Fakultät für Informatik
Technische Universität München

Graph Inference in Biomedical Image Analysis

Markus Rempfler

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:
Prof. Dr. Stephan Günnemann

Prüfende der Dissertation:
1. Prof. Dr. Björn H. Menze
2. Prof. Dr. Daniel Cremers

Die Dissertation wurde am 11.04.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 07.12.2018 angenommen.

Abstract

Graphs are an abstract representation for several structures of interest in biology, such as vessels or neurons, and they are also well-suited for describing spatio-temporal behaviour of multiple objects along time. With the progress of advanced biomedical imaging techniques and an emerging demand for quantitative methods, computer-aided analysis of biomedical image data has gained considerable importance. Amongst such analysis tasks is the estimation of a graph from images or videos. A task we refer to as *graph inference* and which can be located at the intersection of the fields machine learning, computer vision and optimization.

In this thesis, we focus on methods that formulate graph inference as a two-stage process: First, a hypothesis (super-)graph is constructed from the observed data and then, a combinatorial optimization problem is solved to determine the subgraph that best describes the object of interest. With respect to the application, we address two prominent forms of the graph inference problem, namely estimation of vessel networks and cell lineage forests. Specifically, our contributions are as follows. (1) We propose a probabilistic model for utilizing physiological knowledge obtained from high-quality data when reconstructing vascular networks. (2) We identify its underlying optimization problem as minimum cost connected subgraph problem (MCCSP), devise more efficient constraint generation schemes for exact algorithms and benchmark them against heuristics on two medical datasets. (3) We design two primal feasible heuristics and improve the branch-and-cut algorithm for the moral lineage tracing problem (MLTP), a joint clustering and tracking formulation for cell lineage tracing. We demonstrate that they reduce runtime and improve scalability considerably. (4) Finally, we propose a cell lineage tracing framework for lens-free microscopy (LFM) videos, combining the MLTP and fully convolutional neural networks (FCNs), and demonstrate its capabilities to estimate lineages of high quality.

Zusammenfassung

Graphen sind eine abstrakte Repräsentation für verschiedenste biologische Strukturen, wie beispielsweise Neuronen oder Blutgefäße, und sind desweiteren geeignet um das Verhalten mehrerer Objekte in Raum und Zeit zu beschreiben. Mit dem Fortschritt biomedizinischer Bildgebungstechnologien und dem wachsenden Interesse an quantitativen Methodologien haben computergestützte Analysemethoden für biomedizinische Bilddaten zunehmend an Wichtigkeit gewonnen. Darunter fällt auch das Schätzen von Graphen aus Bilddaten und Videos, die sogenannte *Graphinferenz*, welche sich in den Gebieten des maschinellen Lernens, der Bildverarbeitung und der Optimierung einordnen lässt.

In dieser Thesis liegt der Fokus auf zwei-Phasen Methoden zur Graphinferenz. Als erstes wird ein Hypothesisgraph aus den Bilddaten konstruiert, um im Anschluss den besten Subgraphen darin mittels kombinatorischer Optimierung zu bestimmen. Dabei werden zwei weitverbreitete Formen von Graphinferenzapplikationen adressiert: Das Schätzen von Gefäßnetzwerken sowie das Tracken von Zellen und deren Abstammung, d.h. sogenannte Lineage Forests. Im Spezifischen werden die folgenden Beiträge beschrieben: (1) Ein probabilistisches Modell zur Rekonstruktion von Gefäßnetzwerken, welches physiologisches Vorwissen berücksichtigen lässt. (2) Dessen unterliegendes Optimierungsproblem wird als minimum cost connected subgraph problem (MCCSP) identifiziert, wofür effizientere Strategien für exakte Algorithmen vorgeschlagen und diese mit Heuristiken auf zwei Benchmarkdatensätzen verglichen werden. (3) Für das moral lineage tracing problem (MLTP), eine joint Clustering und Tracking Formulierung des Zelltrackingproblems, werden die ersten zwei Suchheuristiken designt und der bisherige branch-and-cut Algorithmus verbessert, was zu wesentlich kürzeren Laufzeiten und einer besseren Skalierbarkeit führt. (4) Schliesslich wird ein System zur Analyse von Zellen in lens-free microscopy (LFM) beschrieben, welches

das MLTP mit fully convolutional neural networks (FCNs) kombiniert, und dessen Leistungsfähigkeit zur Schätzung von qualitativ hochwertigen Lineage Forests empirisch demonstriert wird.

Acknowledgements

First of all, I would like to thank my advisor, Bjoern H. Menze, who offered me to pursue a doctoral degree in his new group in Munich about four years ago. I am grateful for his advice and guidance, sharing his personal experiences, as well as for providing a flexible research environment. I would also like to express my gratitude to my co-referee Daniel Cremers for reviewing this thesis.

Next, I would like to acknowledge the support of my co-authors. Their inputs, be it in the form of data, annotations, advice or feedback, were always highly appreciated. Special thanks goes to Bjoern Andres, who was an invaluable source of advice throughout my doctoral studies, and to Matthias Hasler, from whose expertise I benefitted greatly in the early stage of my research.

Furthermore, I would like to thank my colleagues and friends of the image-based biomedical modeling group, particularly (in alphabetical order) Esther Alberts, Marc Bickel, Marie Bieth, Patrick Christ, Dhritiman Das, Jana Lipkova, Anjany Sekuboyina, Cagdas Ulas and Alexander Valentinitsch. They contributed to an enjoyable work environment and are responsible for many good memories of Munich.

Beyond that, many friends outside my immediate academical environment have helped me to stay motivated and be able to return to my research with replenished energy. I would like to thank a few in particular: A big thanks to Christian Meier and Niklaus Meier, for always being up for an activity whenever I am back in the *Appenzellerland*. Another big thanks to Raphael Meier, for being a great friend since our Bachelor studies and for the countless insightful conversations – in fact, back in 2013, Raphael mentioned that Bjoern might have an interesting topic for a Master thesis, which led to the very first step on the long way to this dissertation. Furthermore, my thanks goes to Heinz Füglister and Christian Käslin, for our culinary

gatherings in and around Zürich, Thomas Gerig, for the refreshing discussions in *schwizerdütsch* at events like MICCAI, ICCV or MLSS, and Jörg Encke for being an extraordinary boulder companion. I would like to express a special thanks to Pauline for making the years of my doctoral studies much more joyful and for (sometimes) making me question my concepts and ideas. Last but not least, I would like to deeply thank my family, in particular my parents, Lucia and Martin, for their unconditional support through all my life.

Contents

List of Figures

List of Tables

Acronyms

BFS . . . . . breadth-first search

CNN . . . . . convolutional neural network

CT . . . . . computed tomography

DOHT . . . . . distance-ordered homotopic thinning

DTP . . . . . disjoint tree problem

FCN . . . . . fully convolutional neural network

FN . . . . . false negative

FP . . . . . false positive

ILP . . . . . integer linear program

LFM . . . . . lens-free microscopy

LP . . . . . linear program

MAP . . . . . maximum a posteriori

MCBMP . . . . . minimum cost bipartite matching problem

MCBP . . . . . minimum cost branching problem

MCCS . . . . . minimum cost connected subgraph

MCCSP . . . . . minimum cost connected subgraph problem

MCDAP . . . . . minimum cost disjoint arborescence problem

MCMCP . . . . . minimum cost multicut problem

µCT . . . . . micro computed tomography

µMRA . . . . . magnetic resonance microangiography

MILP . . . . . mixed integer linear program

MLT . . . . . moral lineage tracing

MLTP . . . . . moral lineage tracing problem

MODA . . . . . multiple object detection accuracy

MOTA . . . . . multiple object tracking accuracy

MRF . . . . . Markov random field

MRI . . . . . magnetic resonance imaging

NN . . . . . neural network

RF . . . . . random forest

SFT . . . . . steerable filter templates

SGD . . . . . stochastic gradient descent

SVM . . . . . support vector machine

TP . . . . . true positive

TRA . . . . . tracking accuracy

TSP . . . . . traveling salesman problem

1

Introduction

Graphs are an abstract representation for several objects of interest in biology and medicine. Examples include tree- and network-like objects like neurons or vessels, where the graph's vertices describe the structure in terms of its spatial location and the edges their anatonomical connectedness. Graphs are also well-suited for describing spatio-temporal behaviour: the trajectories of multiple objects along time can be represented as a graph of multiple (disjoint) chains. In this case, vertices are associated with an objects position in space and time and edges identify identical objects along time. When tracking objects like cells that both move and divide, then the underlying graph that describes their motion and descendance is again a special type of graph, namely a forest, and called the *cell lineage forest*. Figure 1.1 illustrates how a graph can serve as an abstract representation of an anatomical structure, in this case vessels. Two more examples are presented in Fig. 1.2, which depict a vessel network and a cell lineage forest. Beyond this, graphs are utilized to represent functional networks. For example in the analysis of brain connectivity, where the vertices are different regions in the brain and edges describe which regions interact (i.e. stimulate/inhibit) with each other [2]. The same concept is also employed in the analysis of regulatory networks that are common subjects of interest in metabolics [3, 4], genomics [5, 6], or proteomics [7, 8].

Of course, graphs are not only interesting for biological questions. Graph theory is a longstanding and rich field of mathematics and computer science [9, 10]. Its origin is commonly dated back to L. Euler's solution to the *seven bridges of Königsberg* problem in 1736 [11]. Concepts from graph theory have been transferred to a multitude of disciplines. Examples range from analysis of social networks [12] or transportation networks [13], to design, operation and analysis of computer networks [14, 15]. Comprehensive

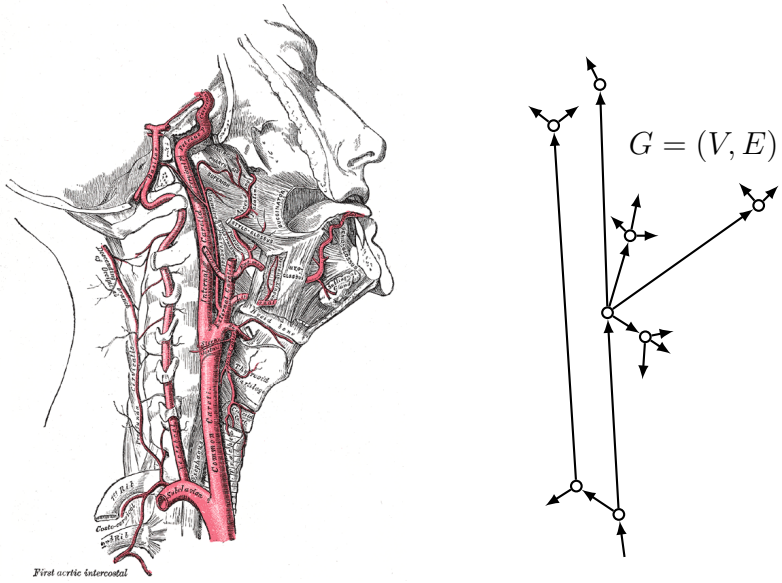


Figure 1.1: Example of a graph as abstract representation of an anatomical structure. The left side illustrates internal carotid and vertebral arteries in the neck (image source: H. Gray [1, Fig. 513]), whereas the right side depicts a graph G that represents these arteries abstractly. Its vertices V, drawn as circles, represent furcations and its edges E indicate vascular segments, drawn as arrows. Open ends in the illustration are shown as edges with only one vertex. Smaller vessels are omitted in graph G for simplicity.

review and discussion of graphs and their analysis in different fields can be found in [16, 17, 18, 19].

Ultimately, analysing the graph of interest is expected to gain quantitative insights, e.g. into biological processes, and allow the researchers to assess their hypothesis. However, these graphs are typically observed indirectly as they are an abstraction for the real-world object of interest. While modern imaging techniques such as computed tomography (CT), magnetic resonance imaging (MRI) or optical microscopy and their specialized variants allow observing a wide range of structures and processes, the obtained data is typically in the form of 2D or 3D images or a times series of such. Hence,

Figure 1.2: Examples of graphs in biomedical image analysis. A vascular network graph (**left**) reconstructed from a CT of a corrosion casted mouse brain and rendered with radius estimates along edges (from Chapter 3, [20]), and a cell lineage forest (**bottom**) estimated from a microscopy video originating from continuous monitoring of 3T3 cells in-vitro (from Chapter 6, [21]).

the underlying graph needs to be estimated from the pixel- or voxel-based information of the acquired images at hand. This estimation is what we consider *graph inference* in this thesis.

We focus on methods for graph inference that formulate the task as a two-stage process. First, a hypothesis graph is constructed from the observed data. This hypothesis graph contains a multitude of potential graphs, i.e. single hypotheses. Then, in the second step, the graph that explains the observations best amongst these hypotheses is determined, which means solving a combinatorial optimization problem. From a technical point of view, there are several aspects to such an approach that have to be addressed. Constructing the hypothesis graph requires robust detectors and models for translating image evidence into candidate nodes and edges, as well as a formal encoding of *feasible* hypotheses that can be built from these candidates. A model for ranking these hypotheses against each other is needed to determine whether a given feasible hypothesis fits the observed data better than another. Finally, an optimization procedure has to be devised that is able to find the optimal feasible hypothesis (or at least a reasonably good one) within acceptable time. Besides a potential trade-off between inference runtime and solution quality, limited available data for parameter fitting and validation further calls for a carefully chosen model complexity. Importance and difficulty of each of these aspects depends on the application. Our work considers two different applications of graph inference in biomedical image analysis: the reconstruction of curvilinear structures such as vasculature or neurons, and tracing cell lineage forests from microscopy videos.

In the next section, we discuss the graph inference problem in more detail and interpret two main applications of this thesis as graph inference. Subsequently, we summarize our main contributions in Section 1.2 and outline the remaining manuscript in Section 1.3.

1.1 Graph Inference

We define graph inference (on an high level of abstraction) as a mapping h from observations $\mathbf{f}$ to a graph $G = (V, E)$:

$$h : \mathbf{f} \mapsto G = (V, E) \ . \tag{1.1}$$

In the context of biomedical image analysis, the observations $\mathbf{f}$ are typically images. Vertices V and edges E of the resulting graph will have varying meaning, depending on the particular application. The graph G might have additional information associated with vertices and/or edges, such as a semantic vertex labeling or a continuous weight for each edges strength. We consider this to be absorbed in V and E at this point for ease of notation. The term graph inference is used only in few works explicitly, e.g. [22, 23, 24, 25], whereas most use application specific terminology. We will next discuss different forms of graph inference.

A distinction can be made with respect to the parts of the graph that are to be inferred, which depends on both application and type of observations. It might be that the vertices V of the graph G are given, e.g. when they are observed entities, such as users in a database, while the unknown factor are (some of) their relationships, i.e. the edges E. In this case, graph inference is cast as a matrix completion problem, where the matrix to be completed is the graphs adjacency matrix [26, 27, 28, 24]. A well-known example is the Netflix challenge [29], where the estimated graph is supposed to serve as a recommender system, another is the estimation of dependencies in multivariate statistical analysis [27]. It could also be that both nodes and edges have to be estimated from the given data. This is typically the case for estimating vascular networks from image data [20, 30, 31, 32], but also for more general tasks such as multi-object tracking [33, 34, 35, 36, 37]. Another variation is the case where a subset of *both* vertices V and edges E is known, while some of them have to be inferred [22]. In others, an entire graph is already known and the goal is to determine one or more subgraphs within it. For example, Qian et al. [38, 39] formulate tasks like intrusion detection in networks or disease outbreak detection as such a subgraph inference problem, where they scan the network for subgraphs with a certain elevated mean statistic. In the same fashion, another work poses the problem of determining a core network of brain connectivity from a population of patient scans as subgraph inference problem [40].

The complex nature of estimating a structure such as a graph from possibly large datasets makes it often necessary to break the mapping h down into several subfunctions. As mentioned previously, we consider approaches that can be interpreted as a two-stage process: in the first step, a set of hypothesis graphs is created. In the second, the most likely hypothesis

graph is selected. We can write this as a composite mapping

$$h = h_2(h_1(\mathbf{f})) \ , \tag{1.2}$$

where h_1 constucts hypotheses and h_2 selects the best (in some sense) amongst them. In a sense, this interpretation separates the typically highly domain-specific hypothesis construction – sometimes also referred to as *preprocessing* – from more general optimization approaches for selecting the final hypothesis. This interpretation has the advantage that knowledge gained in different graph inference tasks can be transferred. For example, in both the intrusion detection of [38, 39] and the core brain connectivity network problem of [40], the second part h_2 is essentially a minimum cost connected subgraph problem (MCCSP) – an optimization problem for which we discuss strategies in Chapter 4. In our work, we will aim at providing a probabilistic interpretation of the second step. That is, we will discuss estimators $h_2(\mathcal{H})$ of the form

$$G^* = h_2\left(\mathcal{H}\right) = \arg\max_{G \in \mathcal{H}} P\left(G|\mathcal{H}, \mathbf{\Theta}\right) \ , \tag{1.3}$$

where h_2 is the maximum a posteriori (MAP) estimate under a probablistic model denoted by $P\left(G|\mathcal{H}, \mathbf{\Theta}\right)$ with parameters $\mathbf{\Theta}$, and $\mathcal{H}$ are the hypotheses constructed by h_1. We will see that this can be computationally challenging, which is the reason why parts of this work focus on more efficient and/or heuristic optimization strategies.

1.2 Summary of Contributions

This thesis is set in the context of graph inference in biomedical image analysis. We have selected two major graph inference problems that have received considerable attention in the past due to their relevance for biological and medical questions: First, the inference of vascular networks or related curvilinear structures such as neurons, and second, the estimation of cell lineage forests from microscopy videos.

In the following, we give a brief introduction to the setting of each publication-based chapter and summarize its content and contributions.

Chapter 3: Reconstructing Cerebrovascular Networks under Local Physiological Constraints by Integer Programming

Cerebrovascular networks are an object of interest in the study of several diseases. Examples include arteriosclerosis and dilative vascular malformations, as well as Alzheimer's and related neuro-degenerative diseases – all of which are suspected to be related to vascular reformation processes [41, 42]. The task of reconstructing the vascular network, i.e. the graph inference, from image data such as specialized MRI or CT sequences is thus an important part of medical image computing. Several works propose methods to vessel segmentation and tracing, ranging from hand-crafted filters to machine learning-based approaches [43, 44, 45, 46, 47, 48, 46, 48, 49, 50, 51]. Related literature is also found in the field of neuron tracing [52, 53, 54] or in the context of curvilinear structure detection in general [55, 56, 57, 58, 30]. In contrast to this, far fewer works focus on estimating the vasculature graph under consideration of its expected topology [59, 60]. A recent method [30] formulates this task in an integer programming framework, resulting in a combinatorial optimization that determines the best subgraph within an overconnected hypothesis graph. This method has been extended or adapted to several variants of the subgraph inference problem. They include anatomical labeling of vessels [31], artery-vein separation [32], proofreading of curvilinear detections [61] or, in this chapters work [20], for transferring statistics from high-resolution to the extraction of networks from low-resolution datasets.

We contribute a probabilistic model for reconstructing a vascular network from noisy data that allows to incorporate local statistics obtained from a high-resolution micro computed tomography (µCT) dataset. We present experiments with µMRA and show that the reconstructed networks have physiologically more plausible properties than without the transferred knowledge. To address the computational complexity that comes with the size of the instances, we propose a scheme to dynamically decompose the optimization problem into subproblems.

Chapter 4: The Minimum Cost Connected Subgraph Problem in Medical Image Analysis

Connectedness constraints are important for the estimation of several objects of interest from biomedical images, most prominently for complexes of thin, elongated structures such as vessels or neurons. This type of constraints find application in [31, 20, 32, 30] for a variety of graph inference tasks on biomedical data. Imposing these constraints on the subgraph inference results in an *MCCSP*, which is known to be NP-hard (a proof by reduction to the Steiner tree problem [62, Chapter 20] is presented in [63]). It is thus difficult to optimize in practice and the main cause for extensive runtimes on certain instances encountered in Chapter 3 [20]. This issue is particularly problematic if the instance is large, as it is often the case with volumetric medical image data. Approaches therefore either aim to reduce the instances size when working with exact solvers, e.g. by using superpixels [64], by substituting an easier surrogate optimization problem [63, 65] or approximative formulations of the connectedness constraint [66, 67, 68].

In this chapter, we discuss the MCCSP in an integer programming framework. This includes in particular the exact formulation of connectedness as in [64] as well as the formulation of connectedness along a precomputed geodesic [66, 67, 68]. We contribute several constraint generation strategies for a branch-and-cut algorithm, as well as objective-dependent constraints that reduce the set of feasible solutions without affecting optimal solutions. These are shown to improve scalability of exact optimization in practice and thus enable a quantitative benchmark of exact and approximative algorithms on two medical datasets of vessels and neurons [69, 53], where we identify the geodesic tree heuristic to be an excellent choice in practice.

Chapter 5: Efficient Algorithms for Moral Lineage Tracing

Analysis of microscopy videos is a crucial step in gaining a better understanding in several fields of biology, including embryonic development [70, 71], tissue formation [72] or metastatic behavior of tumor cells [73]. The estimation of *cell lineage forests* from such microscopy videos is therefore a well-established image analysis problem, which has received considerable attention [74, 75, 76, 77, 78, 79].

The task is commonly tackled in two phases: First, individual cells are detected and/or segmented in every frame and then, in a second phase, linked to represent displaced cells in the subsequent frames or the two daughter cells in case of a division. The second step is often cast as an optimization problem: the minimum cost disjoint arborescence problem (MCDAP) [80, Chapter 53] or variations of it [81, 82, 83, 84, 85]. Solutions to this are disjoint trees of detections, implicitly assuming a one to one relation between detections and cells. A more recently proposed mathematical abstraction called moral lineage tracing problem (MLTP) [86] formulates the task as joint-clustering and tracking problem and thereby defines its feasible solutions as disjoint trees of *clusters* of detections. It is a hybrid of the MCDAP and the minimum cost multicut problem (MCMCP), which is a well-known method for image decomposition [87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98]. As such, it anticipates the possibility that one cell may yield several detections (or is oversegmented), which is not considered wrong but are clustered to represent the underlying cell together. The branch-and-cut algorithm for optimizing the moral lineage tracing problem (MLTP) proposed in [86], however, is prone to a large number of cuts and exhibits slow convergence on large instances. Therefore, many applications of the MLTP in practice are prevented by its high computational cost.

In this chapter, we address this problem. We design two primal feasible local search heuristics for the MLTP and we improve the original branch-and-cut algorithm by separating tighter cutting planes. We demonstrate convergence of our algorithms on the problem instances of [86], solving two (previously unsolved) instances to optimality and obtaining accurate solutions orders of magnitude faster. Further, we show improved scalability on previously inaccessible instances.

Chapter 6: Cell Lineage Tracing in Lens-free Microscopy Videos

The majority of work on automatic cell lineage tracing focusses on processing traditional light microscopy videos. However, recent progress in LFM has rendered it a promising alternative to traditional light microscopy for certain applications, such as continouos monitoring of experiments with cultured cells [99, 100, 101]. LFM is a label-free microscopy technique, i.e. no additional marker like a fluorescent dye is required. Thus, it does not

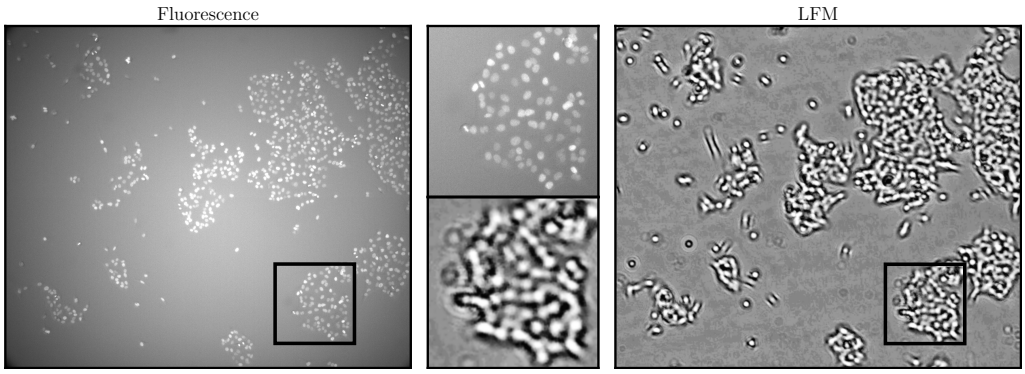


Figure 1.3: A cell culture observed through lens-free microscopy (LFM) (**right**) and traditional fluorescence microscopy (**left**). The central column shows a detail view of the region indicated with a black rectangle in both image types. LFM has the advantage of being a label-free technique, thus avoiding limitations inherent to a fluorescence labeling, such as temporally limited observation due to photobleaching or the fact that the dye may alter the cells behaviour. In return, reliable detection and tracking of the cells in LFM can be more challenging due to the fluctuating shape of the cells appearance or the superposition of the observed interference patterns in areas of dense cell population. We address this task in Chapter 6.

suffer from effects such as photobleaching, which makes it is well-suited for monitoring experiments with cultured cells over an extended period of time. An example for a cell culture observed by LFM and its corresponding view with fluorescence microscopy is shown in Fig. 1.3. Due to the nature of LFM videos, reliable automatic methods are crucial for obtaining the clinically relevant information on the cell culture dynamics. This chapter's work aims at filling this niche and proposes a framework for processing LFM videos.

