal{V}_t$ as the (current) parent of $b \in \mathcal{V}_{t+1}$, while 3) changeParent de-selects such (active) arc ab and instead selects an alternative cb. While final components $\mathcal{V}$ determine intra-frame cuts, the final selection of arcs then

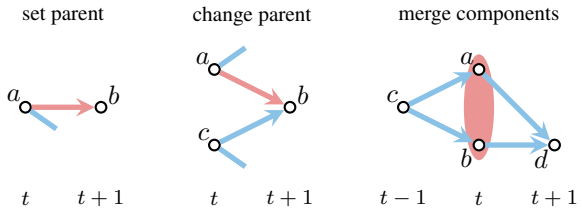

Figure 4. The three transformations of GLA: set a as parent of b (**left**), change the parent of b from c to a (**middle**) or merge two components a and b into one (**right**). The major arc along which the transformation occurs is depicted in red, while other arcs that affect the transformations cost are 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.

determines which temporal edges are cut edges ($x_e = 1$). Unlike GAEC, transformations concerning the temporal edges are reversible due to changeParent. All allowed transformations, merge, setParent and changeParent, are depicted in Fig. 4. The change in objective (4) caused by a particular transformation involving a and b is denoted with $\Delta_{ab}^{\text{transform}}$. In order to determine the cost or reward of a particular transformation, 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 objective. The detailed, incremental calculation of these transformation costs $\Delta_{ab}^{\text{transform}}$ 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 transformation decreases the objective.

Implementation. We use a priority queue to efficiently retrieve the currently best transformation. After applying it, each *affected* transformation is re-calculated and inserted into the queue. We invalidate previous editions of transformations 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 $\mathcal{A}(y)$.

3.2. Kernighan-Lin with Optimal Branchings (KLB)

Algorithm 2 takes an MLTP instance and an initial decomposition, *e.g.* the result of GLA, and attempts to decrease the objective function (4) in each step by changing

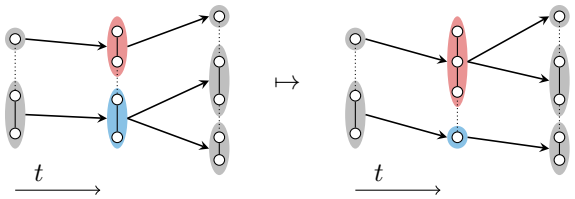

Figure 5. Depicted above is a transformation carried out by the KLB algorithm. One node in the middle image is moved from the blue component to the red component. Consequently, the optimal branching changes.

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
    improveBipartition$(\mathcal{G}, a, b)$ ▷ move nodes across border or merge.
  end for
  for $a \in \mathcal{V}$ **do**
    splitPartition$(\mathcal{G}, a)$ ▷ split partition.
  end for
end while
return cutEdgeLabels$(\mathcal{G})$ ▷ cut-edge labeling x^* from $\mathcal{V}$ and $\mathcal{A}(y^*)$.

the intra-frame partitions in a Kernighan-Lin-fashion [27], an example is illustrated in Fig. 5. Like the algorithm proposed by [28] for the related MCMCP, it explores three different local transformations 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. Transforms that do not decrease the objective will be discarded. In contrast to the setting of a MCMCP, judging the effect of such local modifications on the objective is more difficult, since it requires according changes to the temporal cut-edges. This can be seen when reordering the terms of the MLTP objective f_{MLTP} (4):

$$f_{\text{MLTP}}(x) = \sum_{e \in \bigcup_{t \in \mathcal{T}} E_{t,t+1}} c_e \; + \sum_{e \in \bigcup_{t \in \mathcal{T}} E_t} c_e x_e + f_{\text{MCBP}}(x) \;, \quad (8)$$

where we identify the first sum to be an instance-dependent constant, the second sum is the contribution from intra-frame edges (*i.e.* the decomposition into cells) and the last term, summarized with f_{MCBP} is the sum over all inter-frame edges as well as birth and termination costs. Given a particular KLB-transformation, the change to the intra-frame part is straight-forward to calculate, while the change of the inter-frame part involves solving $\min f_{\text{MCBP}}(.)$ anew. This sub-problem turns out to be a variant of a minimum cost branching problem (MCBP), which we discuss

next. Afterwards, we describe a combinatorial optimizer for this MCBP, and finally provide additional details on its usage within KLB.