In particular, we propose a cell detector for LFM images based on fully convolutional residual networks [102]. Here, the fully convolutional neural network (FCN) is trained to regress a cell probability map constructed from cell center annotations. This approach is related to previous work on learning a distance transformation [57, 103]. We then formulate a probabilistic model for which the MLTP [86] becomes its MAP estimator and employ the efficient heuristics of Chapter 5 for solving it. As mentioned previously, the advantage of this joint clustering and tracking model is that it explicitly handles multiple detections, which is a frequent mistake due to

the nature of the cells appearance in LFM. Experiments on several hours of LFM videos indicate strong performance in terms of various detection and tracking scores.

1.3 Organization

This a publication-based thesis with the following structure: Chapter 1 introduced to the topic of graph inference in the setting of medical image analysis, and summarized our contributions. Chapter 2 gives a brief summary of relevant terminology and key concepts from graph theory, machine learning and integer linear programming, which are used throughout this manuscript.

Chapter 3 to 6 are composed of four publications [20, 104, 105, 21] in their original form. They have been published as journal articles or in peer-reviewed conference proceedings, and are therefore self-contained. Each of these chapters starts with a summary page, containing the full citation of the original publication, a short synopsis of the publications content and the thesis authors contributions. In order to improve the reader's experience, the text layout of the publications was harmonized and their bibliographies have been merged into one single bibliography at the end of this document.

Chapter 7 offers discussion and conclusions over the presented material and suggest directions for future work. Appendices A and B provide supplementary material to Chapter 4 and 5, respectively. Finally, a complete list of publications that have been written during the time period of this doctoral thesis can be found in Appendix C.

2

Background

This work draws from graph theory, combinatorial optimization and machine learning. This chapter aims at giving a concise summary of key concepts and notation used throughout this thesis, but it is not intended to be a representative overview of the most important concepts of each field. For a more complete and in-depth discussion on graph theory, we refer the interested reader [9, 10]. For a thorough discussion on combinatorial optimization and integer linear programs (ILPs), we refer to [106, 80, 62], and to [107, 108, 109, 110, 111] for topics of machine learning.

2.1 Graphs

A graph is defined as a pair $G = (V, E)$ where V is a set and $E = \{vw : v, w \in V\}$ is a set of pairs from V. We call an element $v \in V$ a *vertex* or *node* and an element $e = vw \in E$ an *edge* between vertex v and w. The graph is called *undirected* if the edges are unordered pairs and *directed* otherwise.

Subgraphs. A graph $H = (V', E')$ is called a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.

Paths and cycles. A sequence $P = (v_0, e_0, v_1, e_1, \ldots, e_{k-1}, v_k)$ of vertices $v_i \in V$ and connecting edges $e_i = v_i v_{i+1} \in E$ is called a v_0v_k-*path*. The vertices v_0 and v_k are called source and target. If the source and target vertices are identical $v_0 = v_k$, we call P a *cycle*.

Connectedness. A graph $G = (V, E)$ is called *connected* if there exists a path between any two nodes $v, w \in V$ in G. Otherwise, it is called *disconnected*. A subgraph G' of G that is connected and maximal is called *connected component* of G.

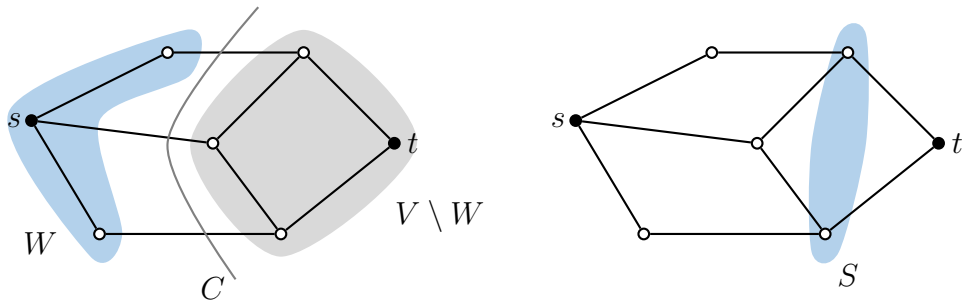


Figure 2.1: **Left**: Illustration of an st-*cut* C that partitions a graph $G = (V, E)$ into two subgraphs with vertex subsets W and $V \setminus W$, separating the vertices s and t. **Right**: A *vertex separator set* S that separates s from t. We note that the drawn S is the minimum (with respect to $|S|$) st-vertex separator set in the example graph G, while the drawn st-cut is not the minimum cut (with respect to $|C|$). However, both are minimal: removing any element from the drawn S or C would result in a loss of their st-separating property. This terminology will come up in Chapter 4.

Cuts. Given a decomposition of a graph $G = (V, E)$ into two subgraphs with $W \subseteq V$ and $U = V \setminus W$, we denote the set of edges $C = \{uw \in E : u \in U \wedge w \in W\}$ as *cut*. A cut C that intersects each vw-path in G is called vw-cut. Figure 2.1 illustrates a cut.

Vertex separator sets. A set of vertices $S \subseteq V$ is called a *vertex separator set* of v and w if each vw-path contains at least one vertex in S. Removing all vertices $s \in S$ thus leaves v and w disconnected. An example of a vertex separator set is given in Fig. 2.1

Trees, chains and forests. A graph $T = (V, E)$ is called a *tree* if it is cycle-free and has exactly $|E| = |V| - 1$. T is called a *chain* if it is a tree and each vertex $v \in V$ has at most two neighbouring nodes. A graph that is composed of a set of trees is called a forest.

Bipartite graphs. A graph $G = (V, E)$ is called *bipartite* if its vertices can be partitioned into two disjoint sets $V = W \cup U$ such that $\forall uw \in E : u \in U \wedge w \in W$, i.e. all edges connect a vertex of W to a vertex of U.

2.2 Integer Linear Programming

Integer linear programming (ILP) is a powerful tool to model discrete and combinatorial optimization problems. An integer linear program (also abbreviated with ILP) has the following standard form:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^n \quad . \end{aligned} \tag{2.1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ define the constraints and $\mathbf{c} \in \mathbb{R}^n$ are the coefficients of the linear objective function. Finally, the constraint $\mathbf{x} \in \mathbb{Z}^n$ requires integrality of the variables $\mathbf{x}$ and render it an ILP. If the integrality constraints are omitted (or more specifically, replaced with $\mathbf{x} \in \mathbb{R}^n$), then we obtain a standard-form linear program (LP). If only a subset of the variables have a requirement for integrality, then it is called a mixed integer linear program (MILP). A graphical representation of an ILP with two variables is given in Fig. 2.2. We call an $\mathbf{x}$ that satisfies all constraints *feasible*, and we denote the set of feasible $\mathbf{x}$ as Ω. For modeling combinatorial problems with ILPs, the subclass of 0-1-ILPs, where the variables are binary, i.e. $\mathbf{x} \in \{0,1\}^n$, is of particular interest.

Relaxations and bounds. Given an optimization problem $\min_{x \in \Omega} f(x)$, removing a subset of its constraints yields an optimization problem $\min_{x \in \Omega'} f(x)$ called *relaxation*. Since $\Omega \subseteq \Omega'$, the solution $\mathbf{x}'$ to the relaxed problem gives a *lower bound* to the original problem with optimal solution x^*, i.e. $f(\mathbf{x}') \leq f(\mathbf{x}^*)$. A natural relaxation for ILPs is the *LP-relaxation*, which simply omits the integrality constraints. Solving the LP-relaxation is a typical approach to obtain lower bounds for ILPs. Any feasible $\mathbf{x}'$ yields an *upper bound* $f(\mathbf{x}^*) \leq f(\mathbf{x}')$ to the optimization problem. A common way to obtain upper bounds for ILPs are rounding or search heuristics.

Gaps. Given a lower and upper bound f_{lower} and f_{upper}, we calculate the *gap* as $\alpha = \frac{|f_{\text{lower}} - f_{\text{upper}}|}{|f_{\text{upper}}|}$. The gap is used to certify how good a (preliminary) solution is (at least). Optimization algorithms typically report the tightest gap, i.e. the one between the best upper and lower bound. Reaching a predefined gap is a common termination criterion when solving ILPs.

Optimization algorithms. Solving general ILPs is NP-hard [112]. In contrast to this, solving LPs is easier. As a consequence, algorithms for solving generic ILPs rely heavily on *LP-relaxations*. Polynomial runtime

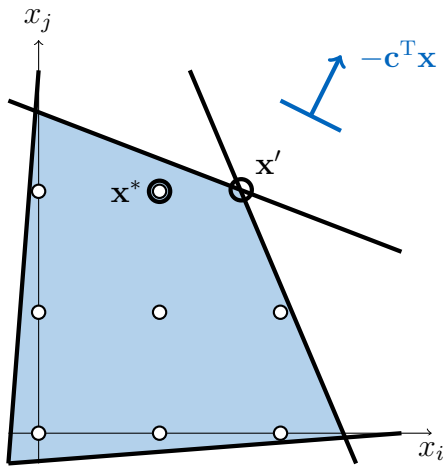


Figure 2.2: Illustration of an ILP with two variables $x_i, x_j \in \{0, 1\}$. The feasible set of integral solutions is depicted with a set of $\circ$, while the feasible set of its LP-relaxation is shown in blue. The boundaries of the inequalities $\mathbf{Ax} \leq \mathbf{b}$ are drawn as solid black lines. The objective function $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ is shown in dark blue. $\mathbf{x}^*$ is the optimal solution to the ILP, while $\mathbf{x}'$ is the solution to its LP-relaxation.

algorithms are known for general LPs, such as the interior point method [113] or the simplex method [114]. In practice, LPs are widely solved with the simplex method [114], even though it does not provide a worst-case polynomial runtime guarantee.

Two families of optimization algorithms for general ILPs are common: *Cutting plane* approaches and *branch-and-bound* approaches. The cutting plane method starts from a relaxation and then, refines the set of feasible points by adding additional inequalities, removing fractal points until an integral solution is obtained. Its use for ILPs goes back to [115], who introduced a procedure to generate a certain class of such inequalities – the so-called Gomory cuts. An illustration of a cutting plane is given in Fig. 2.3. A branch-and-bound algorithm builds a tree of subproblems to the ILP by splitting them into a set of K subproblems such that $\Omega = \bigcup_{k \in K} \Omega_k$ and thus $\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) = \min_{k \in K} (\min_{\mathbf{x} \in \Omega_k} f(\mathbf{x}))$. Determining which split to take is called *branching* step. An example for a split can be found in Fig. 2.3. By keeping track of lower and upper bounds across the subproblems, it is possible to discard subbranches of the problem tree if the

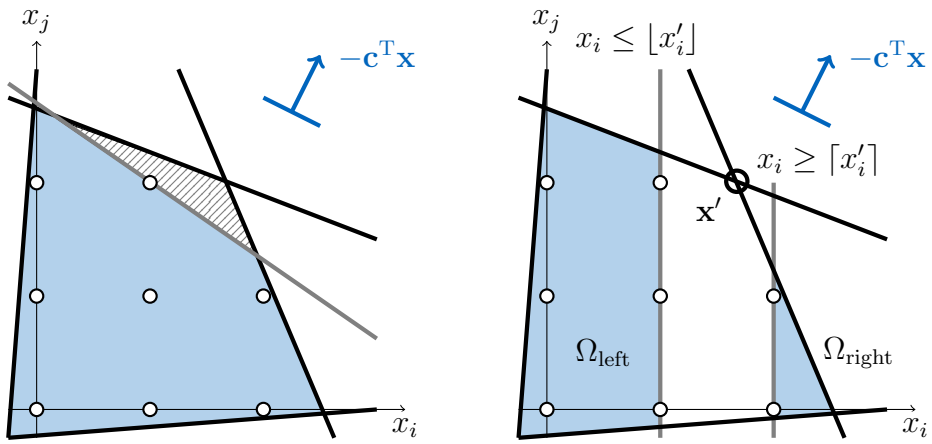


Figure 2.3: Two concepts utilized for solving ILPs illustrated on an ILP with two variables $x_i, x_j \in \{0, 1\}$. The feasible set of integral solutions is depicted with a set of $\circ$, while the feasible set of its LP-relaxation is shown in blue. **Left:** An example of a *cutting plane* drawn as bold gray line. It removes fractal solutions (ruled in fine gray lines) from the LP-relaxation. Cutting plane procedures aim to determine a series of such cutting planes, until the final relaxation is tight and yields an integral solution. **Right:** The same ILP after a split into two subproblems with the feasible sets Ω_{left} and Ω_{right} by the inequalities $x_i \leq \lfloor x_i' \rfloor$ and $x_i \geq \lceil x_i' \rceil$ respectively. $\mathbf{x}'$ is the solution to the previous LP-relaxation where x_i' had a fractal value. Clearly, no integral solution was removed and thus, one of the new subproblems has the same optimal solution as the original ILP. Determining such splits is done in the *branching* step in branch-and-bound algorithms.

current best upper bound is better than the lower bound of the particular subproblem. This is the *bounding* step. Branch-and-bound methods for ILPs were introduced in [116]. Modern off-the-shelf solvers like Gurobi [117] or CPLEX [118] typically implement hybrids of branch-and-bound and cutting plane algorithms, sometimes called *branch-and-cut*.

Lazy constraint generation. In the ILP formulation of several combinatorial problems, the set of constraints that define all feasible solutions can be exponentially large. Examples include the subtour-elimination constraints for the traveling salesman problem (TSP) [119], cycle-consistency constraints in the MCMCP [120, 87], or the connectedness constraints in the

MCCSP [64, 104]. The observation is, however, that for a given instance, only a subset of these constraints is active at its optimum. This gives raise to a strategy we call *lazy constraint generation*: Starting with a relaxation of the problem without any of the (exponentially-sized) subset of constraints, it is solved and the outcome checked for violations of the left-out constraints. If any are found, (some of) the violated constraints are added and the optimization continues. If the current solution does not violate any constraints, we have obtained a feasible solution. This strategy, of course, relies on being able to separate violated constraints for a given intermediate solution with polynomial complexity.

2.3 Machine Learning

Machine learning is concerned with algorithms that *learn* a mapping, for example a classification or a regression function, from a dataset of observations. The field has a strong overlap with pattern recognition, data mining, artificial intelligence or statistical learning theory. Well-known learning algorithms are the support vector machine (SVM) [121], the random forest (RF) [122] or the neural network (NN) [123, 124] and its more recent variants that employ the concepts of deep learning [125, 126, 127, 102, 111]. In the recent years, machine learning techniques have strongly influenced progress in several fields. Especially in medical image analysis, machine learning methods find wide application. For example, the majority of the top-performing methods in recent challenges for segmenting brain tumors [128], liver tumors [129] or stroke lesion [130] integrate a learning scheme at some stage of the processing pipeline.

From a high-level perspective, a machine learning algorithm can be seen as the combination of the following components: a class of hypotheses $\mathcal{H}$, a loss function $\mathcal{L}(\mathcal{X}, h)$ and an optimization scheme to determine $\hat{h} \in \mathcal{H}$ such that it minimizes the loss $\mathcal{L}$ (approximately) for a given dataset $\mathcal{X}$. For instance, a possible hypothesis class $\mathcal{H}$ for a binary classification is the set of all hyperplanes in the input space, in which case an appropriate loss $\mathcal{L}$ would be one that is minimized by a hyperplane that separates the samples of both classes as well as possible. Loss and hypotheses have to be chosen appropriately to the underlying task (classification problems require different losses than regression or clustering problems), and an optimization scheme needs to be appropriate for both $\mathcal{L}$ and $\mathcal{H}$. Properties of the training

set $\mathcal{X}$ have to be considered to avoid issues with *over-* or *underfitting*. In practice, selecting one of the established machine learning algorithms implicitly sets (at least to a certain degree) the choice of hypothesis class $\mathcal{H}$, loss $\mathcal{L}$ and optimization scheme, and can then be fine-tuned by adjusting its hyperparameters. For example, adding or removing layers in the architecture of a convolutional neural network (CNN) effectively changes the hypothesis class $\mathcal{H}$. What is common to almost all algorithms is the distinction of two phases: the *learning* and the *prediction* phase.

Learning. Given a training set $\mathcal{X}$, determine the best hypothesis $h \in \mathcal{H}$ with respect to $\mathcal{L}$:

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \mathcal{L}\left(h, \mathcal{X}\right) \quad . \tag{2.2}$$

This is sometimes also referred to as *training*, *fitting*, or *parameter inference* in the context of probabilistic models. The resulting $\hat{h}$ is the *trained* estimator. The loss $\mathcal{L}(h, \mathcal{X})$ might be difficult to optimize. Thus, several algorithms rely on approximate or heuristic optimization schemes and will therefore not guarantee to find the exact optimum of (2.2). Amongst the most prominent examples are deep NNs, where optimization is done with stochastic gradient descent (SGD). Ultimately, we are not just interested in how well the learned model $\hat{h}$ fits the training data $\mathcal{X}$, but mainly how well it performs on *unseen* data, which was not available during the learning phase, i.e. how well it *generalizes*. Hence, it is recommended to split the available dataset $\mathcal{X}$ into a training and test set, where the former is used to learn $\hat{h}$ and the latter is used to estimate the performance of $\hat{h}$ on new data. Widely used variants of this scheme are cross-validation and bootstrapping [131].

Prediction. Given a (fitted) model $\hat{h}$ and an observation $\mathbf{f}$, we can predict the associated target x:

$$\hat{x} = \hat{h}(\mathbf{f}) \quad . \tag{2.3}$$

This step is also called *inference* or *estimation*. If x is categorial, we call it a classification and if it is continuous a regression. This step is typically defined by the considered model classes $\mathcal{H}$ during training and usually fast to evaluate. For example, for a linear SVM, this step is just comprised of an inner product and a thresholding operation. The target x can even be a complex structure composed of several inter-dependent variables, e.g. a labeling of all pixels in an image. In this case, the prediction step itself can become a (non-trivial) optimization problem [110]. This is common

in the context of computer vision, where a lot of work has focussed on such structured prediction problems and its optimization during prediction time [132, 133]. In fact, *graph inference* as we consider it in this thesis is such a structured prediction.

3

Reconstructing Cerebrovascular Networks under Local Physiological Constraints by Integer Programming

This chapter has been published as **peer-reviewed journal paper**:

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." In: *Medical Image Analysis* 25.1 (2015), pp. 86–94. DOI: 10.1016/j.media.2015.03.008

Synopsis: This work presents an approach to reconstructing vascular networks by considering image evidence, connectedness and geometric relation between vessels. It extends our previous work [134] by a) a probabilistic model for which we identify an ILP as its MAP estimator, and b) additional experiments and discussion.

Contributions of thesis author: Model development and implementation, computational experiments, composition of manuscript.

Abstract

We introduce a probabilistic approach to vessel network extraction that enforces physiological constraints on the vessel structure. The method accounts for both image evidence and geometric relationships between vessels by solving an integer program, which is shown to yield the MAP estimate to a probabilistic model. Starting from an overconnected network, it is pruning vessel stumps and spurious connections by evaluating the local geometry and the global connectivity of the graph. We utilize a high-resolution µCT dataset of a cerebrovascular corrosion cast to obtain a reference network and learn the prior distributions of our probabilistic model and we perform experiments on in-vivo µMRA images of mouse brains. We finally discuss properties of the networks obtained under different tracking and pruning approaches.

3.1 Introduction

Many diseases affect general properties of the cerebrovascular network, examples are arteriosclerosis and dilative vascular malformations changing vessel shape and diameter, but also Alzheimer's and related neuro-degenerative diseases are suspected to affect the general vascularity and global network properties [41, 42]. Studies investigating such diseases frequently use mouse models for experiments and commonly acquire in-vivo cerebrovascular imagery by means of µMRA. While segmenting and tracing tubular structures is a longstanding field of interest in medical image computing [44, 45, 46, 48], we approach here the wider – and somewhat neglected [59] – problem of extracting the full vascular network from image volumes under consideration of local geometric properties and global constraints of the vascular structure.

Most vessel segmentation techniques rely on tubularity measures or other vessel enhancement filters [44], and then apply rule-based or learned decision algorithms to segment the vessels [46, 48, 50]. The network graph - representing vessels by their centerline, complemented with additional information such as local radii – can be extracted from binary segmentations using morphological operators [135, 136], or by tracking vessels directly by minimal path techniques [43], e.g. by applying a fast marching algorithm

[49] or a Dijkstra-like scheme [47]. We point the interested reader to [46, 48] for more extensive reviews. In most applications, however, the extracted graphs need further post-processing: Lu et al. [137], for example, incorporated discriminative classifiers that examine local geometrical features of segments into a hierarchical approach for vessel-structure parsing. In order to deal with imperfections in vascular connectivity of extracted networks, Kaufhold et al. [138] discussed a supervised learning approach to gap filling and network pruning, whereas Schneider et al. [139] recently proposed a generative method for gap in-fill that is guided by a simplified angiogenesis model. While segmentation algorithms are likely to enforce expected local vessel shape and geometry, only few approaches consider both local properties and global network connectivity when extracting the full network: Jiang et al. [60] incorporated assumptions about vessel diameters (Murray's hypothesis [140]) in a global optimization problem restricted to vascular trees. Tree shape priors have also been included into the segmentation of vasculature by [66]. In a different application, Türetken et al. [141] introduced recently an integer programming approach that evaluates path coherence and connectivity of general curvilinear structures, such as streets in remote-sensing images or vessels in confocal image stacks. Starting from an overconnected graph, they are pruning edges that do not fulfill desired structural relationships of neighbouring segment pairs using a path classifier that is trained from annotated 3-D networks.

All of these approaches enforce local coherence within the extracted network – a general property of the vascular network. More complex local properties of a structural network, however, can be described by network *motifs* [16, 19]. Network motifs are frequently recurring subgraphs, also called building blocks, that are characteristic for a type of network, such as bifurcations in vascular networks.

In this paper, we enforce local geometrical properties similar to Jiang et al. [60], exploring the relevance of two basic motifs of vascular networks, i.e., the geometrical properties of continuing segment pairs and of vessel bifurcations and following the idea of pruning of Türetken et al. [141]. We present a probabilistic model which combines this geometric prior with local vessel evidence obtained from a segmentation algorithm [50], and show that the MAP estimate can be computed by an ILP. We learn the global statistic of geometrical properties of the network motifs from a high resolution dataset. Finally, we identify a more efficient scheme to solve the ILP for large

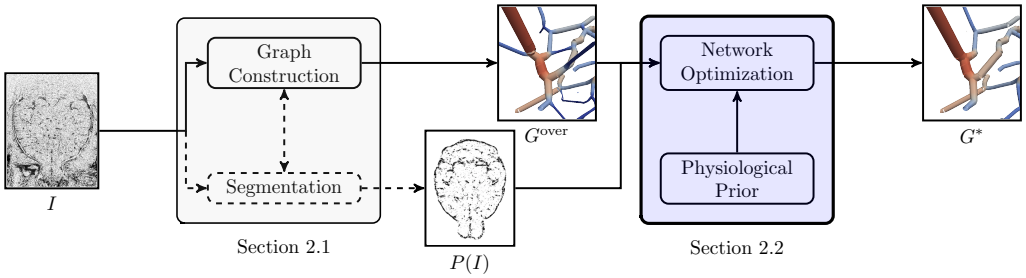


Figure 3.1: Workflow: In a first stage (gray box), the image volume I is processed so as to obtain an overconnected graph G^{over} as well as a confidence measure for vessels such as the confidence map $P(I)$. In the following step, the network G^* is extracted from G^{over} in an optimization scheme that considers both image evidence (according to $P(I)$) and geometric-physiological prior knowledge. In this paper, we focus on the network optimization step (blue), where both image evidence and geometrical relationships of certain network motifs, namely continuing pairs and bifurcations are considered.

datasets and illustrate its application for reconstructing vascular networks from in-vivo µMRA images of the mouse brain.

3.2 Methods

In this section, we detail on the proposed vessel network extraction method that estimates the most probable network under consideration of image evidence and physiological prior knowledge. As depicted in the workflow (Fig. 3.1), this method starts from an overconnected network graph G^{over}. Hence, we briefly review the applied segmentation framework and skeletonization method as used in our experiments.

3.2.1 Vessel Segmentation Method and Construction of the Overconnected Graph

As a first stage, we transform image intensities into confidence maps by using the framework of Schneider et al. [142, 50]: In this approach, multiscale steerable filter templates (SFT) are used as efficient directional filters, offering features that are invariant with respect to the local vessel direction. An

oblique random forest (RF) [143], which determines splits by solving a linear regression with elastic net penalty in each node, is used for a subsequent classification. The RF assigns each voxel v in an image volume I to a probability $p_v \in [0, 1]$, indicating the local presence of a vessel-like structure.

We apply a threshold θ to the probability volume $P(I)$ and skeletonize the resulting binary volume using distance-ordered homotopic thinning (DOHT) [135], a method that iteratively removes voxels without altering the objects topology, to derive a network graph $G(\theta)$. We obtain an overconnected network by generating multiple binary segmentations from $P(I)$ with different thresholds $\{\theta_i\}$, skeletonizing each of them by DOHT to $G(\theta_i)$ and superposing them into one network $G^{\text{over}}(\{\theta_i\})$. The resulting network contains both segments with low confidence (contributed by graphs from low thresholds θ close to 0), but maintains the high spatial accuracy of a graph that is generated from conservative thresholds (i.e., with θ close to 1). Note, however, that any method which generates an overconnected graph G^{over} by proposing local vessel connections could be used instead.

3.2.2 Vessel Network Extraction

The goal of our method is to find the most plausible network G^* out of an overconnected network graph $G^{\text{over}} = (V, E)$ with edges $E = \{e_i\}$ and given image evidence $P(I)$. We encode subgraphs of G^{over} with a set of binary variables $X = \{x_i\}$ where each x_i indicates whether or not the corresponding segment $e_i \in E$ is active (i.e. $x_i = 1$). Therefore, we arrive at the equivalent problem of determining the MAP estimate of $\mathbf{x} \in \{0, 1\}^{|E|}$, for which we describe a probabilistic model (Sect. 3.2.2) that considers image evidence, local properties of specific network motifs as well as global connectivity, and derive an ILP that allows computing the MAP network (Sect. 3.2.2).

Probabilistic Model

We formulate a probabilistic model $P(\mathbf{X} = \mathbf{x}, \Omega | I, G)$ according to Fig. 6.2, where I is the image evidence, G is the given (overconnected) graph and X is the set of binary variables denoting subgraphs of G. Ω is the set of all feasible solutions of $\mathbf{x}$:

$$\Omega = \{\mathbf{x} \in \{0, 1\}^{|E|} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\} \ , \tag{3.1}$$

$$I, G \longrightarrow X \longrightarrow \Omega$$

Figure 3.2: Probabilistic model. I: Image; G: (overconnected) Graph; X: Set of binary variables denoting subgraphs of G; Ω: Set of feasible configurations of $\mathbf{x}$.

with $\mathbf{Ax} \geq \mathbf{b}$ being the short notation for all hard constraints that will be considered such as those enforcing connectivity. This introduces a probabilistic interpretation, as in [144], of the hard constraints that we impose on the extracted networks.

According to the given probabilistic model in Fig. 6.2, we arrive at the posterior distribution for $\mathbf{x}$:

$$P\left(\mathbf{X} = \mathbf{x} | I, G, \Omega\right) \propto P\left(\Omega | \mathbf{X} = \mathbf{x}\right) P\left(\mathbf{X} = \mathbf{x} | I, G\right) \quad . \tag{3.2}$$

Next, we model $P(\mathbf{X} = \mathbf{x}|I, G)$ as a Markov random field (MRF):

$$P(\mathbf{X} = \mathbf{x}|I, G) = \frac{1}{Z} \prod_{x_i \in X} \phi_i\left(x_i; I, G\right)^{\alpha} \prod_{\substack{x_i, x_j \in X: \\ e_i, e_j \text{ adjacent} \\ \nexists e_k \text{ adjacent to } e_i \wedge e_j}} \phi_{i,j}\left(x_i, x_j; I, G\right) \prod_{\substack{x_i, x_j, x_k \in X: \\ e_i, e_j, e_k \text{ adjacent}}} \phi_{i,j,k}\left(x_i, x_j, x_k; I, G\right) \quad , \tag{3.3}$$

where Z is the partition function and $\phi\left(.\right)$ are the potentials, which are defined in the following. $\alpha > 0$ is a parameter to adapt the trade-off between unary and higher order potentials. For each segment represented by x_i, we set the unary potential:

$$\phi_i\left(x_i; I, G\right) = \begin{cases} P\left(x_i = 1 | I_i, E_i\right) & \text{if } \; x_i = 1 \quad , \\ P\left(x_i = 0 | I_i, E_i\right) & \text{otherwise} \quad , \end{cases} \tag{3.4}$$

where $P\left(x_i = 1 | I_i, E_i\right)$ can be understood as image evidence that the segment x_i is part of the underlying vasculature and a valid segment of the network. The higher-order potentials are chosen as:

$$\phi_{i,j}\left(x_i, x_j; I, G\right) = \begin{cases} p_{C,ij} & \text{if } \; x_i x_j = 1 \quad , \\ p_T & \text{otherwise} \quad , \end{cases} \tag{3.5}$$

and

$$\phi_{i,j,k}\left(x_i, x_j, x_k; I, G\right) = \begin{cases} p_{B,ijk} & \text{if } \; x_i x_j x_k = 1 \quad , \\ p_{C,ij} & \text{if } \; x_i x_j = 1 \wedge x_i x_j x_k = 0 \quad , \\ p_T & \text{otherwise} \quad . \end{cases} \tag{3.6}$$

where $p_{C,ij}$ is the likelihood of e_i continuing in e_j, $p_{B,ijk}$ the likelihood of a bifurcation involving e_i, e_j and e_k, and p_T represents the possibility that neither of them occur and the vessel terminates. Note that both binary and ternary potentials in (3.3) account for the relationships of multiple segments, hence we can split $P(\mathbf{X} = \mathbf{x}|I, G)$ into a pure image evidence term – containing only the unary potentials $\phi_i(.)$ – and a prior term consisting of both $\phi_{i,j}(.)$ and $\phi_{i,j,k}(.)$.

Maximum A Posteriori Estimation by Integer Programming.

So far, we translated the problem of finding the most plausible subnetwork in G^{over} into determining the MAP estimate $\mathbf{x}^*$ of (3.2):

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} P(\mathbf{X} = \mathbf{x}|I, G, \Omega) = \arg\max_{\mathbf{x}\in\mathcal{X}} P(\Omega|\mathbf{X} = \mathbf{x})\, P(\mathbf{X} = \mathbf{x}|I, G)\,, \tag{3.7}$$

where $\mathcal{X} = \{0, 1\}^{|E|}$ is the set of all configurations of $\mathbf{x}$. In the following, we are going to derive an ILP to determine $\mathbf{x}^*$. We start by specifying the likelihood $P(\Omega|\mathbf{X} = \mathbf{x})$ to be equal for all feasible $\mathbf{x}$ and 0 else, i.e.

$$P(\Omega|\mathbf{X} = \mathbf{x}) \propto \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega\ , \\ 0 & \text{otherwise}\ . \end{cases} \tag{3.8}$$

Applying this definition leads to:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} P(\Omega|\mathbf{X} = \mathbf{x})\, P(\mathbf{X} = \mathbf{x}|I, G) \tag{3.9}$$

$$= \arg\max_{\mathbf{x}\in\Omega} P(\mathbf{X} = \mathbf{x}|I, G)\ . \tag{3.10}$$

From (6.3), the definition of the MRF (3.3), its potentials (3.4) to (3.6) and the fact that each pseudo-boolean function has a unique multilinear polynomial form, it follows that the MAP estimate $\mathbf{x}^*$ takes the form of the integer program:

$$\min_{\mathbf{x}} \quad J(\mathbf{x}) = \alpha \sum_{x_i\in X} w_i x_i + \sum_{\substack{x_i,x_j\in X:\\ e_i,e_j \text{ adjacent}}} w_{ij} x_i x_j + \sum_{\substack{x_i,x_j,x_k\in X:\\ e_i,e_j,e_k \text{ adjacent}}} w_{ijk} x_i x_j x_k\ , \tag{3.11}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}\ , \tag{3.12}$$

$$x_i \in \{0, 1\} \qquad \forall x_i \in X\ , \tag{3.13}$$

with the weights w_i, w_{ij} and w_{ijk} derived as

$$w_i = -\log \frac{P\left(x_i = 1 | I_i, E_i\right)}{P\left(x_i = 0 | I_i, E_i\right)} \ , \tag{3.14}$$

$$w_{ij} = -\log \frac{p_{C,ij}}{p_T} \ , \tag{3.15}$$

$$w_{ijk} = -\log \frac{p_{B,ijk} p_T^2}{p_{C,ij} p_{C,ik} p_{C,jk}} \ . \tag{3.16}$$

The derivation is given in the appendix.

Computing the Weights

Image Evidence The weights derived from the unary potentials of the MRF allow us to account for image evidence observed for each segment represented by x_i separately. To infer $P\left(x_i = 1 | I_i, E_i\right)$ from the image evidence, we average the probabilistic output p_v of the classification along voxels v assigned to the segment e_i. We define $P\left(x_i = 1 | I_i, E_i\right) = p_i$ and compute:

$$w_i = -\log \frac{p_i}{1 - p_i} \ . \tag{3.17}$$

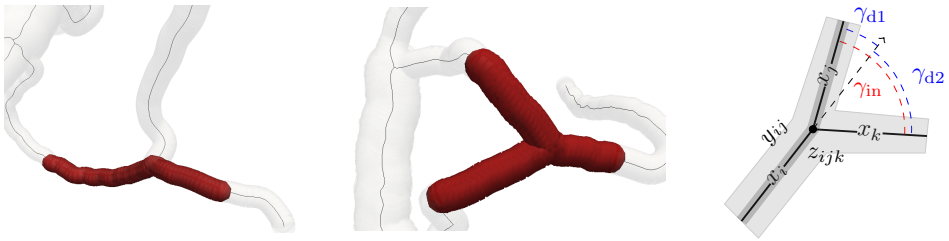
Geometric Prior As depicted in Fig. 3.3, we consider two network motifs and weight them accordingly with the derived w_{ij} and w_{ijk}. In order to compute these, we evaluate angles between the involved segments – denoted with γ_{ij} and γ_{ijk} (cf. Fig. 3.3c) – under consideration of their estimated radii and define:

$$w_{ij} = -\log \frac{p_{C,ij}}{p_T} = -\log \frac{P\left(\gamma_{ij} | \text{continue}, \theta\right) P\left(\text{continue} | \theta\right)}{P\left(\gamma_{ij} | \text{terminate}, \theta\right) P\left(\text{terminate} | \theta\right)} \ , \tag{3.18}$$

$$w_{ijk} = -\log \frac{p_{B,ijk} p_T^2}{p_{C,ij} p_{C,ik} p_{C,jk}} \tag{3.19}$$

$$= -\log \frac{P\left(\gamma_{ijk} | \text{branch}, \theta\right) P\left(\text{branch} | \theta\right) P\left(\text{terminate} | \theta\right)^2}{\prod_{(i',j') \in 2^{\{i,j,k\}}} P\left(\gamma_{i'j'} | \text{continue}, \theta\right) P\left(\text{continue} | \theta\right)} \ , \tag{3.20}$$

where θ is the parametric model that encodes physiologically realistic geometric properties of the network motifs in terms of distributions over the considered geometric features γ_{ij} and γ_{ijk}. The probabilities $p_{C,ij}, p_{B,ijk}$



(a) (b) (c)

Figure 3.3: Illustration of network motifs that are considered by the physiological model and the variables at a potential bifurcation. a) Pairs of continuing segments and b) triplets that form a bifurcation (illustrated in red). c) Variables at a potential bifurcation: Segments, e.g. x_i, are drawn as solid black lines, while a pairwise variable y_{ij} that represents x_i continuing in x_j is depicted by the dark grey overlay (y_{ik} and y_{jk} are omitted for clarity). z_{ijk} corresponds to all three adjacent edges in the bifurcation (light grey). Furthermore, both deviation angles γ_{d} (blue) and the inner angle γ_{in} (red) are shown. Radii are estimated perpendicular to the edge direction. In our experiments, we consider geometric features $\gamma_{ijk} = (\gamma_{\mathrm{in}}, \gamma_{\mathrm{d1}}, \gamma_{\mathrm{d2}})_{ijk}$ for bifurcations and $\gamma_{ij} = (\gamma_{\mathrm{d}})_{ij}$ for continuing segments, while radius estimates are used to determine the main trunk.

and p_T originate from the definitions in (3.5) and (3.6). In this case, $P(\gamma_{ij}|\mathrm{continue}, \theta)$ describes the likelihood of observing angle γ_{ij} in a continuing pair, and $P(\mathrm{continue}|\theta)$ is the prior on how frequent continuing pairs occur. We will discuss the choice of such a model θ in our experiments (Sect. 3.3.3) where we fit them to evidence from high-resolution network data.

Global Connectivity

An essential aspect when extracting vascular networks from noisy or incomplete data is to enforce connectivity between the observed components in the network. In our approach, we enforce this property by hard constraints:

$$\sum_{x_i \in M} x_i < |M| + \sum_{x_j \in N} x_j \qquad \forall M \subset X \setminus x_{\mathrm{seed}} \ , \tag{3.21}$$

where M is a set of connected segments and N its neighbourhood. In other words, a subset M of segments that form one connected structure

can only be active, if there is an incoming segment (or it is adjacent to the seed). As there are exponentially many constraints, we follow a lazy constraint generation approach and iteratively add those which are required (cf. Sect. 3.2.2).

Linear Formulation

We note that the integer program in (3.11) contains second and third order relations between variables. To deal with these, we exploit the binary nature of the variables x_i and introduce additional auxiliary variables $Y = \{y_{ij}\}$ and $Z = \{z_{ijk}\}$ to substitute these products ($x_i x_j$ and $x_i x_j x_k$) in the objective. A set of linear constraints ties the auxiliary variables to the corresponding indicator variables $\{x_i\}$ such that $y_{ij} = x_i x_j$ and $z_{ijk} = x_i x_j x_k$ holds for all feasible solutions:

$$y_{ij} \leq x \qquad \forall x \in \{x_i, x_j\}, \forall y_{ij} \in Y \ , \tag{3.22}$$

$$y_{ij} \geq x_i + x_j - 1 \qquad \forall y_{ij} \in Y \ , \tag{3.23}$$

$$z_{ijk} \leq x \qquad \forall x \in \{x_i, x_j, x_k\}, \forall z_{ijk} \in Z \ , \tag{3.24}$$

$$z_{ijk} \geq x_i + x_j + x_k - 2 \qquad \forall z_{ijk} \in Z \ , \tag{3.25}$$

which leads to the ILP:

$$J(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \alpha \sum_{x_i \in X} w_i x_i + \sum_{y_{ij} \in Y} w_{ij} y_{ij} + \sum_{z_{ijk} \in Z} w_{ijk} z_{ijk} \ , \tag{3.26}$$

$$\text{s.t.} \quad \mathbf{A}'(\mathbf{x}, \mathbf{y}, \mathbf{z}) \geq \mathbf{b}' \qquad , \tag{3.27}$$

$$x_i \in \{0, 1\} \qquad \forall x_i \in X \ , \tag{3.28}$$

$$y_{ij} \in \{0, 1\} \qquad \forall y_{ij} \in Y \ , \tag{3.29}$$

$$z_{ijk} \in \{0, 1\} \qquad \forall z_{ijk} \in Z \ , \tag{3.30}$$

where we summarized the constraints (3.22) to (3.25) together with (3.12) in (3.27). $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is the concatenation of all binary variables to a column vector of all variables. We can solve the linear problem of (3.26) by a branch and cut algorithm implemented in libraries such as [118].

Solving the Integer Programming Problem for Large Datasets

The described integer programming problem of (3.26) with the associated constraints grows quickly for large graphs that result from whole brain scans.

To tackle this problem, we propose to employ a lazy constraint generation scheme together with the following approach:

1. Given the variable set X of the ILP, define a graph $A = (V_A, E_A)$ with a vertex $v_i \in V_A$ for every $x_i \in X$.

2. Add an edge e_{ij} to E_A if and only if there exists a constraint that contains both variables x_i and x_j.

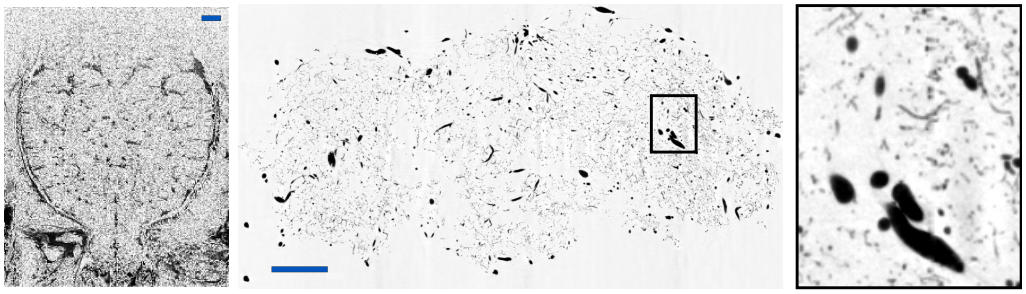
3. Determine the connected components in A. Vertices of each connected component represent a sub-problem that can be solved independently – of course with their according constraints.

Whenever a constraint or variable is added, we adjust the graph A dynamically. Then only sub-problems – i.e. variable sets represented by connected components of A that are affected by the change – need to be solved, speeding up the performance over approaches that solve the complete problem in every run. Lazy constraint generation is a known concept in integer programming, which was, for example, already applied to the famous travelling salesman problem [145], whereas the dynamic sub-problem handling is novelty of our approach. Employing this scheme does not deteriorate the solutions quality, that is its objective value in terms of the cost function $J(.)$ in (3.11). However, the solution $\mathbf{x}^*$ is in general not unique and hence, solutions obtained from different optimizers are not necessarily identical in terms of the network that they encode.

3.3 Experiments

3.3.1 Image Data

We use four 3-D in-vivo µMRA images of the mouse brain, each of size 248 px × 248 px × 109 px with an isotropic voxel spacing of 60 µm, and a µCT of a corrosion cast from the cerebral vasculature of a mouse brain with a volume of 2048 px × 2048 px × 3714 px and a spacing of 2.9 µm. Both image data types are depicted in Fig. 3.4. All five datasets were acquired from different animals with one acquisition protocol for all µMRA images. The µCT is downsampled by a factor of 2 for the subsequent steps.



(a) (b)

Figure 3.4: (a) Example slice of a whole-brain µMRA dataset, (b) Central slice of the corrosion cast µCT with a magnified subregion (right). All images are gray-scale inverted and the blue scalebar is 1 mm.

3.3.2 Preprocessing: Vessel Segmentation and Graph Construction

We use the described vessel segmentation framework [142, 50] to obtain an initial segmentation and to construct network graphs. Its parameters, such as SFT order and scales as well as RF parameters, are adjusted in a leave-one-out cross-validation using manually annotated ground truth labels. Probability maps $P(I)$ are binarized for different thresholds θ and transformed into network graphs using DOHT as discussed in Sect 3.2.1. We segment and track vessels in both µMRA and the µCT volume. The non-overconnected results, obtained from single thresholds $\{0.2, 0.5, 0.9\}$, – denoted as *alternative* below – serve as comparison in the experiments.

3.3.3 Training: Learning the Geometric Prior from the High Resolution Network

We use the geometrical prior to support bifurcations in our overconnected graph that are valid with respect to their diameter and relative angle, and to remove those that are not. To this end, we learn the relative frequencies of radii and deviation angles of vessel segments from the high resolution µCT (see Fig. 3.5b for observed angles). We find $p(\gamma_{ij}|\text{continue}, \boldsymbol{\Theta})$ to be well represented by an exponential distribution (where γ_{ij} is the deviation angle between two continuing vessels), and $p(\gamma_{ijk}|\text{branch}, \boldsymbol{\Theta})$ to be well approximated by a multivariate Gaussian (where γ_{ijk} are the three angles of

a bifurcation) while $\boldsymbol{\Theta}$ is the joint set of parameters of the two distribution models. Radius estimates are utilized to determine the main trunk in a bifurcation that serves as the reference for the angles considered here. An illustration of the angles calculated at every possible bifurcation is given in Fig. 3.3c. Furthermore, we estimate the relative frequencies of the discussed network motifs, $P\,(\text{continue}|\theta)$, $P\,(\text{branch}|\theta)$ and $P\,(\text{terminate}|\theta)$. For this, we take into account that paths in the high-resolution data (n-times higher spatial resolution) are more frequently sampled due to the higher spatial resolution, while the number of bifurcation points remains constant. Hence, for estimating $P\,(\text{continue}|\theta)$, we normalize the number of points sampled along paths by the resolution ratio n between high and low resolution datasets. Otherwise, $P\,(\text{continue}|\theta)$ would be strongly overestimated for the low-resolution data. Parameters θ are fitted to the distributions observed in the µCT using the maximum likelihood estimate. The fitted model is then used to determine weights w_{ij} and w_{ijk} as in (3.18) and (3.20). Samples of favourable network motifs according to the learned distributions are depicted in red in Fig. 3.5c.

3.3.4 Application: Extracting Networks from Low Resolution µMRA Data

We generate an overconnected graph for each of the µMRA test sets using the approach described in Sect. 3.2.1 with multiple thresholds $\theta_i \in \{0.2, 0.5, 0.9\}$. We select these three thresholds to obtain connections from high-recall ($\theta = 0.2$), high-precision ($\theta = 0.9$) and trade-off ($\theta = 0.5$) binarizations. After optimization (with $\alpha = 1$ and a maximal $\alpha_{\max} = \alpha \to \infty$), we compare the extracted network (opt) to networks obtained from individual thresholds $\theta = 0.2, 0.5$ and 0.9 (alternative).