Minimum Cost Branching on $\mathcal{G}$. Given a fixed decomposition into cells $\mathcal{V}$, *i.e.* is a fixed value for all intra-frame cut-edge variables x_e, we can reduce the remaining (partial) MLTP to the following MCBP over $\mathcal{G} = (\mathcal{V}, \mathcal{A})$:

$$\min_{y, y^-, y^+} \quad \sum_{ab \in \mathcal{A}} c_{ab} y_{ab} + \sum_{a \in \mathcal{V}} c_a^+ y_a^+ + \sum_{a \in \mathcal{V}} c_a^- y_a^- \tag{9}$$

$$\text{subject to} \quad \forall a \in \mathcal{V} \;:\; (1 - y_a^+) = \sum_{b \in \delta^-(a)} y_{ba} \tag{10}$$

$$\forall a \in \mathcal{V} \;:\; (1 - y_a^-) \leq \sum_{b \in \delta^+(a)} y_{ab} \leq 2 \tag{11}$$

$$y \in \{0,1\}^{\mathcal{A}}, \quad y^-, y^+ \in \{0,1\}^{\mathcal{V}} \;, \tag{12}$$

where y, y^-, y^+ are substitutes for those original cut variables x, x^+, x^- that are bundled within an arc or component in $\mathcal{G}$. The objective (9) is exactly f_{MCBP} of (8). Each y_{ab} indicates whether arc ab is active ($y_{ab} = 1$) or not ($y_{ab} = 0$). The equality constraint (10) 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, (11) enforces the penalty for termination if necessary, and its upper bound limits the number of children to 2, which enforces the bifurcation constraint. Since $\mathcal{G}$ is acyclic by construction, we do not require cycle elimination constraints that are typically present in general formulations of MCBPs. Observing that $\forall e \in E_{ab} : 1 - y_{ab} = x_e$, *i.e.* 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 derivation is found in the supplement.

Matching-Based Algorithm for the MCBP. We now show that the MCBP (9)-(11) can be solved efficiently by reducing it to a set of minimum cost bipartite matching problems (MCBMPs).

To this end, observe that the graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is acyclic by construction, cf. Fig. 3. Denote by $\mathcal{G}_{t,t+1} = (\mathcal{V}_t \cup \mathcal{V}_{t+1}, \mathcal{A}_{t,t+1})$ the subgraph of $\mathcal{G}$ that corresponds to the consecutive frames t and $t+1$.

Lemma 2. For every $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ arising from a fixed intra-frame decomposition, the solution of the MCBP on $\mathcal{G}$ can be found by solving the MCBP for all $\mathcal{G}_{t,t+1}$ individually.

Proof. The constraints (10) only couple birth variables y_a^+ for $a \in \mathcal{V}_{t+1}$ with arc variables y_{ba} where $ba \in \mathcal{A}_{t,t+1}$.

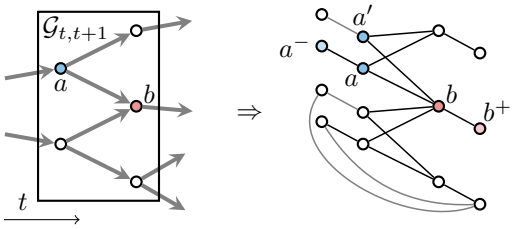

Figure 6. Illustration of the constructed bipartite matching problem (**right**) for an MCBP in the subgraph of two consecutive frames $t, t+1$ (**left**). The matching problem graph consists of the original nodes and edges, duplicates a' for $a \in \mathcal{V}_t$, auxiliary termination nodes a^- and auxiliary birth nodes b^+. Auxiliary edges which have zero cost by construction are gray. For simplicity, we illustrate only two edges between termination and birth nodes. Matched nodes correspond to active arcs in the original $\mathcal{G}_{t,t+1}$.