We present a sensitivity analysis of the α-parameter of our method in Fig. 6.4. For large values of α, the image evidence is emphasized (the network is still connected), while choosing a small value of α prioritizes geometric weights. Although varying α has limited impact on the scores that we calculate for our test data, it is recommendable for other problems to determine the most suitable α depending on the previously applied processing, namely the segmentation and overconnection stage. If the data to be processed is expected to contain malformations that are known to form atypical bifurcations, then it would not be advisable to choose $\alpha < 1$, unless

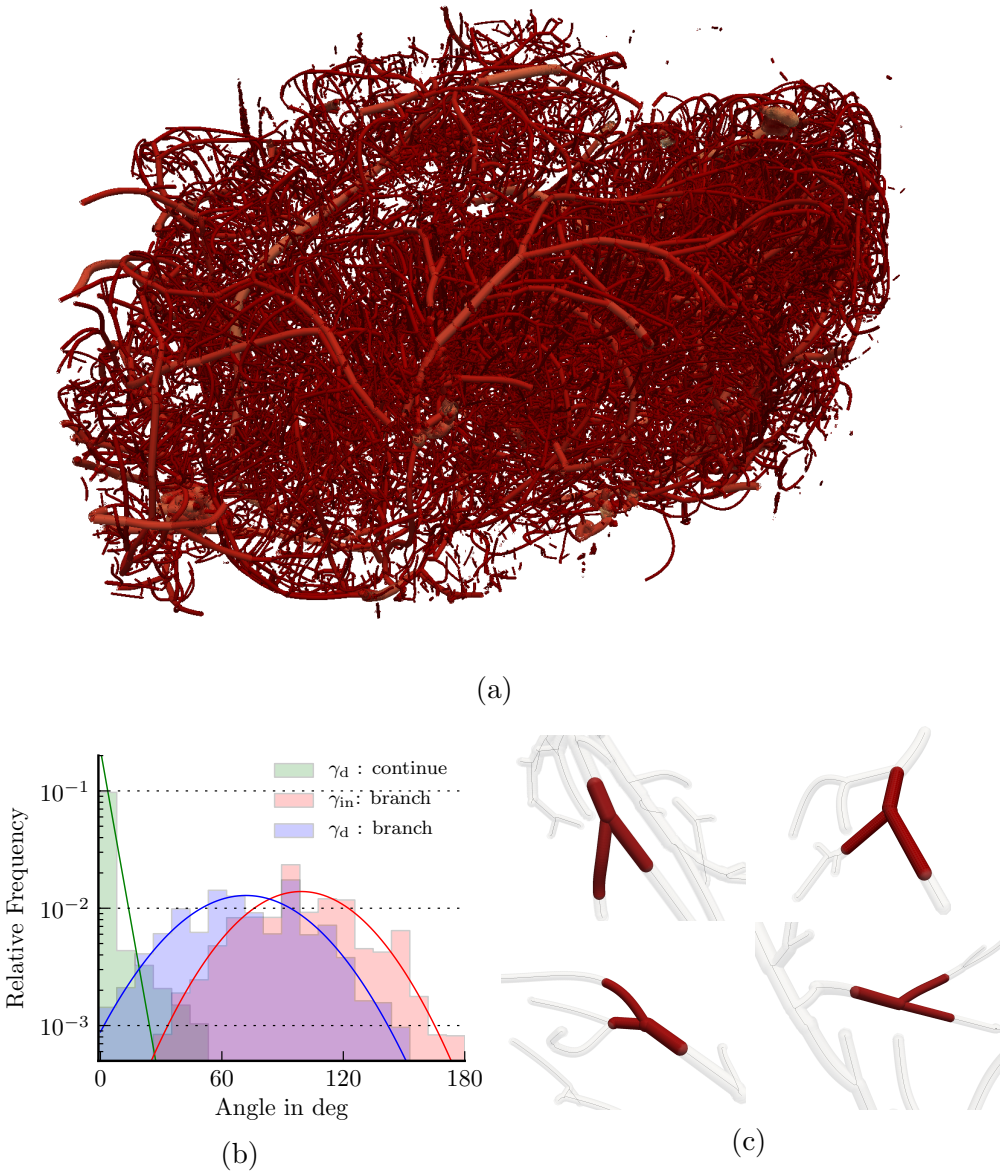


Figure 3.5: (a) Processed network of the corrosion cast µCT; see Fig. 3.4 for scale. (b) Angle histograms computed on the extracted network of the corrosion cast dataset with inferred distributions (solid). Shown are deviation angles of continuing segments (green), deviation (blue) and inner (red) bifurcation angles (see Fig. 3.3c). (c) Favourable configurations of bifurcations (red) according to the fitted model of the corrosion cast µCT.

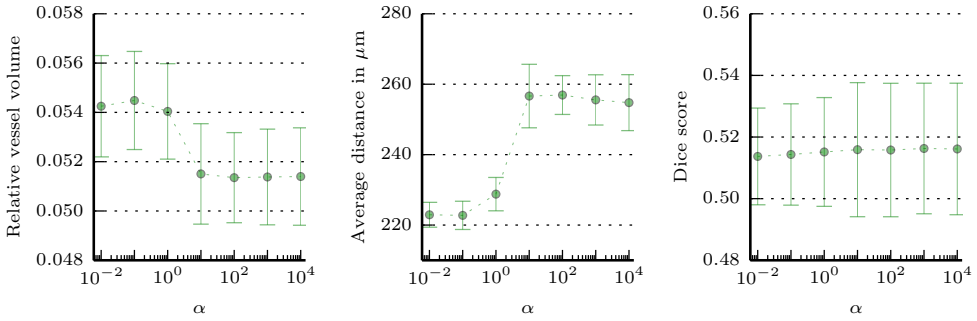


Figure 3.6: Sensitivity study on the weight parameter α performed on two µMRA datasets. The parameter α has a limited impact with respect to the depicted scores. Averages are within 5.13 % and 5.45 % for relative vessel volume, 222 µm and 256 µm average distance and 51.4 % and 51.6 % Dice (cf. Fig. 3.8).

the prior model had been adapted accordingly. In the following experiments, we will set $\alpha = 1$ per default and compute results with α_{max} – i.e. maximum emphasis on the image evidence term – for comparison.

Figure 4.5a shows an optimized network, while Fig. 4.5b-c provides close-up views before and after optimization, respectively. A number of spurious sprouts and loops are visible that are removed during the process. Details in Fig. 4.5d-g show differences between the two thresholded and an optimized network. We find that both thresholded networks (Fig. 4.5e-f) lose the connection of the large branch (center to top-right), while it is retained in the optimized network (Fig. 4.5g).

Figure 3.8 reports quantitative measures of global network properties, such as relative vessel volume, average distance to the next vessel (i.e. extravascular distance) and Dice score. We calculate both relative vessel volume and the extravascular distance over a manually annotated brain mask of 360 mm^3. In order to compute Dice scores, we rasterize the networks using a tubular model of circular shape and compare them with voxel-grained annotations that were obtained manually on a set of selected slices along each direction. As a consequence of this rasterization, inaccuracies on the voxel level are introduced that negatively affect absolute values of the Dice score, which is already very sensitive for thin structures such as vessels. Note, however, that all compared strategies are affected by this to the same

degree. We find the optimized network (opt, $\alpha = 1$) to always group with the more favourable of the alternative, which is a low vessel volume, a moderate, i.e. neither too small nor too large, extravascular distance and a high Dice score, while each of the alternative approaches provides poor results in at least one of these scores. The relative vessel volumes vary between 5.2 % and 6.5 % for our datasets, while [146] reported values from 3.6 % to 4.2 % and also observed some inter-subject variability. Comparing the result of our optimization with parameter $\alpha = 1$ and the maximal choice of $\alpha_{\max}$, we observe that both relative vessel volume and Dice score vary only slightly, while the average distance increases when using $\alpha_{\max}$, hence resulting in networks with a decreased vascular density. While the results support the assumption that combining image evidence with geometric prior benefits network extraction, it will require analysis on more extensive databases to quantify the methods robustness in presence of specific pathologies and malformations.

Comparing the processing time of the proposed solver (Sect. 3.2.2) and a standard method (IBM ILOG CPLEX V12.51 [118]) as reported in Table 3.1, we find that our approach of handling subproblems during the optimization leads to an improvement: runtimes (one run each, on a quadcore CPU, 32 GB RAM) were shorter in three out of four cases, whereas the standard method took between 44 % and 107 % longer. We find that absolute run times may vary widely depending on the size of the overconnected graph and its connectedness as well as parametrization due to the dynamic processing scheme. For those two datasets that run much longer (# 1 and # 2), we observe that the number of removed segments is two to three times higher than for the two others, while the resulting, optimized networks describe all a very similar vascular volume (as seen in Fig. 3.8). We attribute this to a situation where the overconnection scheme is not as efficient as for the two latter datasets and a high number of connections has to be discarded in the regularization. The relative runtimes indicate that the approach of partitioning the ILP into independent subproblems yields larger gains on datasets which required more iterations in the constraint generation, i.e. had longer absolute runtimes.

Table 3.1: Runtimes in minutes on different µMRA datesets for our proposed solving scheme (t_{ilpcc}) and an off-the-shelf solver (t_{cplex}). We observe that our method is faster as the off-the-shelf solver in the majority of cases.

Dataset	# 1	# 2	# 3	# 4
t_{ilpcc}	2060.6	186.0	18.6	8.8
t_{cplex}	2958.6	385.0	31.6	8.7
$\frac{t_{\text{cplex}}-t_{\text{ilpcc}}}{t_{\text{ilpcc}}}$	0.44	1.07	0.70	-0.01

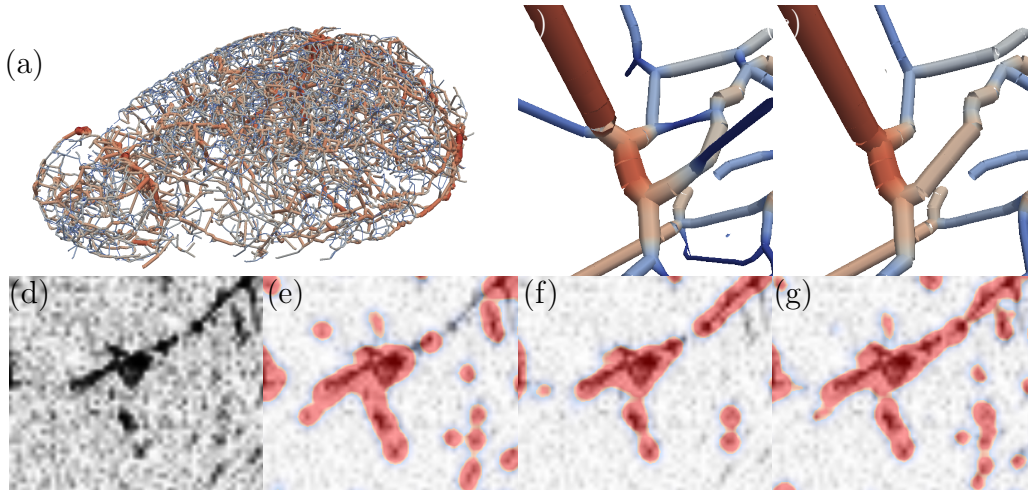


Figure 3.7: Visualisation of the results. (a) Rendered vascular network extracted with our method (opt, $\alpha = 1$) (b) Detail view before and (c) after optimization. Colours change with vessel diameter. (d) Raw image. (e) Rasterized DOHT network $\theta = 0.5$ and (f) with $\theta = 0.9$. (g) Postprocessed network with our method. Note that for the rasterization, a simple tube model is used and therefore, not a perfect voxel-grained segmentation is to be expected but rather a qualitative visualization that indicates whether or not a structure is present in the network model (shown as red overlay).

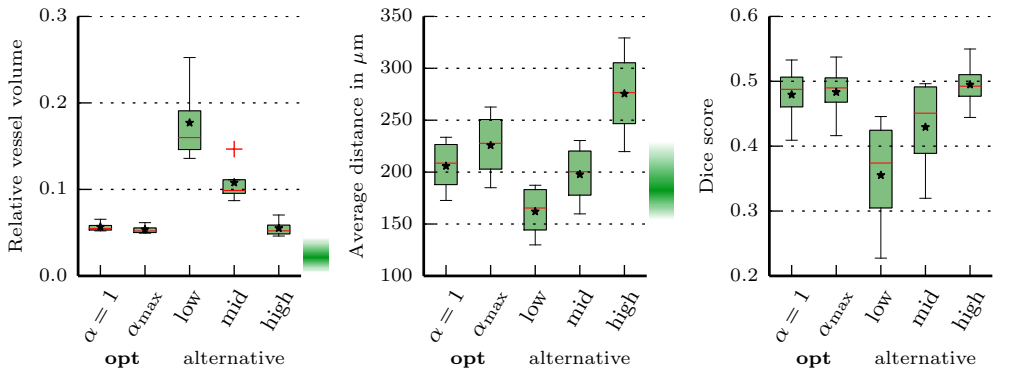


Figure 3.8: Comparison between the optimized networks (opt) with $\alpha = 1$, $\alpha_{\max} = \alpha \to \infty$ and simple networks obtained at thresholds $\theta = 0.2, 0.5$ and 0.9 (alternative: low $= 0.2$, mid $= 0.5$, high $= 0.9$). Boxplots (median in red, mean as $\star$) depict the statistics on all four µMRA datasets. The gradients (green) to the right of both plots of physiological scores indicate physiologically plausible ranges (cf. [146]). The Dice score has been computed by comparing voxel-wise annotations with the rasterized the networks using a simple tube model (negatively affecting absolute Dice values for all approaches to the same degree). Our regularization (opt, $\alpha = 1$) always groups with the more favourable of the alternatives, i.e. it has a small vessel volume, a moderate average distance and yet an acceptable high Dice score.

3.4 Conclusions

We have introduced a probabilistic approach for extracting vessel networks and we can compute the MAP estimate efficiently by solving an ILP. We learned physiological-geometric properties considered in the probabilistic model from a high-resolution corrosion cast µCT of a murine cerebrovascular network and applied it to low-resolution, in-vivo µMRA images, leading to superior extracted networks in terms of macroscopic measures.

Our method can be applied as post-processing step to existing vessel segmentation pipelines in order to incorporate physiological knowledge for improved network extraction. As such, it can be combined with other graph generation methods that are potentially able to overconnect the graph, such as discussed in [138, 139]. It can be used with any other geometrical-physiological prior knowledge about properties, for example considering

vessel shape, length, curvature and flow direction, or a non-parametric learning as in [137]. Even a spatially-variant prior could be designed and incorporated to address different vascular properties in certain areas, similar to the parametric maps used for latent classes in [147, 148]. Furthermore, higher-order network motifs that occur frequently in vasculature could be included into the prior model in the same way as bifurcations. It should be noted, however, that increasing the complexity of the prior model would likely require a larger database for training.

For future studies, it would be interesting to assess the robustness of the approach with respect to specific pathologies and abnormalities in the vascular network. Another direction could be the extension of the algorithm to jointly infer a subject-specific parametrization of the physiological model and predict the underlying network.

Acknowledgements

This research was supported by the Technische Universität München - Institute for Advanced Study (funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n 291763, the Marie Curie COFUND program of the the European Union), by grants from the EMDO foundation, Swiss National Science Foundation grant 136822, and the Swiss National Center of Competence in Research on Computer Aided and Image Guided Medical Interventions (NCCR Co-Me) supported by the Swiss National Science Foundation. Use of the Advanced Photon Source was supported by the US Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

Supplement

In Sect. 3.2.2, we found that the MAP estimate $\mathbf{x}^*$ is given by

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\Omega} P(\mathbf{X} = \mathbf{x}|I, G) \ . \tag{3.31}$$

Here, we are going to derive the cost function from the definition of the MRF (3.3). We start by using the definitions of the potential functions $\phi\,(.)$,

(3.4) to (3.6), in the logarithm of $P(\mathbf{X}=\mathbf{x}|I,G)$:

$$\begin{aligned}
&\log P(\mathbf{X}=\mathbf{x}|I,G) \propto \\
&\quad \alpha \sum_{x_i \in X} \log\left(P(x_i=1|I_i,E_i)^{x_i} P(x_i=0|I_i,E_i)^{1-x_i}\right) + \sum_{\substack{x_i,x_j\in X:\\ e_i,e_j \text{ adjacent}\\ \nexists e_k \text{ adjacent to } e_i\wedge e_j}} \log\left(p_{C,ij}^{x_i x_j} p_T^{1-x_i x_j}\right) \\
&\quad + \sum_{\substack{x_i,x_j,x_k\in X:\\ e_i,e_j,e_k \text{ adjacent}}} \log\left(p_{B,ijk}^{x_i x_j x_k} p_T^{(1-x_i)(1-x_j)(1-x_k)} \prod_{(i',j')\in S} p_{C,i'j'}^{x_{i'}x_{j'}-x_i x_j x_k}\right) \qquad (3.32) \\
&= \alpha \sum_{x_i\in X} x_i \log\frac{P(x_i=1|I_i,E_i)}{P(x_i=0|I_i,E_i)} + \log P(x_i=0|I) + \sum_{\substack{x_i,x_j\in X:\\ e_i,e_j \text{ adjacent}\\ \nexists e_k \text{ adjacent to } e_i\wedge e_j}} x_i x_j \log\frac{p_{C,ij}}{p_T} + \log p_T \\
&\quad + \sum_{\substack{x_i,x_j,x_k\in X:\\ e_i,e_j,e_k \text{ adjacent}}} \left(x_i x_j x_k \log\frac{p_{B,ijk} p_T^2}{\prod_{(i',j')\in S} p_{C,i'j'}} + \sum_{(i',j')\in S} \left(x_{i'} x_{j'} \log\frac{p_{C,i',j'}}{p_T}\right) + \log p_T \right) , \\
& \qquad (3.33)
\end{aligned}$$

where we applied multilinear representations for pseudo-boolean functions and S is the set of pairs out of $\{i,j,k\}$, i.e. $S = 2^{\{i,j,k\}}$. Discarding the constant terms, rearranging the pairwise terms originating from the ternary potentials and merging them with the uniquely pairwise terms – which generates per definition no doubles – then leads to

$$\begin{aligned}
&\log P(\mathbf{X}=\mathbf{x}|I,G) \propto \\
&\quad \alpha \sum_{x_i\in X} x_i \log\frac{P(x_i=1|I_i,E_i)}{P(x_i=0|I_i,E_i)} + \sum_{\substack{x_i,x_j\in X:\\ e_i,e_j \text{ adjacent}}} x_i x_j \log\frac{p_{C,ij}}{p_T} \qquad (3.34) \\
&\quad + \sum_{\substack{x_i,x_j,x_k\in X:\\ e_i,e_j,e_k \text{ adjacent}}} x_i x_j x_k \log\frac{p_{B,ijk} p_T^2}{p_{C,ij} p_{C,ik} p_{C,jk}} \ . \qquad (3.35)
\end{aligned}$$

Finally, taking the negative logarithm turns the maximization into a minimization:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\Omega} -\log P(\mathbf{X}=\mathbf{x}|I,G) \qquad (3.36)$$

$$= \arg\min_{\mathbf{x}\in\Omega} \ \alpha \sum_{x_i \in X} -\log \frac{P\left(x_i = 1 | I_i, E_i\right)}{P\left(x_i = 0 | I_i, E_i\right)} x_i + \sum_{\substack{x_i, x_j \in X: \\ e_i, e_j \text{ adjacent}}} -\log \frac{p_{C,ij}}{p_T} x_i x_j \quad (3.37)$$

$$+ \sum_{\substack{x_i, x_j, x_k \in X: \\ e_i, e_j, e_k \text{ adjacent}}} -\log \frac{p_{B,ijk} p_T^2}{p_{C,ij} p_{C,ik} p_{C,jk}} x_i x_j x_k \ , \quad (3.38)$$

which is equivalent to the integer program:

$$\min_{\mathbf{x}} \quad J(\mathbf{x}) = \ \alpha \sum_{x_i \in X} w_i x_i + \sum_{\substack{x_i, x_j \in X: \\ e_i, e_j \text{ adjacent}}} w_{ij} x_i x_j + \sum_{\substack{x_i, x_j, x_k \in X: \\ e_i, e_j, e_k \text{ adjacent}}} w_{ijk} x_i x_j x_k \ , \quad (3.39)$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} \ , \quad (3.40)$$

$$x_i \in \{0, 1\} \qquad \forall x_i \in X \ , \quad (3.41)$$

with the weights given as:

$$w_i = -\log \frac{P\left(x_i = 1 | I_i, E_i\right)}{P\left(x_i = 0 | I_i, E_i\right)} \ , \quad (3.42)$$

$$w_{ij} = -\log \frac{p_{C,ij}}{p_T} \ , \quad (3.43)$$

$$w_{ijk} = -\log \frac{p_{B,ijk} p_T^2}{p_{C,ij} p_{C,ik} p_{C,jk}} \ . \quad (3.44)$$

4

The Minimum Cost Connected Subgraph Problem in Medical Image Analysis

This chapter has been published as **peer-reviewed conference paper**.

M. Rempfler, B. Andres, and B. H. Menze. "The Minimum Cost Connected Subgraph Problem in Medical Image Analysis." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III*. ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Cham: Springer International Publishing, 2016, pp. 397–405. DOI: 10.1007/978-3-319-46726-9_46

Synopsis: This work discusses the minimum cost connected subgraph (MCCS) problem in the context of medical image analysis. We contribute to exact optimization procedures by proposing novel constraint generation strategies and and objective-dependent constraints. Finally, we benchmark them against approximative algorithms on two medical benchmark datasets.

Contributions of thesis author: Conception and implementation of algorithms, computational experiments and interpretation, composition of manuscript.

Abstract

Several important tasks in medical image analysis can be stated in the form of an optimization problem whose feasible solutions are connected subgraphs. Examples include the reconstruction of neural or vascular structures under connectedness constraints.

We discuss the minimum cost connected subgraph (MCCS) problem and its approximations from the perspective of medical applications. We propose a) objective-dependent constraints and b) novel constraint generation schemes to solve this optimization problem exactly by means of a branch-and-cut algorithm. These are shown to improve scalability and allow us to solve instances of two medical benchmark datasets to optimality for the first time. This enables us to perform a quantitative comparison between exact and approximative algorithms, where we identify the geodesic tree algorithm as an excellent alternative to exact inference on the examined datasets.

4.1 Introduction

The *minimum cost connected subgraph* (MCCS) optimization problem arises in several medical image analysis tasks, most prominently for segmenting neural structures [30] or reconstructing vascular networks [20], where the MAP subgraph under connectedness constraints is inferred. Variations of this optimization problem have been proposed for anatomical labelling of vasculature [31] or artery-vein separation [32]. Imposing connectedness serves as regularizer, suppressing spurious detections and complementing incomplete observations, and it is often a requirement for further processing steps, e.g. if the reconstructed vasculature shall be used for biophysical simulations.

While [30, 20, 31, 32] successfully solve an MCCS problem on heavily preprocessed, application-specific, sparse graphs, it would also be interesting to enforce connectedness on both very dense or large grid-graphs, for example in low-level segmentation tasks (Fig. 6.1, left), for 3D/4D reconstruction problems (Fig. 6.1, middle and right) or when it is not possible to reliably reduce the candidate graphs size. In these cases, however, the computational

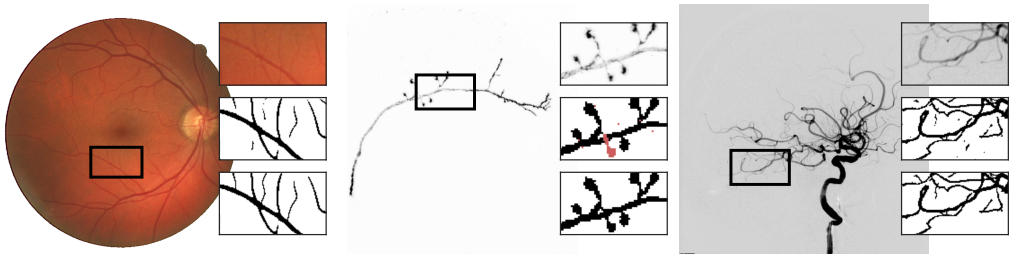


Figure 4.1: *Examples for the MCCS on grid graphs.* **Left**: Segmentation of vasculature in retinal images. **Middle**: Reconstruction of a neuron from a 3D stack. Excessive disconnected components are shown in red for better visibility. **Right**: Delineation of vessels in a digital subtraction angiography (DSA) time series. The detail views show: raw image (**top**), without connectedness (**middle**) and with connectedness (**bottom**). Imposing connectedness constraints, i.e. requiring an minimum cost connected subgraph (MCCS), helps to reconnect disconnected terminals and remove spurious detections without penalizing thin tubular structures.

complexity becomes challenging. In fact, it was shown to be NP-hard in [63]. Nowozin & Lampert [64] propose an exact algorithm that tightens an outer polyhedral relaxation of the connected subgraph polytope by cutting planes. However, without guarantee to terminate in polynomial time, it was found to be too slow to solve typical instances of medical benchmark datasets to optimality. To this end, two heuristical algorithms were proposed by Chen et al. [65] and Stühmer et al. [66]. They either use an approximative formulation of the connected subgraph polytope by means of a precomputed geodesic shortest path tree [66] or iteratively solve a surrogate problem that is based on altered weights of the original problem [65]. Both approaches are fast enough for medical applications and were reported to yield qualitatively promising results. A quantitative comparison, however, has been prevented by the prohibitively expensive computation of exact solutions to the MCCS problem.