Similarly, the constraints (11) only couple termination variables y_a^- for $a \in \mathcal{V}_t$ with arc variables y_{ab} where $ab \in \mathcal{A}_{t,t+1}$. Thus, the objective function and the constraints split into a set of MCBPs corresponding to the subgraphs $\mathcal{G}_{t,t+1}$ of $\mathcal{G}$. Hence, solving $|\mathcal{T}| - 1$ many sub-MCBPs individually gives the solution of the MCBP on $\mathcal{G}$. □

Lemma 3. An MCBP on $\mathcal{G}_{t,t+1}$ can be transformed into an equivalent minimum cost bipartite matching problem (MCBMP).

Proof. For a given MCBP on $\mathcal{G}_{t,t+1}$, we construct an MCBMP as follows (illustrated in Fig. 6): 1) insert a duplicate a' for each node $a \in \mathcal{V}_t$ and add an arc $a'b$ for each original arc $ab \in \mathcal{A}_{t,t+1}$ with identical cost $c_{a'b} = c_{ab}$. 2) For each node $a \in \mathcal{V}_t$, insert a node a^- and an arc aa^- with its cost being c_a^-, *i.e.* the cost of terminating in a. Repeat this for all duplicate nodes a' but set the according cost $c_{a'}^- = 0$. Similarly, add a node b^+ for each $b \in \mathcal{V}_{t+1}$ and an arc b^+b with a cost of c_b^+. 3) Connect each pair of auxiliary nodes b^+ and a^- (or a'^-) with an arc if $ab \in \mathcal{A}_{t,t+1}$ with a cost of 0. The resulting graph is clearly bipartite.

Now, consider the MCBMP on this graph: A match (a, b) or (a', b) corresponds to $y_{ab} = 1$, respectively $y_{a'b} = 1$, a match of (a, a') to $y_a^- = 1$ and vice versa for birth variables y_b^+. Exactly one incoming arc for each node of $\mathcal{V}_{t+1}$ or the link to its birth node b^+ is matched, satisfying (10). In the same fashion, each $a \in \mathcal{V}_t$ is assigned to a node $b \in \mathcal{V}_{t+1}$ or its termination node a^-, satisfying the left hand side of (11). Assigning a duplicate node a' to a node $b \in \mathcal{V}_{t+1}$ allows having bifurcations, *i.e.* satisfies the right-hand side of (11), while its alternative choice, matching it to its zero-cost termination node has no effect on the cost. Finally, the zero-cost arcs between the auxiliary birth and termination nodes a^- and b^+ are matched whenever a pair of a or and b is matched (due to lack of alternatives). □

The MCBMP can be solved in polynomial time by the hungarian algorithm [30, 35]. Applying it to each of the $|\mathcal{T}|-1$ subgraphs of $G_{t,t+1}$ thus leaves us with an efficient minimizer for the MCBP.

Implementation of KLB. The algorithm maintains the weighted $\mathcal{G}=(\mathcal{V},\mathcal{A})$, the current objective in terms of each of the three parts of (8), and solves the MCBP on $\mathcal{G}$ by the matching-based algorithm described in the previous section. We initially solve the entire MCBP, but then, within both methods that propose transformations, improveBipartition and splitPartition, we exploit the locality of the introduced changes. By applying Lemma 2, we note that for a given $\mathcal{V}$, modifying two of its cells a and b in frame t will only affect arcs that go from $t-1$ to t and from t to $t+1$. In other words, Δf_{MCBP} can be computed only from the subproblems of $(t-1,t)$ and $(t,t+1)$. In practice, we find that the effect is often also spatially localized, hence we optionally restrict ourselves to only updating the MCBP in a range of d_{MCBP} (undirected) arc hops from a and b, where the modification occured. This d_{MCBP} parameter should be explored and set depending on the instance, since choosing it too small may result in misjudged moves and thus, in wrong incremental changes to the current objective. Note, however, that feasibility is still maintained in any case. We handle this by solving the *entire* MCBP once at the end of every outer iteration. Doing so ensures that the final objective is always correct and allows us to detect choices of d_{MCBP} that are too small. Since we observe that it takes relatively few outer iterations, we find the overhead by these extra calls to be negligible.

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.

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

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. [14] and [19]). This argument can be analogously transferred to inequalities (2) and (3).

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 4 shows that with both results combined, we can equivalently replace (1) – (3) by the set of tighter inequalities (13) and (14). Proofs are provided in the supplementary material. In relation to our improved version of the branch-and-cut algorithm, we refer to (13) as *cycle* and to (14) as *morality* constraints.

Lemma 4. For every hypothesis graph $G=(V,E)$ it holds that $x\in X'_G$ iff $x\in\{0,1\}^E$ and x satisfies

$$\forall t\in\mathcal{T}\,\forall\{v,w\}\in E_t\cup E_{t,t+1}$$
$$\forall \text{ chordless } vw\text{-paths } P \text{ in } G_t^+ :$$
$$x_{vw}\le\sum_{e\in P}x_e \tag{13}$$
$$\forall t\in\mathcal{T}\,\forall v',w'\in V_t \text{ such that } \{v',w'\}\notin E_t$$
$$\forall v'w'\text{-cuts } S \text{ in } G_t\,\forall \text{ chordless } v'w'\text{-paths } P \text{ in } G_t^+ :$$
$$1-\sum_{e\in S}(1-x_e)\le\sum_{e\in P}x_e \tag{14}$$

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 (14) is exactly the combination of a *cut* inequality $1-x_{v'w'}\le\sum_{e\in S}(1-x_e)$ and a *path* inequality $x_{v'w'}\le\sum_{e\in P}x_e$ in the sense of *lifted multicuts* [19]. 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.

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 5. 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:

$$\forall t\in\mathcal{T}\,\forall v\in V_{t+1}\,\forall S\in V_t v\text{-cuts}(G_t^+) :$$
$$1-x_v^+\le\sum_{e\in S\setminus E(V_t(v),V_{t+1}\setminus\{v\})}(1-x_e). \tag{15}$$

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

$$\forall t\in\mathcal{T}\,\forall v\in V_t\,\forall S\in vV_{t+1}\text{-cuts}(G_t^+) :$$
$$1-x_v^-\le\sum_{e\in S\setminus E(V_t\setminus\{v\},V_{t+1}(v))}(1-x_e) \tag{16}$$

hold true.

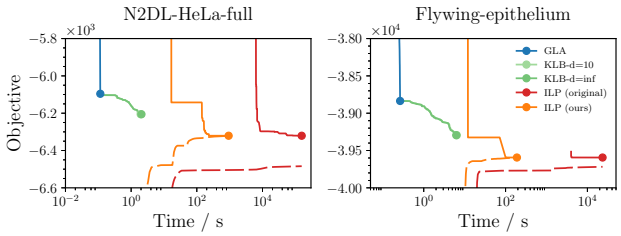

Figure 7. Comparison of algorithms for the MLTP in terms of runtime, objective (solid) and bounds (dashed) on the large instances of [21]. Our heuristics are able to determine feasible solutions quickly, while our branch-and-cut algorithm (ILP ours) converges to the optimal solution in up to one hundredth of the time of the original branch-and-cut algorithm (ILP original) and provides tight bounds in both cases. On these instances, KLB exhibits no significant runtime difference between the two choices of d_{MCBP}.

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

$$\sum_{e \in E_C} x_e - \sum_{e \in E_w} x_e \leq \left\lfloor \frac{|V(W)| - 1}{2} \right\rfloor, \qquad (17)$$

which is valid for multicut polytopes [14]. 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 [14].

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, (17) holds with respect to x.

Implementation. For a subset of the constraints, we use the commercial branch-and-cut solver Gurobi (7.0) [18] 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 (13) – (16) as well as violated bifurcation constraints and repeat. To this end, we adapt the separation procedures of [21] to account for our improvements in a straight-forward manner. We further add odd wheel inequalities for wheels with 3 outer nodes as described above (so-called *3-wheels*) to the starting LP relaxation.