In this paper, we revisit the MCCS in an integer linear programming (ILP) framework for MAP estimation under connectedness constraints. First, we contribute to the exact optimization by proposing a) *objective-dependent constraints* that reduce the size of the polytope and hence, reduce the number of potential solutions to explore, and b) *constraint generation strategies* beyond the standard *nearest* and *minimal* separator strategy,

which we show to have a strong impact on the runtime of the ILP. Both propositions together enable us to compute the MCCS on several instances of two medical benchmark datasets – addressing vessel segmentation and neural fiber reconstruction – to optimality. Our second contribution is a first quantitative comparison of the exact algorithm and the two heuristics in terms of runtime, objective function and semantic error metrics.

4.2 Background

We are interested in the most likely binary labeling $\mathbf{x} \in \{0,1\}^{|V|}$ of the nodes V in the graph $G = (V, E)$. A node i is active if $x_i = 1$. By imposing connectedness constraints, i.e. $\mathbf{x} \in \Omega$, the MAP estimate becomes a MCCS problem:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\{0,1\}^{|V|}}{\arg\max}\, P\left(\mathbf{X} = \mathbf{x}|I, \Omega\right) = \underset{\mathbf{x}\in\Omega}{\arg\max}\, P\left(\mathbf{X} = \mathbf{x}|I\right) \quad , \tag{4.1}$$

where I is the image evidence and Ω denotes the set of $\mathbf{x}$ that are connected subgraphs of G. In this section, we discuss two formulations of Ω, the exact formulation that follows [64] and the geodesic tree formulation of [66].

4.2.1 Exact Connectedness

Following [64], we can describe Ω with the following set of linear inequality constraints

$$\forall i, j \in V, (i,j) \notin E : \forall \mathcal{S} \in S(i,j) \quad x_i + x_j - 1 \leq \sum_{k\in\mathcal{S}} x_k \quad , \tag{4.2}$$

where $\mathcal{S}$ is a set of vertices that separate i and j, while $S(i,j)$ is the collection of all vertex separator sets for i and j. In other words, if two nodes i and j are active, then they are not allowed to be separable by any set of inactive nodes. Thus, a path of active nodes has to exist. In practice, this set of constraints is too large to be generated in advance. However, given a labelling $\mathbf{x}$ we can identify at least a subset of the violated connectedness constraints in polynomial time, add them to the ILP and search for a new feasible solution. This approach is known as *lazy constraint generation*. In Section 4.3.2, we detail on identifying and adding these constraints.

Rooted case. In many medical segmentation problems, it is reasonable to assume that a root node can be identified aforehand with an application-specific detector, manually or by a heuristic, such as picking the strongest node in the largest component. If a known root r exists, it suffices to check connectedness to the root node instead of all pairs of active nodes. The constraints in (4.2) then become

$$\forall i \in V \setminus \{r\}, (r,i) \notin E : \forall \mathcal{S} \in S(i,r) \quad x_i \leq \sum_{k \in \mathcal{S}} x_k \ . \tag{4.3}$$

4.2.2 Geodesic Tree Connectedness

Alternative to the exact description of all connected subgraphs that we discussed in the previous section, we can formulate a connectedness prior as in [66] on a *geodesic shortest path tree* $T(G) = (V, A \subseteq E)$ rooted in r. Here, $T(G)$ is precomputed based on the unary potentials, i.e. with edge weights defined as $f(i,j) = \frac{1}{2}(\max(w_i, 0) + \max(w_j, 0))$. The set of feasible solutions is then given by the inequalities:

$$\forall i \in V \setminus \{r\}, (p,i) \in T(G) \quad x_i \leq x_p \ , \tag{4.4}$$

where p is the parent of i in the geodesic tree $T(G)$. With this set of constraints, a node i can only be active if his parent p in the geodesic tree is also active, thus connecting all active nodes to the root r along the branches of $T(G)$. Advantages of this approach are that only $|V| - 1$ constraints are necessary to describe the set of feasible solutions and that the relaxation is tight. On the other hand, the inequalities of (4.4) describe a strict subset of (4.3), unless $T(G) = G$. Hence it might discard an optimal solution that is feasible in (4.3).

4.3 Methods

Given the probabilistic model $P(\mathbf{X} = \mathbf{x}|I)$ of (4.1) is a random field over $G = (V, E)$, we can write its MAP estimator $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \{0,1\}} P(\mathbf{X} = \mathbf{x}|I, \Omega)$ as an ILP. We will assume for the remaining part that $P(\mathbf{X} = \mathbf{x}|I) =$

$\prod_{i \in V} P(x_i|I)$, leading to the ILP:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in V} w_i x_i \ , & (4.5) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \ , & (4.6) \\ & \mathbf{x} \in \{0,1\}^{|V|} \ , & (4.7) \end{aligned}$$

where (4.6) are the connectedness constraints, i.e. either (4.3) or (4.4), (4.7) enforces integrality, and w_i are the weights that can be derived as $w_i = -\log \frac{P(x_i=1|I)}{1-P(x_i=1|I)}$. Higher order terms of the random field can be incorporated by introducing auxiliary binary variables and according constraints as done in [20]. Note, however, that [65] reported problem instances with weak or no pairwise potentials – as we are addressing them here – to be amongst the most difficult.

4.3.1 Objective-dependent Constraints

Given the problem with unary terms, we observe that, for any connected component $\mathcal{U} \subset V$ composed of *unfavourable* nodes only, i.e. $\forall i \in \mathcal{U}, w_i > 0$, it can only be active in the optimal solution if there are at least two active nodes in its neighbourhood:

$$\forall i \in \mathcal{U} \quad 2x_i \leq \sum_{j \in \cup_{k \in \mathcal{U}} \delta(k) \setminus \mathcal{U}} x_j \ , \tag{4.8}$$

where $\delta(k)$ is the set of neighouring nodes to k. In other words, unfavourable nodes can not form a leaf in the optimal solution (otherwise, removing the unfavourable nodes would give us a better solution without loosing connectedness). In the special case of $|\mathcal{U}| = 1$, we can add the constraint from the beginning. This removes feasible solutions from Ω that are a priori known to be suboptimal, hence reducing the search space in the optimization and making it unnecessary to add a large set of separator inequalities.

Higher-order weights. Even though we only define (4.8) for unary weights, it is possible to adapt the constraint to higher-order models by changing the condition to $w_i + \min_{j \in \delta(i)} w_{ij} > 0$, provided the pairwise weights w_{ij} are only introduced for neighbouring nodes i, j such that $(i, j) \in E$.

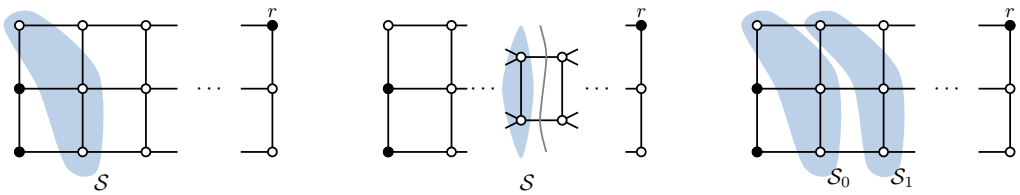


Figure 4.2: *Constraint generation strategies.* Illustration of the nearest separator (left), minimal separator (middle) and k-nearest (right) strategies. Active nodes are shown in black, inactive nodes are white and the identified separator sets $\mathcal{S}$ are marked in blue. $\mathcal{S}$ is subsequently used to generate the corresponding constraint in (4.2) or (4.3).

4.3.2 Constraint Generation Strategies

The extensive number of inequalities needed for (4.3) makes it necessary to identify violated constraints during the optimization and add them to the problem. We note that it suffices to treat individual connected components as one entitity, since establishing a connection automatically connects all pairs of nodes between them. Identifying violated constraints boils down to finding a vertex separator set $\mathcal{S}$ between two disconnected, active components in the current solution. The constraints corresponding to $\mathcal{S}$ are then generated according to (4.2) or (4.3) for all nodes in the given connected component.

At the heart of this technique is the observation that only a subset of inequalities is active at the optimum of a given problem instance. However, depending on the choice of the inequalities that we add in each step, we may explore (and therefore construct) different parts of the polytope Ω, most likely requiring a different number of iterations.

In the following, we first review the two standard strategies, namely the *nearest* and *minimal* separator, and then propose several novel, alternative strategies.

Nearest separator. In this standard approach, the vertex separator set in the immediate neighbourhood of the active component is picked for generating the new constraint. This strategy has been used, for example, in [20]. It is motivated by its simplicity and the fact that it often coincides with the minimal separator strategy for small components.

Minimal separator. A minimal (in terms of $|\mathcal{S}|$) separator set is obtained

by solving a max-flow problem between any two disjoint active components at hand and selecting the smaller vertex set on either side of the resulting min-cut. For the max-flow, we set the flow capacity c in edge (i, j) as $c(i, j) = \max(1 - x_i, 1 - x_j)$. The strategy was applied in [64].

Equidistant separator. Alternatively, we can identify the separator set $\mathcal{S}$ that is equidistant to the current active component and all other components by running a breadth-first search (BFS) from either side. Similar to the max-flow of the minimal separator, the distance measure is only accounting for non-active nodes. This strategy originates in the observation that the weakest evidence between two components is often found half-way into the connecting path.

k-Nearest and k-Interleave. We run a BFS from the active component C and collect the k (disjoint) separator sets $\{\mathcal{S}_n\}_{n=0}^{k-1}$ composed of all nodes with identical distance. The search terminates if k equals the number of nodes in C or if another active node is reached. For the k-interleave, only separators with even distance are chosen. The intuition behind these strategies is that a wider range of neighbours (and their neighbours) has to be considered for the next solution.

4.4 Experiments & Results

Datasets & Preprocessing.

We conduct experiments on two medical datasets: First, on the DRIVE database of retinal images [69], each being 565×584 px. We use the probability estimates $P(x_i = 1|I)$ for a pixel i being vasculature from the recent state-of-the-art approach of [56] for our unaries. Second, we run experiments on the olfactory projection fibers (OPF) dataset [53], composed of 8 3D confocal microscopy image stacks. We use the stacks prepared in [30], where we estimate $P(x_i = 1|I)$ of voxel i being part of the fiber by a logistic regression on the image intensities. We segment the nerve fiber under the requirement of connectedness on the 3D grid graph of $256 \times 256 \times n$ nodes with $n \in \{30, \ldots, 51\}$ depending on the case. The probability $P(x_i = 1|I)$ of voxel i being part of the fiber is estimated by a logistic regression on the image intensities. Both datasets are illustrated in Fig. 6.1.

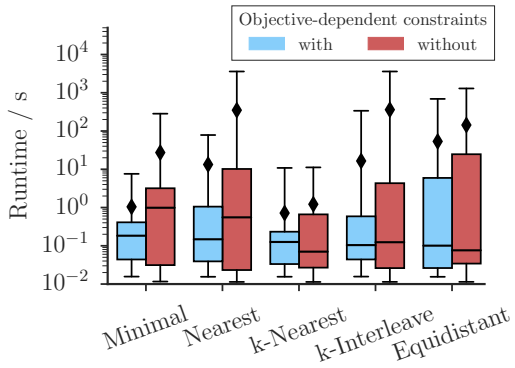


Figure 4.3: Runtime with and without the proposed objective-dependent constraints on 64×64 instances. Mean values are depicted by ⧫, whiskers span $[\min, \max]$ values. Unsolved instances are excluded for readability. We find that all strategies benefit from the additional constraints. Additional *per-instance* information can be found in the supplement.

Optimization.

We solve the ILP (4.5) by the branch-and-cut algorithm of the solver Gurobi [117] with a default relative gap of 10^{-4}. Objective-dependent constraints for single nodes (Section 4.3.1) are added from the beginning. For the exact connectedness (Section 4.2.1), the strategies described in Sect. 4.3.2 are implemented as a callback: Whenever the solver arrives at an integral solution $\mathbf{x}'$, violated constraints are identified and added to the model. If no such violation is found, i.e. $\mathbf{x}'$ is already connected, then it is accepted as new current solution $\mathbf{x}^*$. For the geodesic tree connectedness (Section 4.2.2), all constraints are added at once. In order to arrive at a fair comparison, we define the root node for both approaches.

Experiment: Objective-dependent constraints.

To examine the impact of the objective-dependent constraints, we subsample 25 subimages of 64×64 px from the DRIVE instances and run the ILP once with and once without the additional first order constrains of (4.8). As shown in Fig. 4.3, we find that all strategies benefit from the additional constraints.

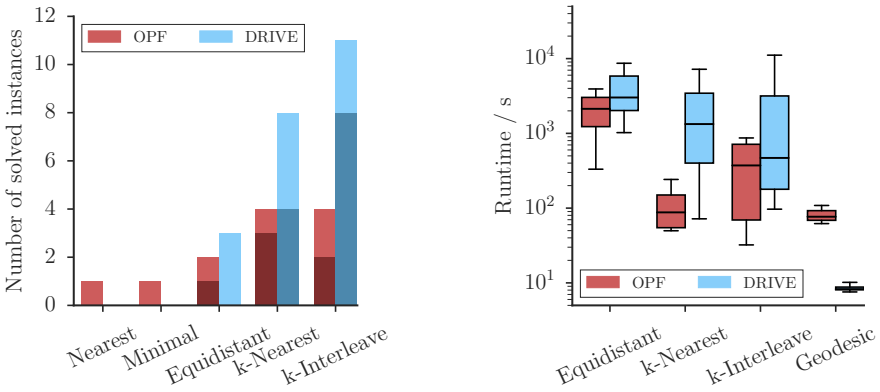


Figure 4.4: **Left**: Number of solved instances per strategy. The darker bar indicates how often a strategy was the *fastest* to solve an instance. **Right**: Runtime on solved instances. Strategies with too few solved instances are not included. k-Nearest and k-Interleave are found to be the most successful exact strategies.

Experiment: Comparing exact and approximative algorithms.

We compare exact and geodesic tree MCCS on both datasets. On 2D images, we additionally compare to the method by [65] called Topocut. As a baseline, we compute the maximum connected component in the non-constrained solution (Maxcomp). The results are presented in Fig. 5.7 and Table 4.1 (additional information *per instance* is provided in the supplement). We observe that 6/8 and 12/20 instances were solved to optimality with our propositions, while standard strategies solved ≤ 1. k-Nearest and k-interleave are the two most successful exact strategies in terms of solved instances and speed. In terms of segmentation scores, the two heuristics are on par with the exact algorithm, while all of them outperform the baseline. We find the geodesic approach to match the exact solution with respect to objective values in all instances (within a relative difference of 10^{-4}), whereas Topocut often obtains slightly lower objective values than the geodesic approach. A qualitative comparison between an exact and geodesic solution is presented in Fig. 4.5.

Table 4.1: Segmentation scores in terms of **F1**-score, (**P**recision, **R**ecall) in % on the solved instances. All approaches outperform the baseline (MaxComp), while no significant difference can be found between them.

	OPF			DRIVE		
	F1	(P	R)	F1	(P	R)
Maxcomp	68.5	(67.7,	71.9)	78.7	(87.2,	72.1)
Geodesic	76.2	(69.1,	85.4)	80.1	(86.2,	75.2)
Topocut	-	-	-	80.1	(86.4,	74.9)
Exact	76.2	(69.1,	85.4)	80.1	(86.2,	75.2)

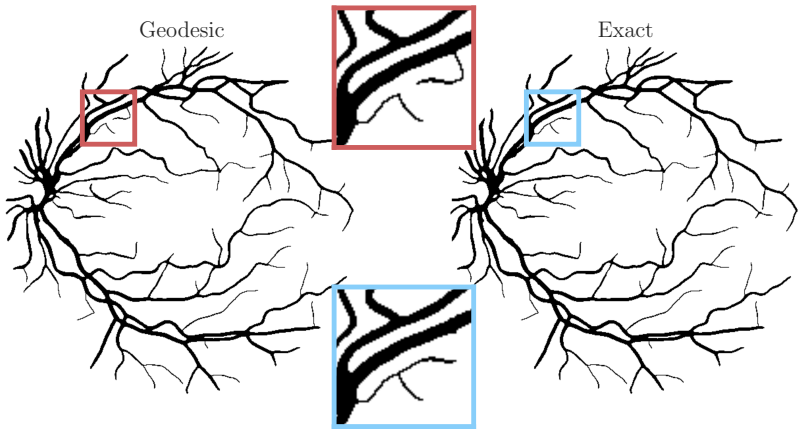


Figure 4.5: Comparison of exact and approximative connectedness: Major differences as the one indicated are encountered mainly if solutions are competing under the model $P(\mathbf{X} = \mathbf{x}|I)$ and thus almost equivalent w.r.t. objective value.

4.5 Conclusions

We have shown that exact optimization of the MCCS, as it is typical for neural and vascular structure reconstruction tasks, strongly benefits from the proposed objective-dependent constraints and the constraint generation strategies. In a first quantitative comparison between exact and approximative approaches on two datasets, we found that the geodesic tree formulation is a fast, yet highly competitive alternative to exact optimization.

While we focussed on large grid-graphs that are most important for low-level segmentation and reconstruction, we expect that our findings transfer to MCCS problems and related ILP-based formulations on sparse graphs, e.g. those discussed in [30, 20, 31, 32], and thus consider this a promising direction for future work. Besides, it will be intersting to investigate the effect of our propositions in the presence of higher-order terms.

Acknowledgements

With the support of the Technische Universität München – Institute for Advanced Study, funded by the German Excellence Initiative (and the European Union Seventh Framework Programme under grant agreement n 291763).

5

Efficient Algorithms for Moral Lineage Tracing

This chapter has been published as **peer-reviewed conference paper**.

M. Rempfler*, J.-H. Lange*, F. Jug, C. Blasse, E. W. Myers, B. H. Menze, and B. Andres. "Efficient Algorithms for Moral Lineage Tracing." In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 4705–4714. DOI: 10.1109/ICCV.2017.503

*Authors contributed equally.

Synopsis: The moral lineage tracing problem (MLTP) [86] is a mathematical abstraction of the joint segmentation and tracking of living cells as they move and divide. This work addresses runtime issues of previous optimization algorithms for the MLTP by proposing (i) two primal feasible local search heuristics and (ii) an improved branch-and-cut algorithm.

Contributions of thesis author: Design and implementation of local search heuristics, computational experiments with heuristics, interpretation, composition and revision of manuscript.

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. [86] 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.

5.1 Introduction

Recent advances in microscopy have enabled biologists to observe organisms on a cellular level with higher spatio-temporal resolution than before [149, 100, 150]. Analysis of such microscopy sequences is key to several open questions in biology, including embryonic development of complex organisms [70, 71], tissue formation [72] or the understanding of metastatic behavior of tumor cells [73]. 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 [74, 75, 76, 77, 78, 79].

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. 6.1). The tracking subproblem is complicated by cells that enter or leave the field of view, or low temporal resolution that allows

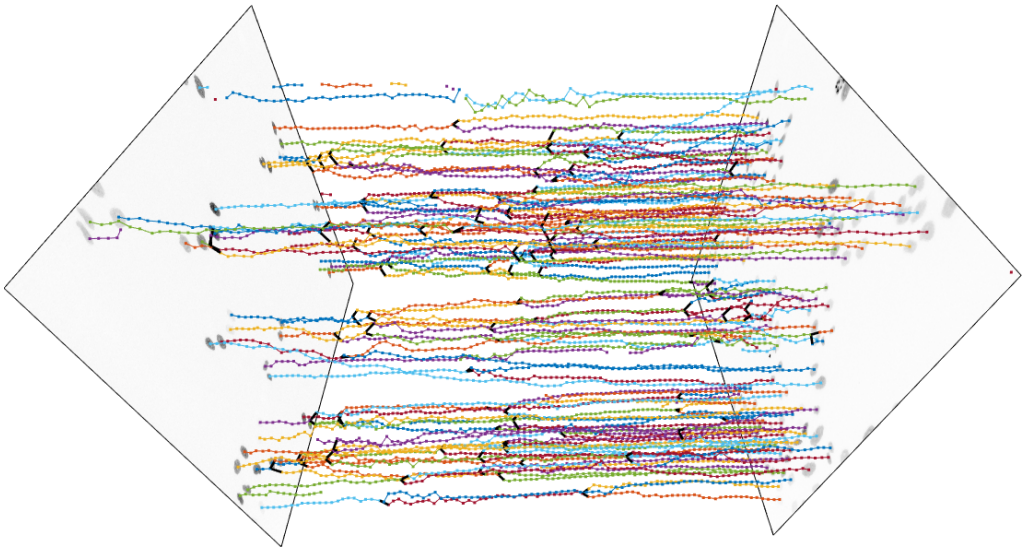


Figure 5.1: Depicted above is a lineage forest of cells from a sequence of microscopy images. The first image of the sequence is shown on the left. The last image is shown on the right. Cell divisions are depicted in black.

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 [33, 34, 35, 36, 37] or reconstruction of tree-like structures [151, 20, 104, 30, 152]. It has been cast in the form of different optimization problems [81, 82, 83, 85, 84] that can deal with some of the mentioned difficulties, e.g., by selecting from multiple segmentation hypotheses [85, 84].

Jug et al. [86], on the other hand, have proposed a rigorous mathematical abstraction of the joint problem which they call the *MLTP*. It is a hybrid of the *MCMCP*, which has been studied extensively for image decomposition [87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98], and the *minimum cost disjoint arborescence* problem, variations of which have been applied to reconstruct lineage forests in [81, 82, 83, 84, 85] or tree-like structures [151, 152, 30]. 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. 5.2). Solving this optimization problem therefore tackles both subtasks – segmentation and

tracking – simultaneously. While Jug et al. [86] 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 [95, 153] 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 [86] 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 [86], 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.

5.2 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 [86] as an 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. 5.2. For convenience, we further write $G_t = (V_t, E_t)$ for the subgraph corresponding to frame t and

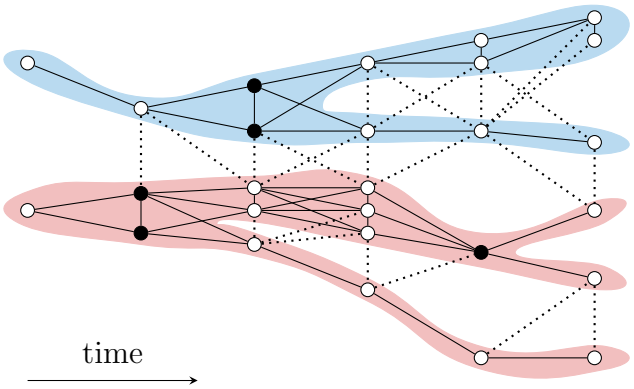


Figure 5.2: The *moral lineage tracing (MLT)*[1]: Given a sequence of images decomposed into cell fragments (depicted as nodes in the figure), cluster fragments into cells in each frame and *simultaneously* associate cells into lineage forests over time. Solid edges indicate joint cells within images and descendant relations across images. Black nodes depict fragments of cells about to divide.

$G_t^+ = (V_t^+, E_t^+)$ 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 $(V, E_t \setminus L)$.

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 [86], 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 5.2.1 ([86]). 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 (5.1) – (5.3):

$$\forall t \in \mathcal{T} \forall C \in \text{cycles}(G_t) \forall e \in C :$$
$$x_e \leq \sum_{e' \in C \setminus \{e\}} x_{e'} \tag{5.1}$$
$$\forall t \in \mathcal{T} \forall \{v, w\} \in E_{t,t+1} \forall P \in vw\text{-paths}(G_t^+) :$$
$$x_{vw} \leq \sum_{e \in P} x_e \tag{5.2}$$
$$\forall t \in \mathcal{T} \forall \{v_t, v_{t+1}\}, \{w_t, w_{t+1}\} \in E_{t,t+1} (\text{with } v_t, w_t \in V_t)$$
$$\forall S \in v_t w_t\text{-cuts}(G_t) \forall P \in v_{t+1} w_{t+1}\text{-paths}(G_{t+1}) :$$
$$1 - \sum_{e \in S} (1 - x_e) \leq x_{v_t v_{t+1}} + x_{w_t w_{t+1}} + \sum_{e \in P} x_e \tag{5.3}$$

Jug et al. refer to (5.1) as *space cycle*, to (5.2) as *space-time cycle* and to (5.3) as *morality* constraints. We denote by X'_G the set of all $x \in \{0,1\}^E$ that satisfy (5.1) – (5.3). For the formulation of the additional *bifurcation* constraints, which guarantee that the associated lineage cut is binary, we refer to [86, 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, [86] defines

[1] The figure is a correction of the one displayed in [86].

[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. [154].

the following *moral lineage tracing problem* (MLTP)

$$\min_{x,x^+,x^-} \quad \sum_{e\in E} c_e x_e + \sum_{v\in V} c_v^+ x_v^+ + \sum_{v\in V} c_v^- x_v^- \tag{5.4}$$

$$\text{subject to} \quad x \in X_G, \quad x^+, x^- \in \{0,1\}^V, \tag{5.5}$$

$$\forall t \in \mathcal{T} \forall v \in V_{t+1} \forall S \in V_t v\text{-cuts}(G_t^+) :$$

$$1 - x_v^+ \leq \sum_{e\in S} (1 - x_e), \tag{5.6}$$

$$\forall t \in \mathcal{T} \forall v \in V_t \forall S \in v V_{t+1}\text{-cuts}(G_t^+) :$$

$$1 - x_v^- \leq \sum_{e\in S} (1 - x_e). \tag{5.7}$$

The inequalities (5.6) and (5.7) are called *birth* and *termination* constraints, respectively.

5.3 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 [155] 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 $\mathcal{V}$. 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. 5.3. 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.

5.3.1 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 [95] heuristic for the MCMCP in the sense that it always takes

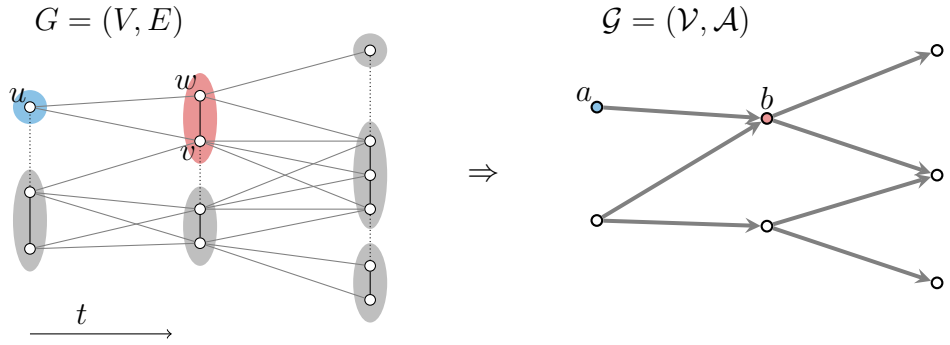


Figure 5.3: For a fixed decomposition of the frames (depicted with black solid/dashed cut edges), we associate a directed graph $\mathcal{G}$ over the components $\mathcal{V}$. The arcs $\mathcal{A}$ bundle all edges going from any node of one cell to any node of another cell in the successive frame. 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**
  else applyTransform$(\mathcal{G}, a, b)$ ▷ updates partitions of $\mathcal{G}$ and selects arcs $\mathcal{A}(y)$.
    break
  end if
end while
return edgeLabels$(\mathcal{G})$ ▷ cut-edge labeling x^* from $\mathcal{V}$ and $\mathcal{A}(y)$.

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 determines which

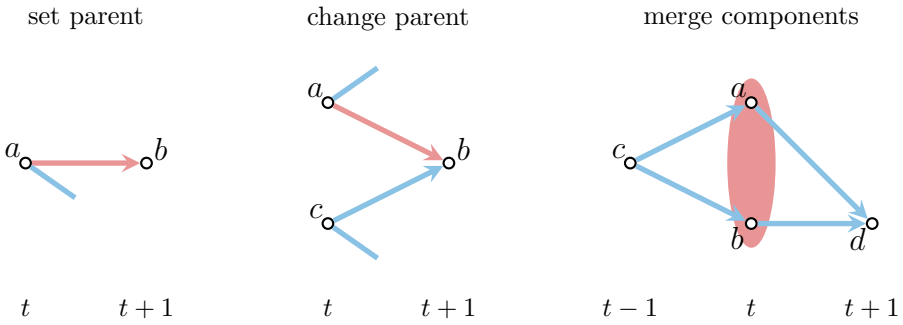


Figure 5.4: The three transformations of GLA: set a as parent of b (**left**), change the parent of b from c to a (**middle**) or merge two components a and b into one (**right**). The major arc along which the transformation occurs is depicted in red, while other arcs that affect the transformations 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.

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. 5.4. The change in objective (5.4) 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 (5.3)), 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.

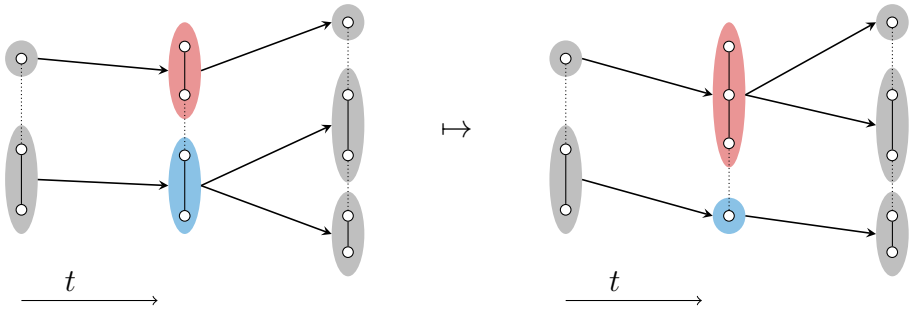


Figure 5.5: Depicted above is a transformation carried out by the KLB algorithm. One node in the middle image is moved from the blue component to the red component. Consequently, the optimal branching changes.

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 $\mathcal{A}(y)$.

Algorithm 2 KL with Optimal Branchings (KLB)

```
while progress do
    for a, b ∈ 𝒱 do
        if ∄uv ∈ E_t : u ∈ V_a ∧ v ∈ V_b then
            continue
        end if
        improveBipartition(𝒢, a, b)        ▷ move nodes across border or merge.
    end for
    for a ∈ 𝒱 do
        splitPartition(𝒢, a)               ▷ split partition.
    end for
end while
return cutEdgeLabels(𝒢)                    ▷ cut-edge labeling x* from 𝒱 and 𝒜(y*).
```

5.3.2 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 (5.4) in each step by changing the intra-frame partitions in a Kernighan-Lin-fashion [155], an example is illustrated in Fig. 5.5. Like the algorithm proposed by [95] 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} (5.4):

$$f_{\text{MLTP}}(x) = \sum_{e \in \bigcup_{t \in \mathcal{T}} E_{t,t+1}} c_e \quad + \sum_{e \in \bigcup_{t \in \mathcal{T}} E_t} c_e x_e + f_{\text{MCBP}}(x) \ , \tag{5.8}$$

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 $\mathcal{G} = (\mathcal{V}, \mathcal{A})$:

$$\min_{y, y^-, y^+} \quad \sum_{ab \in \mathcal{A}} c_{ab} y_{ab} + \sum_{a \in \mathcal{V}} c_a^+ y_a^+ + \sum_{a \in \mathcal{V}} c_a^- y_a^- \tag{5.9}$$

$$\text{subject to} \quad \forall a \in \mathcal{V} \ : \ (1 - y_a^+) = \sum_{b \in \delta^+(a)} y_{ba} \tag{5.10}$$

$$\forall a \in \mathcal{V} \ : \ (1 - y_a^-) \leq \sum_{b \in \delta^-(a)} y_{ab} \leq 2 \tag{5.11}$$

$$y \in \{0,1\}^{\mathcal{A}}, \quad y^-, y^+ \in \{0,1\}^{\mathcal{V}} \ , \tag{5.12}$$

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 (5.9) is exactly f_{MCBP} of (5.8). Each y_{ab} indicates whether arc ab is active ($y_{ab} = 1$) or not ($y_{ab} = 0$). The equality constraint (5.10) 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, (5.11) 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 (5.9)-(5.11) 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. 5.3. 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 5.3.1. 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.

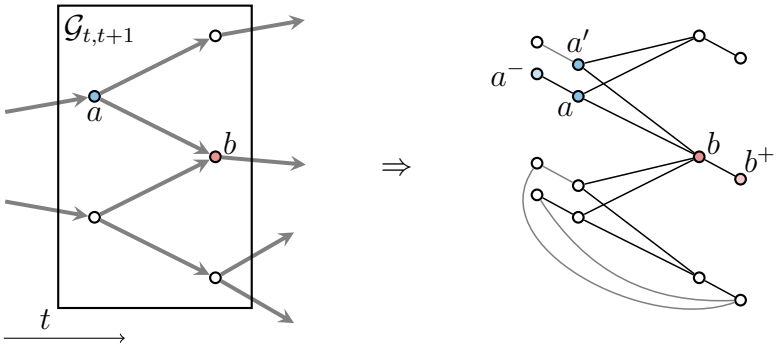


Figure 5.6: Illustration of the constructed bipartite matching problem (**right**) for an MCBP in the subgraph of two consecutive frames $t, t+1$ (**left**). The matching problem graph consists of the original nodes and edges, duplicates a' for $a \in \mathcal{V}_t$, auxiliary termination nodes a^- and 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 $\mathcal{G}_{t,t+1}$.

Proof. The constraints (5.10) only couple birth variables y_a^+ for $a \in \mathcal{V}_{t+1}$ with arc variables y_{ba} where $ba \in \mathcal{A}_{t,t+1}$. Similarly, the constraints (5.11) 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 5.3.2. 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. 5.6): 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 (5.10). 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 (5.11). 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 (5.11), 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 [156, 157]. 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 (5.8), 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 5.3.1, 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.

5.4 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 (5.1) – (5.3), (5.6) – (5.7) 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 (5.1) and, furthermore, it is well-known that chordless cycle inequalities are facet-defining for multicut polytopes (cf. [120] and [154]). This argument can be analogously transferred to inequalities (5.2) and (5.3).

Moreover, the inequalities of (5.3) 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 5.4.1 shows that with both results combined, we can equivalently replace (5.1) – (5.3) by the set of tighter inequalities (5.13) and (5.14). Proofs are provided in the supplementary material. In relation to our improved version of the branch-and-cut algorithm, we refer to (5.13) as *cycle* and to (5.14) as *morality* constraints.

Lemma 5.4.1. For every hypothesis graph $G = (V, E)$ it holds that $x \in X'_G$ iff $x \in \{0, 1\}^E$ and x satisfies

$$\begin{aligned}
&\forall t \in \mathcal{T} \forall \{v, w\} \in E_t \cup E_{t,t+1} \\
&\forall \text{ chordless } vw\text{-paths } P \text{ in } G_t^+ : \\
&\qquad x_{vw} \leq \sum_{e \in P} x_e \qquad (5.13) \\
&\forall t \in \mathcal{T} \forall v', w' \in V_t \text{ such that } \{v', w'\} \notin E_t \\
&\forall v'w'\text{-cuts } S \text{ in } G_t \forall \text{ chordless } v'w'\text{-paths } P \text{ in } G_t^+ : \\
&\qquad 1 - \sum_{e \in S} (1 - x_e) \leq \sum_{e \in P} x_e \qquad (5.14)
\end{aligned}$$

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 (5.14) 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* [154]. 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 5.4.2. For every hypothesis graph $G = (V, E)$, the vectors $x \in X'_G$, $x^+, x^- \in \{0, 1\}^V$ satisfy inequalities (5.6) iff the following inequalities hold:

$$\begin{aligned} &\forall t \in \mathcal{T} \forall v \in V_{t+1} \forall S \in V_t v\text{-cuts}(G_t^+) : \\ &1 - x_v^+ \leq \sum_{e \in S \setminus E(V_t(v), V_{t+1} \setminus \{v\})} (1 - x_e). \end{aligned} \tag{5.15}$$

Similarly, $x \in X'_G$, $x^+, x^- \in \{0, 1\}^V$ satisfy (5.7) iff

$$\begin{aligned} &\forall t \in \mathcal{T} \forall v \in V_t \forall S \in v V_{t+1}\text{-cuts}(G_t^+) : \\ &1 - x_v^- \leq \sum_{e \in S \setminus E(V_t \setminus \{v\}, V_{t+1}(v))} (1 - x_e) \end{aligned} \tag{5.16}$$

hold true.

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

$$\sum_{e \in E_C} x_e - \sum_{e \in E_w} x_e \leq \left\lfloor \frac{|V(W)| - 1}{2} \right\rfloor, \tag{5.17}$$

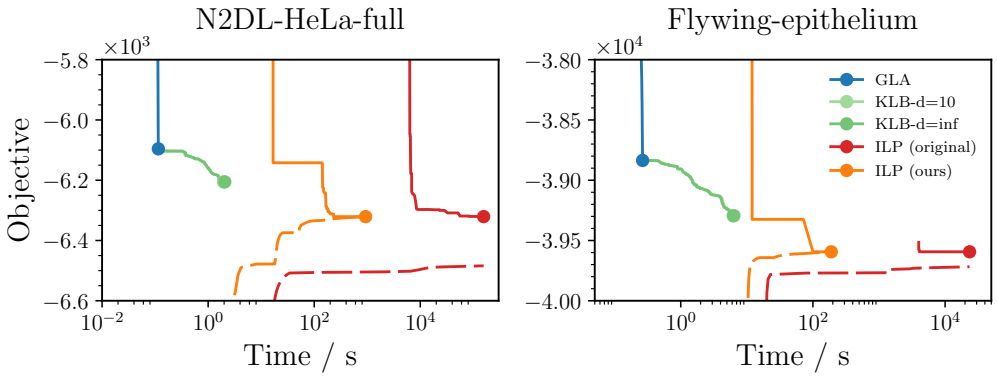


Figure 5.7: Comparison of algorithms for the MLTP in terms of runtime, objective (solid) and bounds (dashed) on the large instances of [86]. Our heuristics are able to determine feasible solutions quickly, while our branch-and-cut algorithm (ILP ours) converges to the optimal solution in up to one hundredth 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}.

which is valid for multicut polytopes [120]. 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 [120].

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, (5.17) holds with respect to x.

Implementation. For a subset of the constraints, we use the commercial branch-and-cut solver Gurobi (7.0) [117] 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 (5.13) – (5.16) as well as violated bifurcation constraints and repeat. To this end, we adapt the separation procedures of [86] 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.

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.

5.5 Experiments & Results

Instances and Setup. We evaluate our algorithms on the two large instances of [86]: *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 [86].

Our choice of birth and termination costs follows [86], 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 $d_{\text{MCBP}} = 10$.

Convergence Analysis. The convergence of our algorithms in comparison to the branch-and-cut algorithm of [86] is reported in Fig. 5.7 and Table 5.1. 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 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.

Table 5.1: 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 $d_{\text{MCBP}} = 10$.

	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-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) [86]	23460.80	-39593.90	-39717.80	0.0031	0.0000	156542.00	-6320.81	-6484.02	0.0258	0.0000

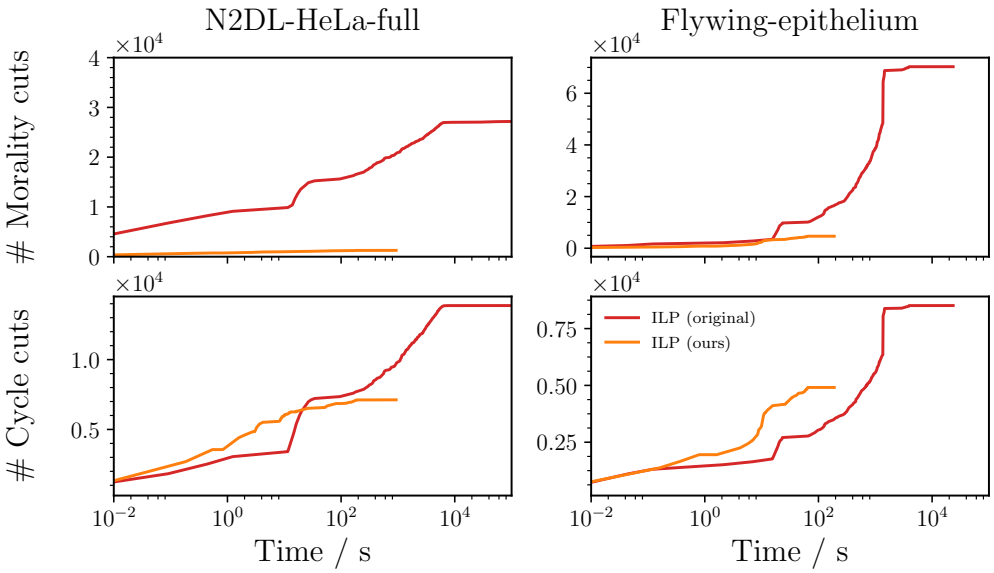


Figure 5.8: Number of morality cuts (**top**), i.e. (5.3) or (5.14), and cycle cuts (**bottom**), i.e. (5.1) and (5.2) or (5.13), separated in the different branch-and-cut algorithms. We observe that our branch-and-cut algorithm requires considerably fewer morality cuts, while the number of cycle cuts (including both space-cycles and space-time-cycles) is in the same order of magnitude.

The improved branch-and-cut algorithm retrieves feasible solutions considerably faster and provides tighter bounds than the algorithm of [86]. 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. 5.8, 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. 5.9. 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 $d_{\text{MCBP}} = 10$ was found to be stable in both cases. More extensive

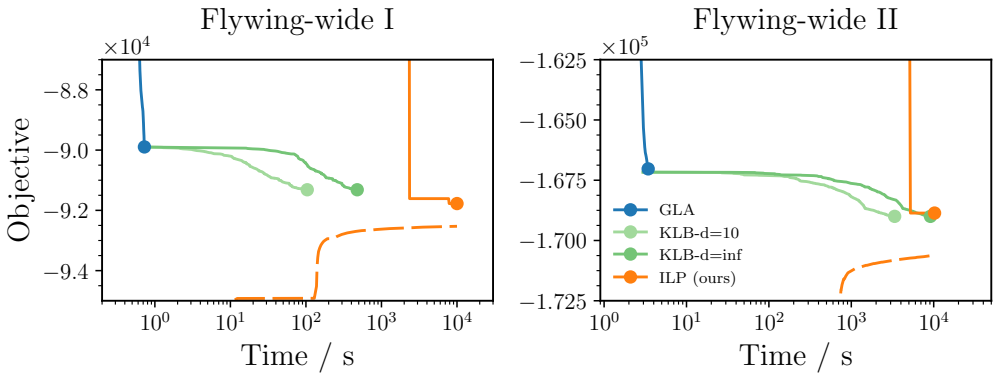


Figure 5.9: Results on the more extensive instances *Flywing-wide I* and *II*. Our branch-and-cut algorithm with 3-wheel constraints provides slightly tighter bounds, with which we determine the gaps for GLA to be 2.9 % (*I*) and 2.1 % (*II*), and 1.3 % (*I*) and 0.95 % (*II*) for KLB. Exploiting spatial locality when re-solving the MCBPs considerably reduces runtime of KLB.

Table 5.2: 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 [158] 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

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 [78] for *Flywing-epithelium*. The results are reported in Table 5.2. 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* [158], whose scores were originally reported in [86].

5.6 Conclusion

We have introduced local search algorithms for the recently introduced MLTP [86], 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 [86] 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 [21]).

6

Cell Lineage Tracing in Lens-free Microscopy Videos

This chapter has been published as **peer-reviewed conference paper**.

M. Rempfler, S. Kumar, V. Stierle, P. Paulitschke, B. Andres, and B. H. Menze. "Cell Lineage Tracing in Lens-Free Microscopy Videos." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part II*. ed. by M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne. Cham: Springer International Publishing, 2017, pp. 3–11. DOI: 10.1007/978-3-319-66185-8_1

Synopsis: Lens-free microscopy (LFM) is a low-cost technology particularly suited for continous monitoring of cell cultures in-vitro. In this work, we propose (i) a cell detector for LFM videos based on fully convolutional residual networks, and (ii) a probabilistic model based on moral lineage tracing (MLT) which clusters detections and tracks cells through time simultaneously.

Contributions of thesis author: Algorithm design and implementation, computational experiments and interpretation, composition of manuscript.

Abstract

In vitro experiments with cell cultures are essential for studying growth and migration behaviour and thus, for gaining a better understanding of cancer progression and its treatment. While recent progress in lens-free microscopy (LFM) has rendered it an inexpensive tool for continuous monitoring of these experiments, there is only little work on analysing such time-lapse sequences.

We propose (1) a cell detector for LFM images based on residual learning, and (2) a probabilistic model based on *moral lineage tracing* that explicitly handles multiple detections and temporal successor hypotheses by clustering and tracking simultaneously. (3) We benchmark our method on several hours of LFM time-lapse sequences in terms of detection and tracking scores. Finally, (4) we demonstrate its effectiveness for quantifying cell population dynamics.

6.1 Introduction

Cell growth and migration play key roles in cancer progression: abnormal cell growth can lead to formation of tumors and cancer cells can spread to other parts of the body, a process known as metastasis. In vitro experiments are essential to understand these mechanisms and for developing anti-cancer drugs. In these experiments, the cells are typically observed with conventional light microscopes. Thanks to recent advances in CMOS sensor technology, LFM [99, 100] has become a promising alternative. In LFM a part of the incident wavefront originating from the light source is scattered by the sample, in this case the cell. The scattered light then interferes with the unscattered part of the wavefront and the resulting interference pattern is recorded with a CMOS sensor. The components required for LFM are extremely small and cheap. Thus, LFM provides the means for a wide range of applications where a conventional light microscope would be either too big or simply too expensive, such as the continuous monitoring of growing cell cultures inside standard incubators [101].

To quantify the clinically relevant information on cell growth and migration from the large amount of images that are acquired in such continuous

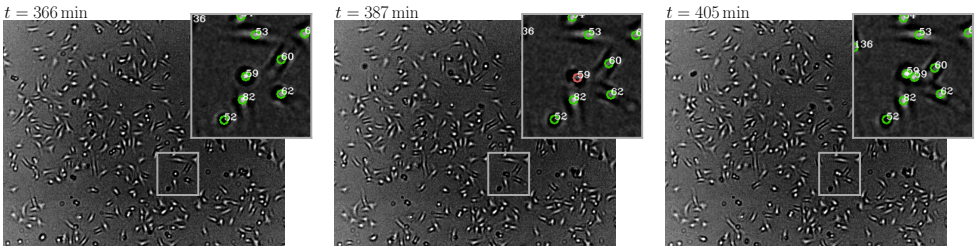


Figure 6.1: The cell lineage tracing problem with LFM data. We aim to detect all cells and establish their relation over time, i.e. determine the lineage forest. While the LFM technology allows for frequent image acquisition (3 min / frame in this case), challenges arise due to overlapping interference patterns of close objects, fluctuating shape and size of the cells appearance, and particles that generate similar patterns as the cells. The detail views show cell locations as a circle and identify their lineage tree.

monitoring, reliable automatic image analysis methods are crucial. Counting the number of cells in a time series of images gives access to the dynamics of cell growth. Locating and tracing individual cells provides information about cell motility, and over the course of a sequence, reconstructing the lineage trees gives insights into cell cycle timings and allows more selective analysis of cell sub-cultures.

There are several works on these tasks in traditional light microscopy, e.g. focussing on cell segmentation [159], detection and counting [103, 160, 58] or tracking [161, 162, 86, 78, 105], but very few deal with LFM data. One of the few exceptions is [163] which employs a regression framework for estimating the total cell count per image. We aim at the more complex goal of not only counting but also localizing cells and reconstructing their spatio-temporal lineage forest (c.f. Fig. 6.1). Methods for the latter task range from Kalman filtering [161] to keep track of moving cells, or iteratively composing tracklets by using the Viterbi algorithm [164], and have been compared in [78]. More recently, Jug et al. [86] have proposed a mathematically rigorous framework for lineage reconstruction, the so-called MLTP. The MLTP differs fundamentally from all mathematical abstractions of cell tracking whose feasible solutions are either disjoint paths or disjoint trees of detections. Unlike these approaches that select only one detection for each cell in every image, feasible solutions of the MLTP select and cluster

an arbitrary set of such detections for each cell. This renders the lineage trees defined by feasible solutions of the MLTP robust to the addition of redundant detections, a property we will exploit in this work.