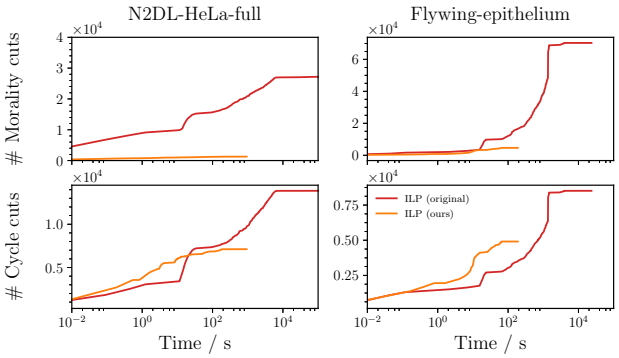

Figure 8. Number of morality cuts (**top**), *i.e.* (3) or (14), and cycle cuts (**bottom**), *i.e.* (1) and (2) or (13), separated in the different branch-and-cut algorithms. We observe that our branch-and-cut algorithm requires considerably fewer morality cuts, while the number of cycle cuts (including both space-cycles and space-time-cycles) is in the same order of magnitude.

For every integer feasible solution that Gurobi finds, we fix the connected components of the intra-frame segmentation and solve the remaining MCBP. This allows for the early extraction of feasible lineage forests from the ILP.

5. Experiments & Results

Instances and Setup. We evaluate our algorithms on the two large instances of [21]: *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. Their 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 [21].

Our choice of birth and termination costs follows [21], *i.e.* 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 two versions of KLB: The first one is denoted with KLB-d=inf and solves the MCBP within the (reachable) subgraph of $t \pm 1$, while the second, KLB-d=10, additionally exploits spatial locality, *i.e.* it uses $d_{\mathrm{MCBP}} = 10$.

Convergence Analysis. The convergence of our algorithms in comparison to the branch-and-cut algorithm of [21] is reported in Fig. 7 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 $1.86\,\%$. Both variants of KLB obtain the same solution in terms of cut-edge labeling and show no

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. KLB-d=inf solves the MCBP in the entire reachable subgraph of $\{t-1, t, t+1\}$, while KLB-d=10 additionally uses spatial locality with $d_{\mathrm{MCBP}} = 10$.

Method	Flywing-epithelium					N2DL-HeLa-full				
	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-d=10	6.42	-39294.65			0.0076	1.95	-6205.54			0.0186
KLB-d=inf	6.24	-39294.65			0.0076	2.06	-6205.54			0.0186
ILP (ours)	189.41	-39593.90	-39593.90	0.0000	0.0000	931.07	-6320.81	-6320.81	0.0000	0.0000
ILP (original) [21]	23460.80	-39593.90	-39717.80	0.0031	0.0000	156542.00	-6320.81	-6484.02	0.0258	0.0000

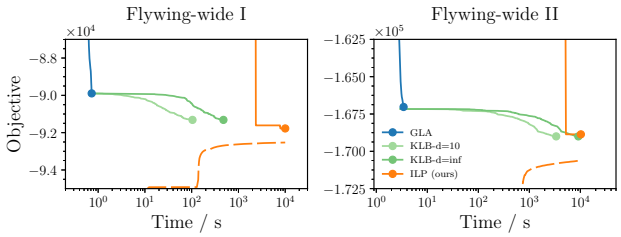

Figure 9. Results on the more extensive instances *Flywing-wide I* and *II*. Our branch-and-cut algorithm with 3-wheel constraints provides slightly tighter bounds, with which we determine the gaps for GLA to be 2.9 % (*I*) and 2.1 % (*II*), and 1.3 % (*I*) and 0.95 % (*II*) for KLB. Exploiting spatial locality when re-solving the MCBPs considerably reduces runtime of KLB.

considerable runtime difference. 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 KLB implementation could potentially be sped up by updating components (of disjoint $\mathcal{G}_{t-1:t+1}$) in parallel.

The improved branch-and-cut algorithm retrieves feasible solutions considerably faster and provides tighter bounds than the algorithm of [21]. The instances *Flywing-epithelium* and *N2DL-HeLa* are solved to optimality in less than 200 s, respectively 1000 s, while the original algorithm did not find any feasible solutions in that time. As shown in Fig. 8, we observe that our modifications of the branch-and-cut algorithm greatly reduce the number of morality cuts.

On the larger instances *Flywing-wide I* and *II*, we present our results in Fig. 9. 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 0.95 % (*II*) for KLB. Again, both variants of KLB obtain identical solutions. Here, exploiting spatial locality helps: KLB-d=inf runs in 477 s (*I*) and 9129 s (*II*), while KLB-d=10 reduces this to 104 s and 3359 s, respectively. The particular choice of $d_{\mathrm{MCBP}} = 10$ was found to be stable in both cases. More extensive results with varying d_{MCBP} can be found in the supplement.