In this paper, we contribute a framework for analysis of LFM sequences. First, we design and benchmark robust cell detectors for LFM time-lapse sequences derived from most recent work on CNNs and residual learning. Second, we discuss the MLTP in the context of LFM data. In particular, we define a probability measure for which the MLTP is a MAP estimator. This allows us to define the costs in the objective function of the MLTP w.r.t. probabilities that we estimate from image data. We validate it experimentally on two annotated sequences. Finally, we demonstrate the capability of our approach to quantify biologically relevant parameters from sequences of two in vitro experiments with skin cancer cells.

6.2 Methods

We consider the lineage tracing task as a MAP inference over a hypothesis graph containing a multitude of potential lineage forests. We discuss the probability measure and its MAP estimator, the MLTP in Section 6.2.1. In order to construct the hypothesis graph from a sequence of LFM images, we devise a cell detector in Section 6.2.2, which estimates a cell probability map for each given image. The workflow is illustrated in Fig. 6.2.

6.2.1 Lineage Tracing

Hypothesis graph.

We construct a spatio-temporal hypothesis graph $G = (V, E)$ as follows: For every image I_t in the sequence, we apply a cell detector and define one node $v \in V_t$ for every local maximum in $P(c_s = 1|I_t)$, the estimated probability map for finding a cell at a particular location s in image I_t. Additionally, we define *hypothesized* successors to each node that has one or more favourable parents in the previous frame but no immediate successor. This helps avoiding gaps in the final tracklets. The nodes $v \in V$ represent cells, yet do not need to be unique, i.e. one cell may give rise to several nodes. We then construct edges in space $E_t^{\text{sp}} = \{uv \in V_t \times V_t : d(u, v) \leq d_{\max}\}$, i.e. between any two nodes that lie within a distance of $d_{\max}$, and in the same fashion, we

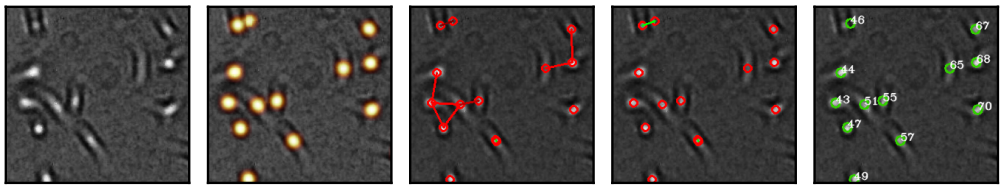


Figure 6.2: Illustration of our workflow. From left to right: 1) Raw microscopy image, 2) image overlayed with cell probability map generated by the detector, 3) nodes of the hypothesis graph with spatial edges constructed from cell probabilities, 4) optimized lineage where spatial edges that were cut are removed, and 5) each cluster is represented as one cell with its lineage tree identifier. Temporal edges are not depicted for simplicity.

construct temporal edges $E_t^{\text{tmp}} = \{uv \in V_t \times V_{t+1} : d(u,v) \leq d_{\max}\}$ between nodes in adjacent frames.

Probabilistic model.

We introduce a family of probability measures, each defining a conditional probability of any lineage forest, given an image sequence. We describe the learning of this probability from a training set of annotated image sequences as well as the inference of a maximally probable lineage forest, given a previously unseen image sequence. The resulting MAP estimation problem will assume the form of an MLTP with probabilistically justified costs.

First, we encode subgraphs in terms of cut edges with binary indicator variables $\mathbf{x} \in \{0,1\}^E$. If edge uv is cut, i.e. $x_{uv} = 1$, it means that nodes u and v do *not* belong together. In order to ensure that the solution describes a lineage forest, we rely on the formulation of the MLTP [86], which describes the set of inequalities that are required to do so. In short, these constraints ensure: 1) *spatial* and *temporal consistency*, i.e. if nodes u and v as well as v and w belong together, then u and w must also belong together. 2) Distinct tracklets cannot merge at a later point in time. These are the so called *morality* constraints. 3) *Bifurcation* constraints allow cells to split in no more than two distinct successors. We will denote the set of $\mathbf{x}$ that describe valid lineage forests with X_G. For a more extensive discussion of these constraints, we refer to [86, 105]. We next model the measure of

probability:

$$P\left(\mathbf{x}|X_G,\boldsymbol{\Theta}\right) \propto P\left(X_G|\mathbf{x}\right) \prod_{uv\in E} P\left(x_{uv}|\boldsymbol{\Theta}\right) \prod_{v\in V} P\left(x_v^+|\boldsymbol{\Theta}\right) \prod_{v\in V} P\left(x_v^-|\boldsymbol{\Theta}\right), \tag{6.1}$$

$$\text{where } P\left(X_G|\mathbf{x}\right) \propto \begin{cases} 1 & \text{if } \mathbf{x} \in X_G, \\ 0 & \text{otherwise} \end{cases}. \tag{6.2}$$

It is comprised of four parts. First, we have $P\left(X_G|\mathbf{x}\right)$ representing the uniform prior over X_G. Second, the cut probability $P\left(x_{uv}|\boldsymbol{\Theta}\right)$ describing the probability of u and v being part of the same cell (either in space or along time), and third and fourth, the birth and termination probabilities $P\left(x_v^+|\boldsymbol{\Theta}\right)$ and $P\left(x_v^-|\boldsymbol{\Theta}\right)$ for each node $v \in V$. The variables $x_v^+, x_v^- \in \{0,1\}$ are indicating whether the respective event, birth or termination, occurs at node v. $\boldsymbol{\Theta}$ denotes the joint set of parameters. We use these parts to incorporate the following assumptions: Two detections u and v that are close are more likely to originate from the same cell, hence we choose $P\left(x_{uv}=1|\boldsymbol{\Theta}\right) = \min(\frac{d(u,v)}{\theta^{\text{sp}}}, 1)$. Similarly, two successive detections u at t and v at $t+1$ are more likely to be related the closer they are, is captured by $P\left(x_{uv}=1|\boldsymbol{\Theta}\right) = \min(\frac{d(u,v)}{\theta^{\text{tmp}}}, 1)$. Finally, we assume that birth and termination events occur at a low rate, which is incorporated by $P\left(x_v^+=1|\boldsymbol{\Theta}\right) = \theta^+$ and $P\left(x_v^-=1|\boldsymbol{\Theta}\right) = \theta^-$. We fit these parameters $\boldsymbol{\Theta}$ on training data in a maximum likelihood fashion: For θ^- and θ^+ this boils down to calculating the relative frequency of the respective events on the annotated lineage. For the spatial and temporal parameters θ^{sp} and θ^{tmp}, we first complement the lineage forest with edges within $d_{\max}$ as $\mathcal{E}$. We then maximize the log-likelihood $\log \mathcal{L}(\theta) = \sum_{uv\in\mathcal{E}} \log P\left(x_{uv}|\theta\right)$ by an extensive search over the interval $\theta \in [\theta_{\min}, \theta_{\max}]$, where we found $[1, 80]$ to be appropriate.

The MAP estimate $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{X}} P\left(\mathbf{x}|\boldsymbol{\Theta}, X_G\right)$ can be written as solution to the MLTP:

$$\min \left\{ \sum_{uv\in E} c_{uv} x_{uv} + \sum_{v\in V} c_v^+ x_v^+ + \sum_{v\in V} c_v^- x_v^- \;\middle|\; \mathbf{x} \in X_G \cap X_V \right\}, \tag{6.3}$$

where the coefficients become $c_{uv} = -\log \frac{P(x_{uv}=1|\boldsymbol{\Theta})}{1-P(x_{uv}=1|\boldsymbol{\Theta})}$ for edges, and vice versa for c_v of the node events. X_V is the set of $\mathbf{x}$ that satisfy the auxiliary constraints which tie birth and termination indicator variables x_v^- and x_v^+

to the respective edge variables. We optimize (6.3) with the KLB algorithm described in [105].

6.2.2 Cell Detection with Residual Networks

Cells in LFM images are usually only marked at their center of mass and not segmented since their interference pattern, i.e. their appearance in the image, does not accurately describe their true shape and would therefore be ambiguous in many cases. Thus, we are interested in a detector that outputs the set of cell centers in image I_t. Strong performance of the detector is crucial for the lineage reconstruction as its errors can affect the final lineage trees over many frames. To achieve this, we build on the recent work on residual networks [102]. However, instead of directly regressing bounding boxes or center coordinates in a sliding window fashion, we train our network, denoted with $f(I_t)$, on a surrogate task: We approximate $f(I_t) \approx P\left(c_s = 1 | I_t\right)$, the probablity map of finding a cell at a particular location s in I_t. This detector is fully convolutional and its output $f(I_t)$ has the same size as I_t. We found this to facilitate the training as it enlargens the spatial support of the sparse cell center annotations and gracefully handles the strongly varying cell density. Similar findings were made with techniques that learn a distance transform to detect cells, e.g. in [103]. We describe next how we arrive at a suitable architecture for this task and how to construct $P\left(c_s = 1 | I_t\right)$ from point-wise cell annotations.

Network Architecture. We start from the architecture of *ResNet-50* [102]. We first truncate the network at layer 24 to obtain a fully convolutional detector. We found that truncating in the middle of the original ResNet-50, i.e. at layer 24, results in best resolution of the output response maps and allows to distinguish close cells. We then add one convolutional layer of $1 \times 1 \times 256$ and one up-convolutional layer (also known as deconvolutional layer) of $8 \times 8 \times 1$ with a stride of 8. The former combines all feature channels, while the latter compensates for previous pooling operations and ensures that the predicted cell probability map has the same resolution as the input image I_t. Finally, a sigmoid activation function is used in the last layer to ensure that $f(I_t)$ is within the interval $[0, 1]$ at any point.

Loss Function & Training. We sample training images of size 224×224 from all frames of the training corpus. For each training image I_k, we construct a corresponding cell probability map $P\left(c_s = 1 | I_k\right)$ by placing a Gaussian kernel G^σ with $\sigma = 8$ at each annotated center. This implicitly

represents the assumption that all cells have about the same extent, which is reasonable for our microscopy data. Each image is normalized to zero mean and unit variance. During training, we minimize the cross entropy loss between the predicted map $f(I_t)$ and $P(c_s|I_t)$ in order to let our network approximate the constructed cell probability map. We fine tune the network (pre-trained weights from ResNet-50 [102]) with a learning rate of 10^{-3} for 100 epochs with batch size of 8. In each epoch, we sample 4000 training images. Since the annotated dataset for training is typically small and shows strong correlation between cells in consecutive frames, we used dropout of 0.5 after the last convolutional layer to avoid overfitting.

6.3 Experiments & Results

Datasets. We use a sequence depicting A549 cells, annotated over 250 frames in a region of interest (ROI) of 1295×971 px, for all training purposes. For testing, we annotated two distinct sequences monitoring 3T3 cells of 350 and 300 frames in a ROI of 639×511 px (*3T3-I*) and 1051×801 px (*3T3-II*), respectively. Images were acquired at an interval of 3 min with $1.4\,\mu\text{m} \times 1.4\,\mu\text{m}$ per pixel.

Benchmarking detectors. We compare four different network configurations, including the described ResNet-23, ResNet-11, a variant of it which was truncated at layer 11, the UNet [159] and CNN-4. In UNet, we obtained better results when replacing the stacks in the expansive path with single up-convolution layers which are merged with the corresponding feature maps from contracting path. CNN-4 is a plain vanilla CNN with three 5×5 convolutional layers followed by max pooling and finally, one up-convolutional layer of $8 \times 8 \times 1$ to compensate for the down-sampling operations. We use the same training procedure (Section 6.2.2) for all detectors, but adjust the learning rate for UNet and CNN-4 to 10^{-2}.

We match annotated cells to detections within each frame with the hungarian algorithm and consider only matches closer than 10 px ($\approx$ a cell center region) as a true positive (TP). Unmatched annotations are counted as false negative (FN), unmatched detections as false positive (FP). The results are presented in Fig. 6.3, where we find the ResNet-23 to be the most robust detector.

Lineage tracing. To compare the quality of different lineages, we match again annotations and detections within each frame to calculate the number

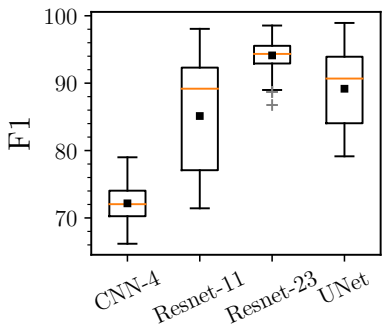


Figure 6.3: Performance of different detectors over all test frames. Boxplots depict median as orange line, mean as black square and outliers as grey +. The F1 scores are shown in %. We find that ResNet-23 is the most robust detector in our experiment with an average F1 of 94.1 %. It is followed by the UNet with 89.2 %, ResNet-11 with 85.1 % and finally, CNN-4 with only 72.2 %.

of TP, FP and FN as described before. We then determine the number of false links, i.e. how often two matched nodes do not have the same parent. From these, we calculate multiple object detection accuracy (MODA) and multiple object tracking accuracy (MOTA) [165]. Moreover, we derive the number of edit operations needed to get from the predicted lineage to the ground truth lineage, and calculate the tracking accuracy (TRA) score proposed in [78]. We use unit weight for each type of edit (add or delete node or edge). This is justified by the fact that we have point annotations for cells instead of segmentations, making both addition and deletion equally expensive to correct.

For the MLTP, we compare the effect of varying θ^{tmp}, θ^{sp} together with hypothesis graphs generated from the different detectors in Fig. 6.4. The optimal parameter choice for ResNet-23 is at 10, i.e. a relatively small merge radius of favourable merges, while the other detectors considerably benefit from wider ranges. In Table 6.1, we compare different lineage tracing approaches. Our baseline is linear assignment problem tracking (LAPT) [162]. The disjoint trees method (DTP), uses our ResNet-23 detections but solves the disjoint trees problem instead, i.e. it considers only one detection per cell. We find that MLTP outperforms both in terms of detection and tracking metrics.

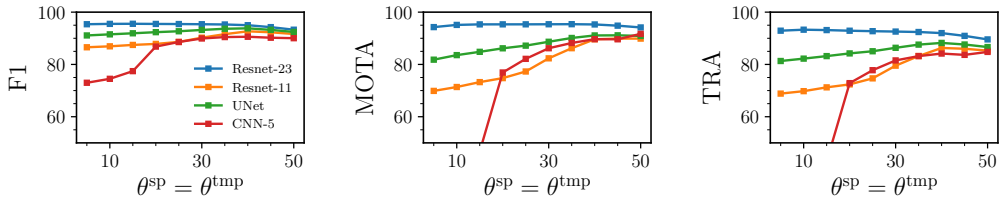


Figure 6.4: Sensitivity analysis of the lineage tracing model with different detectors. We increase both edge cut parameters θ^{tmp} and θ^{sp} together. While the optimal choice in combination with ResNet-23 is relatively small, i.e. at 10, the other detectors, which suffer from many spurious detections, benefit from a wider range. Most notably, the performance with CNN-4 improves up to a competitive TRA of 84.8 %.

Table 6.1: Quantitative evaluation of traced lineages. Precision, recall, F1 and MODA are averaged over all frames of a sequence, while MOTA and TRA are global scores for the entire lineage forest. All measures are in %. Disjoint trees (DTP) uses our ResNet-23 detections and is equivalent to MLTP with disabled spatial edges and no hypothesized successors. LAPT is linear assignment problem tracking [162] and our baseline.

Instance	Method	Precision	Recall	F1	MODA	MOTA	TRA
3T3-I	LAPT	86.39	88.99	87.63	85.88	83.87	80.46
	DTP	93.67	92.84	93.22	93.67	90.22	87.11
	MLTP	97.09	93.19	95.07	97.18	95.67	92.58
3T3-II	LAPT	85.12	87.35	86.19	84.68	82.65	79.13
	DTP	94.02	95.89	94.93	93.85	91.49	89.87
	MLTP	96.46	96.12	96.28	96.45	95.43	93.76

Assessing cell population dynamics. We apply our method on data from two experiments with skin cancer cells. In each, one population is exposed to an inhibitor substance while the control is not. Figure 6.5 depicts the resulting statistics. We observe the expected difference in growth rate, yet a more constrained motility of the control cells, which is caused by the limited space.

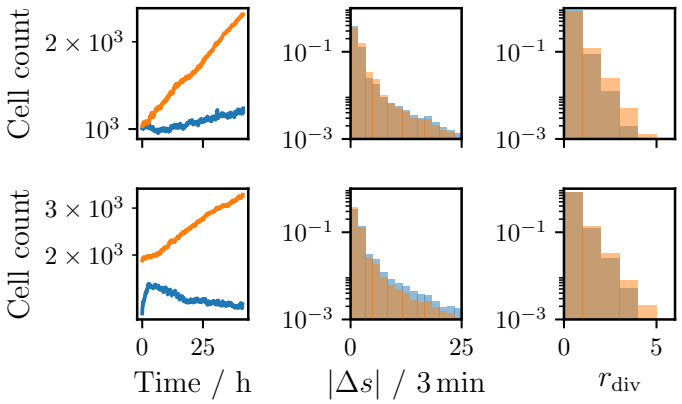


Figure 6.5: Cell dynamics measured on two experiments with skin cancer cell lines. One population (blue) is exposed to an inhibitor substance, while the other (orange) is not. From left to right: Cell count over time, histograms on cell motility (μm/3 min) and divisions r_{div}/h. Cells that divide often are more abundant in the control group.

6.4 Conclusions

We have presented a framework for automatic analysis of LFM time-lapse sequences. It transfers two recently proposed methods, residual learning [102] and moral lineage tracing [86, 105], to the task at hand. We have shown experimentally that it is able to determine cell lineage forests of high quality and thereby quantify several measures of interest for the analysis of in vitro experiments, including cell population dynamics and motility.

Acknowledgements

With the support of the Technische Universität München – Institute for Advanced Study, funded by the German Excellence Initiative (and the European Union Seventh Framework Programme under grant agreement n 291763).

7

Concluding Remarks

With the progress of advanced biomedical imaging techniques and an emerging demand for quantitative methods, computer-aided analysis of biomedical image data has gained considerable importance. Amongst such analysis problems is the task of estimating a graph from observed images or videos – a task we called *graph inference*. Our contributions focus on two prominent forms of the problem, namely vessel networks and cell lineages. Due to the publication-based nature of this thesis, the Chapters 3 to 6 are self-contained and in their original form. This final chapter therefore provides a brief summary as well as a more general discussion of the work, including directions for future research.

First, we have presented a probabilistic model for imposing local statistics into the graph inference step for vascular networks in Chapter 3. This model enables utilizing physiological knowledge gained from high-resolution data, such as the µCT of a corrosion casted cerebral vasculature in our case, when processing image volumes of lower resolution. There are a few remarks on this part of our work. First, as with any method that imposes a prior on the prediction step, one has to be careful to not accidentally suppress actual vasculature that is of surprising shape. Devising a workflow that gradually goes from a uniform to the learned prior and monitoring the changes could be helpful to address this. Second, measuring the overall quality of a network is difficult, which is why we resorted to a comparison based on macroscopic, physiological measures. A theoretically more appealing option would be a metric based on the graph edit distance [166]. This concept is used for single neuron reconstructions [167] or for cell lineages [78, 168], both of which are a special case of the graph edit distance where the graphs are trees. This property renders calculation of the edit distance tractable. For more general classes of graphs, approximative calculation schemes would need to

be explored and validated. That being said, obtaining a ground truth vessel network as reference graph is a laborious task to begin with. For many potential use cases, it is probably recommendable to devise a metric that is specific to the particular application. Such a specific metric is most likely simpler to annotate for and easier to interpret in the end, but is also the key step to adjusting the method to the specific needs, potentially saving valuable computational resources and time. Regarding computation time, whenever connectedness constraints are part of the model, we strongly recommend to use the branch-and-cut algorithm with a more efficient constraint generation scheme developed Chapter 4, which is considerably faster.

In Chapter 4, we have proposed and benchmarked different constraint generation strategies for solving the MCCSP with a branch-and-cut algorithm. One of the key findings was that adding multiple violated constraints at once improved scalability of the branch-and-cut algorithm. This enabled a benchmark on larger instances than before, i.e. on pixel-grid graphs originating from segmentation problems, which led to our second key finding of this chapter: the approximative formulation of connectedness along the geodesic tree (proposed by [66]) yields solutions of high quality while providing polynomial complexity. Thus, it is considerably faster than exact branch-and-cut solvers, even with improved constraint generation strategies. While our experiments focussed on MCCSPs with unary costs, the constraint generation strategies are also applicable in the presence of higher-order cost functions, since adding higher-order terms does not alter the subset of feasible connected graphs. In fact, Shen et al. [169] apply one of the proposed strategies (k-nearest) with both unary and binary potentials even in a multiclass setting, i.e. a disjoint connected subgraph for each foreground class is determined. In a scenario where higher-order terms become the dominant part of the MCCSPs objective function, however, it is no longer clear whether the geodesic tree formulation would still perform well. Because of higher-order terms being more difficult to capture when calculating the geodesic tree, this tree might not be a good approximation to the actual connectedness anymore, in which case the quality in comparison to an exact formulation of connectedness would suffer. A potential approach for handling such difficult cases could be a learned surrogate geodesic (along which the tree is calculated), e.g. in the form of a classifier that estimates the most likely parent for each node. This would be closely related to the work of [170], where a CNN is trained to estimate a topological distance used within a watershed algorithm.

Encouraged by the results on constraint generation strategies in MCCSPs, we moved our focus to the MLTP [86], a joint clustering and tracking formulation for cell lineage tracing. Its originally proposed branch-and-cut algorithm suffered from many cuts and thus, slow convergence on large instances. We addressed this in Chapter 5 by two contributions. One of them is the improvement of the original branch-and-cut algorithm by separating thighter cutting planes. In contrast to the constraint generation strategies in Chapter 4, this not only changes the order and/or amount of the added inequalities, but results in a more compact representation of feasible solutions. In addition to this, we devised the two first primal feasible search heuristics for the MLTP. These heuristics are used either within the branch-and-cut algorithm or for instances that are simply too large for exact optimization algorithms due to their weaker scalability. A line of future research could extend these heuristics to variants of the MLTP, e.g. including separate node weights analogously to the MCMCP variant discussed in [153], or in the presence of long-range weights in the form of so-called lifted edges as in [95, 154, 37]. Another potentially useful modification of the MLTP for which the heuristics would need to be adjusted is the addition of weights for particular events, such as repeated division within (short) time intervals similar to one of the models in [82].

Already during our work on optimizers for the MLTP, we found the cell lineage tracing task in LFM to be an interesting use-case. LFM is a low-cost microscopy technique, which is well-suited for continuous live cell monitoring due to being label-free. However, the lineage tracing task in LFM videos had received almost no attention. Hence, we aimed at filling this niche. We proposed a framework using FCNs for detecting cells in combination with a probabilistic model to estimate the cell lineage for which the MLTP becomes its MAP estimator. In our experiments, we demonstrated the advantages of using the MLTP instead of traditional disjoint tree problem (DTP) formulations, and showed that this framework is able to generate lineages of high quality. Considering the high diversity of data that can be generated from LFM, e.g. through different cell lines or different densities, future work certainly needs to conduct an extended study on a larger corpus of sequences. In fact, such study is ongoing in [171]. Besides such an extended validation, future research could improve parts of the current approach. The probabilistic model could include any of the above mentioned extensions to the MLTP such as lifted edges or event weights. Another option is replacing the relatively simple cut weight model with a discriminatively

trained classifier that distinguishes nodes of the same entity (i.e. the same cell) from those of different entities. This has been proposed in the context of pedestrian tracking [172, 37]. These approaches should be transferable to our setting, even though they most likely require adjustments to account for the particularities of cells in LFM. Spatial proximity and low-level features like edges, for example, are expected to be much more important for distinguishing individual cells than for re-identifying pedestrians that relies strongly on high-level visual features.