Solution Quality. We compare the solution quality of our two heuristics by segmentation (SEG) and tracking (TRA) metrics as used in [33] for *Flywing-epithelium*. The results 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 ILP solutions achieve slightly better scores in both measures than the heuristics. All presented algorithms outperform the baseline, the *packing analyzer* [1], whose scores were originally reported in [21].

Table 2. Comparison of the similarity to 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

6. Conclusion

We have introduced local search algorithms for the recently introduced MLTP [21], a mathematical framework for cell lineage reconstruction, which treats both subproblems, image decomposition and tracking, jointly. We propose two efficient heuristics for the MLTP: a fast agglomerative procedure called GLA that constructs a feasible lineage bottom-up, and a variant of the KL-algorithm which attempts to improve a given lineage by switching nodes between components, merging or splitting them. The latter algorithm repeatedly solves a MCBP conditioned on fixed partitions. We show that this subproblem can be solved as a minimum cost bipartite matching problem, which is of independent interest. Furthermore, we improve the branch-and-cut algorithm of [21] by separating tighter cutting planes and employing our result about the MCBP subproblem. Our branch-and-cut algorithm solves previous instances quickly to optimality. For both the previous and larger instances, our heuristics efficiently find high quality solutions. This demonstrates empirically that our methods alleviate runtime issues with MLTP instances and makes *moral lineage tracing* applicable in practice (*e.g.* in [38]).

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. 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, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *ICCV*, 2011. 2

[5] 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. 2

[6] 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. 2

[7] S. Bagon and M. Galun. Large scale correlation clustering optimization. *CoRR*, abs/1112.2903, 2011. 2

[8] 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. 2

[9] T. Beier, F. A. Hamprecht, and J. H. Kappes. Fusion moves for correlation clustering. In *CVPR*, pages 3507–3516, 2015. 2

[10] 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. 2

[11] 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

[12] 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

[13] 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

[14] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1):87–115, 1993. 6, 7

[15] 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, 2

[16] 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

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

[18] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016. 7

[19] A. Horňáková, J.-H. Lange, and B. Andres. Analysis and optimzation of graph decompositions by lifted multicuts. In *ICML*, 2017. 2, 6

[20] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. ArtTrack: Articulated multi-person tracking in the wild. In *CVPR*, 2017. 1

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

[22] 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, 2

[23] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnörr. Higher-order segmentation via multicuts. *Computer Vision and Image Understanding*, 143:104–119, 2016. 2

[24] 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, 2

[25] 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

[26] 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

[27] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970. 3, 4

[28] 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. 2, 3, 4

[29] 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. 2

[30] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[31] E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, and B. Andres. Joint graph decomposition and node labeling: Problem, algorithms, applications. In *CVPR*, 2017. 2

[32] 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

[33] 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, 8

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

[35] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 6

[36] 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, 2

[37] M. Rempfler, B. Andres, and B. Menze. The minimum cost connected subgraph problem in medical image analysis. In *MICCAI*, pages 397–405, 2016. 1

[38] M. Rempfler, S. Kumar, V. Stierle, P. Paulitschke, B. Andres, and B. Menze. Cell lineage tracing in lens-free microscopy videos. In *MICCAI*, 2017. (in press). 8

[39] M. Rempfler, M. Schneider, G. D. Ielacqua, X. Xiao, S. R. Stock, J. Klohs, G. Székely, B. Andres, and B. H. Menze. Reconstructing cerebrovascular networks under local physiological constraints by integer programming. *Medical Image Analysis*, 25(1):86 – 94, 2015. 1

[40] 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, 2

[41] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht. Conservation tracking. In *ICCV*, pages 2928–2935, 2013. 1, 2

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

[43] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017. 1

[44] 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

[45] 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, 2

[46] 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, 2

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

[48] J. Yarkony, A. Ihler, and C. C. Fowlkes. Fast planar correlation clustering for image segmentation. In *ECCV*, pages 568–581, 2012. 2

[49] 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. 2

[50] 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