As many other fields, recent progress in deep learning has strongly influenced the latest approaches in medical image analysis. For example, the majority of methods participating in the most recent editions of challenges for brain tumor segmentation (BRATS) [173] or liver tumor segmentation (LITS) [129] employ CNNs or FCNs, at least at some stage of their pipeline. This applies also for (sub-)tasks in graph inference. CNNs and FCNs are employed as detection or segmentation methods for vasculature [174, 175] and cells [159, 176, 177]. These more complex models, trained with a sufficient amount of data, provide stronger *unary* potentials for the model for graph inference. This improves the overall quality and robustness of detections and can render it easier to determine an optimal or near-optimal solution to the joint problem. The latter is particularly helpful when working with MAP estimators for graph inference that take the form of an ILP and are then optimized by a branch-and-cut algorithm, where it can save valuable computation time. All in all, it is to be expected that the trend continues: adapting the next generation of machine learning and computer vision methods for detection and segmentation to the biomedical case and its particularities (e.g. the difficulty to obtain larger amounts of annotated data and a typically lower consensus between annotators) will most likely remain a main direction of research that will improve graph inference approaches.

Beyond that, an emerging sub-field of research is *geometric deep learning* [178], which aims at transferring the concepts of deep learning (in particular those of CNNs) to non-euclidean domains. This includes learning on graphs [179, 180, 181, 182, 183], variants of which are starting to find application in medical image computing as well [184, 185]. In our context of graph inference from biomedical image data, two aspects are particularly interesting. First, the subgraph estimation can be considered a (constrained) labeling problem on the hypothesis graph, which could be performed with an adapted version of a graph CNN. Effectively, such graph CNN could be trained to approximate the subgraph labeling of the combinatorial opti-

mization algorithm. If successful, this could save computation time during inference and potentially enable an efficient joint-training of the model's components as the approximated final inference step could be included in an end-to-end training scheme. Depending on feasibility and implementation, it could even provide the means to identify valid constraints from a large training dataset of (annotated) graphs. However, it is not clear how, to which degree, and at which computational complexity it is possible to approximate topological constraints like connectedness or restriction of higher-order furcations with such approaches. This is perhaps a main question for future research on graph inference in biomedical image analysis.

Appendices

A

Supplementary Material to the Minimum Cost Connected Subgraph Problem in Medical Image Analysis

This chapter has been published as **supplementary material to** [21].

M. Rempfler, B. Andres, and B. H. Menze. "The Minimum Cost Connected Subgraph Problem in Medical Image Analysis." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III*. ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Cham: Springer International Publishing, 2016, pp. 397–405. DOI: 10.1007/978-3-319-46726-9_46

A.1 Additional Experimental Results

Details on the *objective-dependent constraints* experiment with 25 subsampled 64×64 problem instances are given in Table A.1 and Fig. A.1.

For the full datasets, additional experimental results *per instance* for the different constraint generation strategies are shown in Table A.2 and Table A.3. The gap is calculated as $\text{gap} = 1 - \frac{\text{objective}}{\text{bound}}$ and shown in percent. Note that two strategies might obtain solutions with identical objective

A. Supplementary Material to the Minimum Cost Connected Subgraph Problem in Medical Image Analysis

Table A.1: Number of solved instances (#S), fastest to solve (#F) and instances with gap $\leq 1\,\%$ per strategy.

	without			with		
	#S	#F	$\#\epsilon_1$	#S	#F	$\#\epsilon_1$
Equidistant	19	2	19	19	4	19
k-Nearest	21	13	21	21	8	21
k-Interleave	19	3	20	21	7	21
Minimal	11	0	11	11	0	15
Interleave	12	2	13	14	1	14
Nearest	10	1	10	11	1	14

value, yet establish different bounds. We limited runtime to maximum 4 h and 8 h for DRIVE and OPF, respectively.

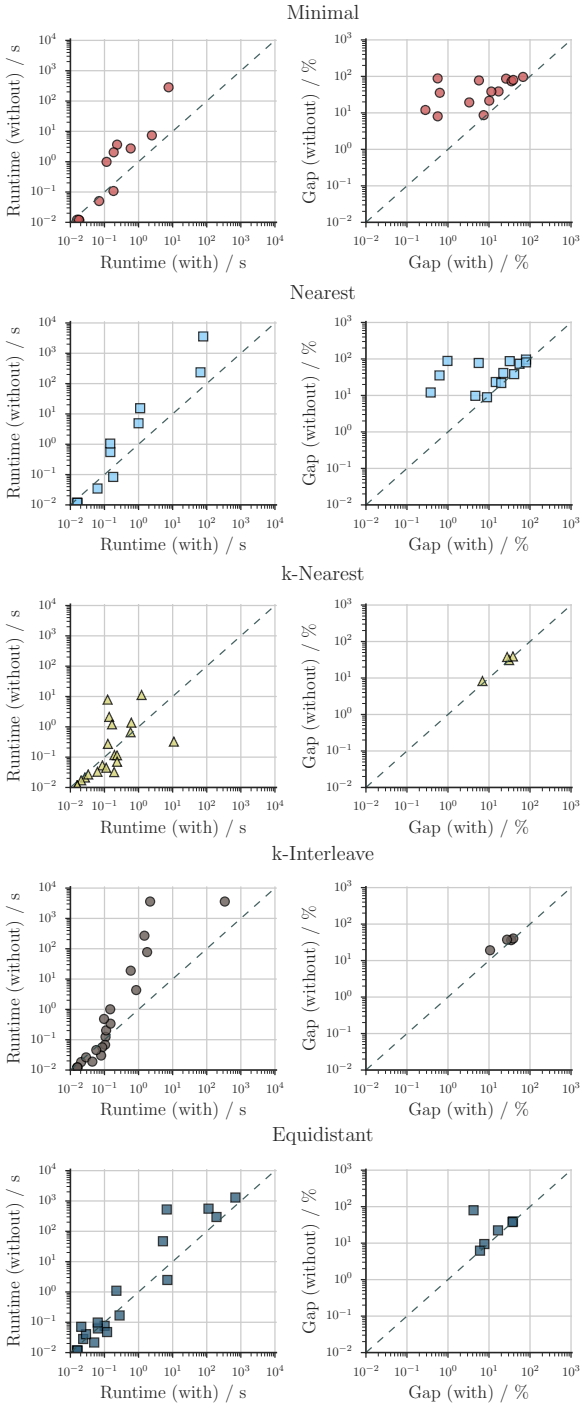


Figure A.1: Comparison of the objective-dependent constraints on the small problem instances. **Left:** Runtime with objective-dependent constraints (abscissae) versus runtime without these additional constraints (ordinate). *Unsolved* instances are compared on the **right:** Gap of established solution-bound pair with the objective-dependent constraints (abscissae) and without (ordinate). We observe a stronger benefit of the additional constraints for instances that take more than a second to be solved. The established gap in unsolved problems tends to be smaller with the additional constraints.

Table A.2: Detailled results on the DRIVE dataset. All gaps are shown in %.

	Minimal		Equidistant		Nearest		k-Nearest		k-Interleave		Geodesic		Topocut	
	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap
1	-25.221	99.951	-51036.060	0.076	-25.221	99.951	-25.221	99.951	-51034.143	**0.009**	-51027.773	-	-51033.528	-
2	-6730.676	88.220	-57056.888	0.020	-6730.676	88.220	-57059.422	**0.000**	-57059.422	**0.002**	-57059.422	-	-57022.820	-
3	-426.925	98.514	-11566.359	59.671	-426.925	98.515	-10230.143	64.268	-426.925	98.509	-28495.543	-	-28363.606	-
4	-2136.957	95.723	-49915.736	**0.009**	-2136.957	95.722	-49915.444	**0.001**	-49915.408	**0.006**	-49915.736	-	-49917.342	-
5	-52.461	99.883	-44778.050	0.036	-52.461	99.883	-44782.041	**0.001**	-44782.031	**0.008**	-44782.190	-	-44743.784	-
6	-2935.626	93.132	-42202.005	1.168	-2935.626	93.132	-42240.223	0.967	-42233.743	0.957	-42620.005	-	-42517.828	-
7	-9227.616	79.193	-44247.003	**0.010**	-9227.616	79.193	-44246.127	**0.005**	-44246.892	**0.010**	-44247.278	-	-44191.763	-
8	-8203.665	76.574	-8204.111	76.543	-8203.665	76.575	-8204.111	76.492	-8203.665	76.481	-34837.718	-	-34784.324	-
9	-44.521	99.853	-30083.291	0.696	-44.521	99.853	-44.522	99.853	-44.521	99.853	-30215.726	-	-30078.909	-
10	-10928.011	74.891	-43360.988	0.220	-10928.011	74.891	-43375.895	**0.001**	-43375.894	**0.002**	-43375.380	-	-43340.989	-
11	-218.391	99.548	-48162.391	0.087	-218.391	99.548	-48158.325	**0.010**	-48157.739	0.022	-48160.034	-	-48129.812	-
12	-2259.923	94.873	-43960.046	0.178	-2259.923	94.873	-43048.430	2.170	-43991.618	**0.010**	-43991.618	-	-43985.121	-
13	-9378.653	79.817	-46330.246	0.162	-9378.653	79.817	-45934.617	0.879	-46336.008	**0.010**	-46338.359	-	-46250.965	-
14	-6211.964	86.212	-44790.480	0.435	-6211.964	86.212	-6212.368	86.188	-44952.960	0.021	-44956.168	-	-44919.007	-
15	-43.096	99.901	-43411.817	0.449	-43.096	99.901	-42470.562	2.573	-43554.831	0.030	-43558.363	-	-43531.902	-
16	-9116.216	78.385	-42095.265	0.086	-9116.216	78.386	-9116.693	78.343	-42089.816	**0.006**	-42090.386	-	-42092.120	-
17	-9824.399	73.731	-37305.838	0.081	-9824.399	73.732	-37302.845	**0.008**	-37304.001	**0.001**	-37304.466	-	-37303.848	-
18	-25.801	99.947	-48615.828	**0.010**	-44154.734	9.243	-48615.519	**0.002**	-48615.806	**0.001**	-48615.989	-	-48609.983	-
19	-8593.296	84.006	-52849.344	1.491	-8593.296	84.006	-52717.862	1.541	-8593.296	83.952	-53498.420	-	-53376.441	-
20	-8468.807	79.247	-39938.908	2.005	-8468.807	79.248	-8469.268	79.189	-40146.413	1.347	-40677.896	-	-40648.447	-

Table A.3: Detailled results on the OPF dataset. All gaps are shown in %. Topocut has not been available for 3D data.

	Minimal		Equidistant		Nearest		k-Nearest		k-Interleave		Geodesic	
	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap	Objective	Gap
1	-28195.261	**0.001**	-28195.261	**0.003**	-28198.749	**0.003**	-28194.760	**0.003**	-28195.261	**0.003**	-28195.261	-
3	-8381.838	28.451	-11709.022	**0.009**	-8382.906	28.456	-8382.906	28.443	-8381.838	28.431	-11709.022	-
4	-28337.583	0.014	-28337.583	0.064	-28341.613	0.035	-28337.583	**0.000**	-28337.583	**0.010**	-28337.583	-
5	-3996.556	40.172	-6556.257	1.828	-3997.058	-40.183	-4085.549	38.823	-6671.876	**0.008**	-6671.876	-
6	-12892.958	0.674	-12894.707	0.673	-12894.707	0.695	-12908.342	**0.008**	-12906.592	0.279	-12906.592	-
7	-11547.678	0.355	-11549.190	0.652	-11549.190	0.438	-11549.190	**0.009**	-11547.678	**0.009**	-11547.678	-
8	-12387.719	6.868	-12399.757	10.182	-12389.380	7.019	-12399.757	2.734	-12400.033	2.781	-12400.033	-
9	-16675.030	6.032	-17367.460	2.014	-16677.321	6.110	-17032.020	3.026	-17365.075	1.335	-17552.617	-

B

Supplementary Material to Efficient Algorithms for Moral Lineage Tracing

This chapter has been published as **supplementary material to** [105].

M. Rempfler, J.-H. Lange, F. Jug, C. Blasse, E. W. Myers, B. H. Menze, and B. Andres. "Efficient Algorithms for Moral Lineage Tracing." In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 4705–4714. DOI: 10.1109/ICCV.2017.503

B.1 Transformation Costs for GLA

Here, we detail on the calculation of the change of objective for the transformations applied in GLA (c.f. Section 5.3.1). We start with setParent. Setting a as parent of b will change the objective by

$$\Delta_{ab}^{\text{set}} = c_{ab} - c_b^+ - \mathbb{1}\left(|\text{children(a)}| = 0\right) c_a^- \ , \tag{B.1}$$

where $c_{ab} = -\sum_{e \in E_{ab}} c_e$, $c_a^+ = \sum_{v \in V_a} c_v^+$ and $\mathbb{1}\left(\ldots\right)$ is the indicator function. It accounts for the activated arc ab, the fact that b no longer marks the birth of a new cell and, if a did not have a child previously, it takes the vanishing termination cost into account. A similar reasoning applies to changeParent. When we change the parent of b from a' to a, we get the

following transformation cost:

$$\begin{aligned}\Delta^{\text{change}}_{a'b\to ab} =& c_{ab} - \mathbb{1}\left(|\text{children(a)}| = 0\right) c^-_a \\ & - c_{a'b} + \mathbb{1}\left(|\text{children(a}')| = 1\right) c^-_{a'} \ ,\end{aligned} \tag{B.2}$$

where we have to consider the possibility that a' could form a terminus after the transform. Finally, for a merge of two components a and b of the same frame, we calculate:

$$\begin{aligned}\Delta^{\text{merge}}_{ab} = \ & c_{ab} - \Delta^{\text{birth}}_{ab} - \Delta^{\text{term}}_{ab} \\ & + \sum_{ad\in\mathcal{A}: d\in\text{children(b)}} c_{ad} \quad + \sum_{bd\in\mathcal{A}: d\in\text{children(a)}} c_{bd} \ ,\end{aligned} \tag{B.3}$$

where the last two sums account for arcs to active children, which will be contracted into active arcs with the merge, and therefore change their state and affect the objective. Birth $\Delta^{\text{birth}}_{ab}$ and termination costs $\Delta^{\text{term}}_{ab}$ depend on the current parents and children, that is:

$$\Delta^{\text{term}}_{ab} = \begin{cases} c^-_a & \text{if hasChild(b)} \wedge \neg\text{hasChild(a)} \ , \\ c^-_b & \text{if hasChild(a)} \wedge \neg\text{hasChild(b)} \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{B.4}$$

and

$$\Delta^{\text{birth}}_{ab} = \begin{cases} c^+_a + c_{pa} & \text{if hasParent(b)} \wedge \neg\text{hasParent(a)} \ , \\ c^+_b + c_{pb} & \text{if hasParent(a)} \wedge \neg\text{hasParent(b)} \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{B.5}$$

with pa and pb being the arc from the parent of b or a, respectively. Note that the merge is not feasible if a and b have distinct parents.

B.2 Minimum Cost Branching Coefficients

We derive the weights for the MCBP used in our KLB heuristic (Section 5.3.2). Given a fixed intra-frame partitioning and the corresponding $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, we note that all edges E_{ab} of an arc from component a to component b must have the same state (otherwise, space-time constraints would be violated). We can thus represent them with a set of binary arc

indicator variables y_{ab} satisfying $\forall e \in E_{ab} : 1 - y_{ab} = x_e$. Similiarly, birth and termination indicator variables x^+ and x^- can be grouped with respect to their component, i.e. $\forall v \in V_a : y_a^+ = x_v^+$ (and analogous for y^- and x^-), since all nodes v within a cell must have the same state. Substituting these branching variables into (5.4), leads to:

$$
\begin{aligned}
&\sum_{e\in E} c_e x_e + \sum_{v\in V} c_v^+ x_v^+ + \sum_{v\in V} c_v^- x_v^- \\
=&\sum_{e\in \bigcup_{t\in\mathcal{T}} E_t} c_e x_e + \sum_{ab\in\mathcal{A}} \sum_{e\in E_{ab}} c_e(1-y_{ab}) \\
&+ \sum_{a\in\mathcal{V}} y_a^+ \underbrace{\sum_{v\in V_a} c_v^+}_{c_a^+} + \sum_{a\in\mathcal{V}} y_a^- \underbrace{\sum_{v\in V_a} c_v^-}_{c_a^-} \\
=&\sum_{e\in \bigcup_{t\in\mathcal{T}} E_t} c_e x_e + \sum_{e\in \bigcup_{t\in\mathcal{T}} E_{t,t+1}} c_e + \sum_{ab\in\mathcal{A}} y_{ab} \underbrace{\left(-\sum_{e\in E_{ab}} c_e\right)}_{c_{ab}} \\
&+ \sum_{a\in\mathcal{V}} y_a^+ c_a^+ + \sum_{a\in\mathcal{V}} y_a^- c_a^- \\
=&\sum_{e\in \bigcup_{t\in\mathcal{T}} E_t} c_e x_e \quad + \sum_{e\in \bigcup_{t\in\mathcal{T}} E_{t,t+1}} c_e \\
&+ \sum_{ab\in\mathcal{A}} c_{ab} y_{ab} + \sum_{a\in\mathcal{V}} y_a^+ c_a^+ + \sum_{a\in\mathcal{V}} y_a^- c_a^- \quad ,
\end{aligned}
$$

where the first sum only depends on the fixed intra-frame partitioning, the second term is constant and the remaining three terms correspond to the objective of the MCBP, where we identify the coefficients c_{ab}, c_a^+ and c_a^-. Whenever the arcs selected by y form a branching (which at most bifurcates), then the corresponding x satisfy morality (and bifurcation) constraints.

B.3 Proofs for Section 5.4

Proof of Lemma 5.4.1. We first show that any $x \in \{0,1\}^E$ satisfying all of (5.1) – (5.3) also satisfies (5.13) and (5.14) by contraposition. First, assume $x \in \{0,1\}^E$ violates an inequality of (5.13) 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 (5.1). If $\{v, w\} \in E_{t,t+1}$, then the inequality is included in (5.2). 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 (5.3) 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 (5.1) corresponding to $\{v, w\} \cup P'$ that is violated by x.

Next, suppose $x \in \{0, 1\}^E$ violates an inequality of (5.14) 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 (5.3) 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 (5.13) and (5.14), then it also satisfies (5.1) – (5.3). 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 (5.1) – (5.2) is implied by a combination of inequalities from (5.13). This is a standard argument for multicut polytopes, cf., for instance, [154]. 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 (5.3) are implied by a combination of inequalities from (5.13) and (5.14).

Proof of Lemma 5.4.2. 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 (5.15) 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 (5.6) also holds. Conversely, suppose (5.15) is violated. Then there exists some $t \in \mathcal{T}$ and $v \in V_{t+1}$, $S \in V_t v\text{-cuts}(G_t^+)$ 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 (5.6) 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 (5.2) where P_{uv} is the subpath of P from u to v. This is a contradiction to $x \in X'_G$.

B.4 Additional Results

We report additional, detailled results in terms of runtime, bounds, objective for feasible solutions, and derived gaps obtained on the two additional instances *Flywing-wide I* and *II* in Table B.1. In Fig. B.1, we present a more detailled analysis of the effect of locality parameter d_{MCBP} of our KLB heuristic.

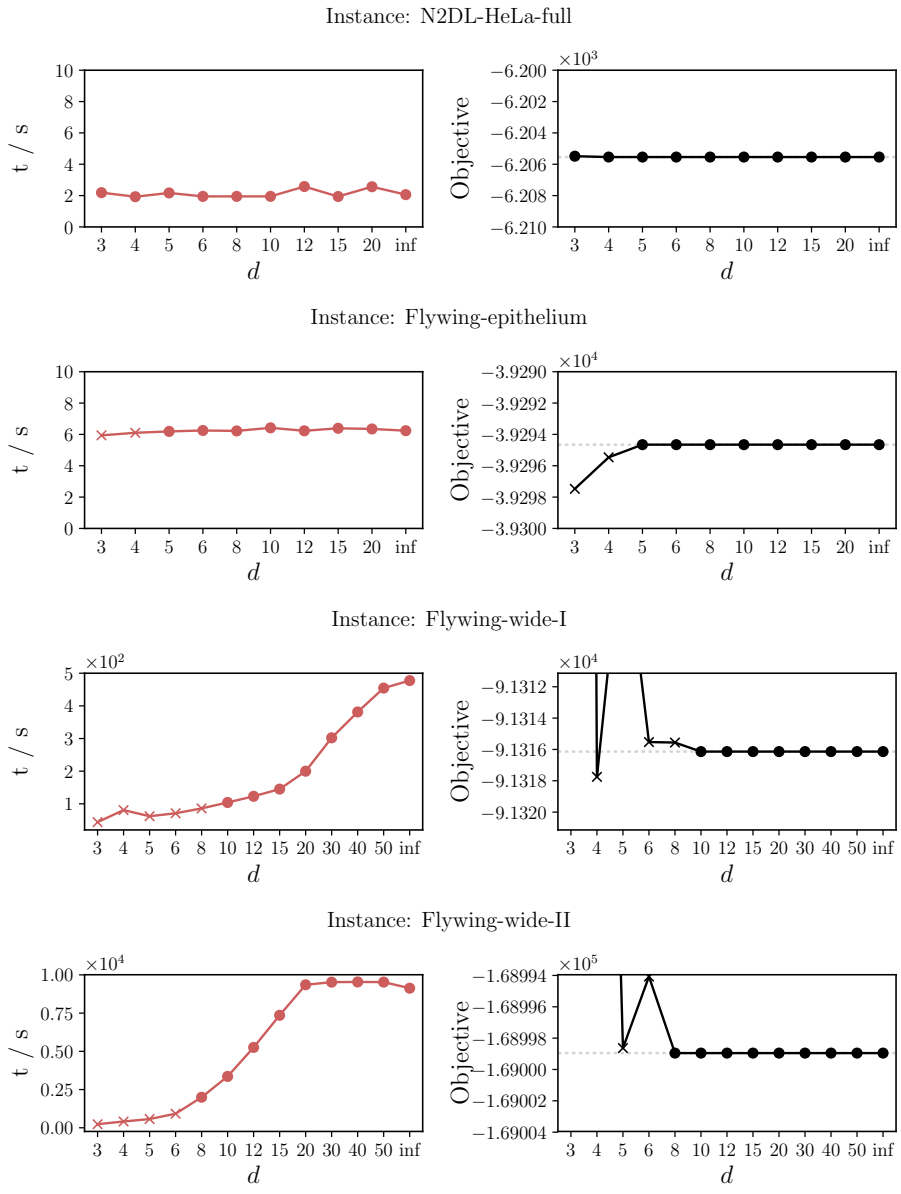


Figure B.1: Comparison of varying d_{MCBP} within KLB in terms of runtime (**left**) and obtained objective (**right**). Parametrizations that were found to (sometimes) misjudge the change of objective due to a too restricted locality are marked with ×, while the others are depicted as •. For the latter parametrizations, we observe that all obtain the same objective value on all instances. However, their runtime varies considerably for the larger two Flywing-wide instances.

Table B.1: Detailed quantitative comparison of algorithms for the MLT on the two additional instances *Flywing-wide I* and *II*. BestGap is calculated using the tightest bound of any algorithm.

Method	Time / s	objBest	objBound	BestGap
		Flywing-wide I		
GLA	0.72	-89895.00		0.0293
KLB-d=10	104.09	-91316.14		0.0133
KLB-d=inf	477.50	-91316.14		0.0133
ILP (ours)	10000.80	-91774.40	-92528.30	0.0082
		Flywing-wide II		
GLA	3.43	-167029.00		0.0214
KLB-d=10	3359.34	-168998.95		0.0095
KLB-d=inf	9129.41	-168998.95		0.0095
ILP (ours)	10245.80	-168862.00	-170606.00	0.0103

C

List of Publications

The following publications were written *during this thesis*.

Journal Articles

- **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." In: *Medical Image Analysis* 25.1 (2015), pp. 86–94. DOI: 10.1016/j.media.2015.03.008.

- **M. Rempfler**, V. Stierle, K. Ditzel, S. Kumar, P. Paulitschke, B. Andres, and B. H. Menze. "Tracing Cell Lineages in Videos of Lens-free Microscopy." In: *Submitted to Medical Image Analysis (invite for MICCAI 2017 Special Issue)* (2018).
- P. F. Christ, F. Ettlinger, F. Grün, M. E. A. Elshaer, J. Lipková, S. Schlecht, F. Ahmaddy, S. Tatavarty, M. Bickel, P. Bilic, **M. Rempfler**, F. Hofmann, M. D'Anastasi, S. Ahmadi, G. Kaissis, J. Holch, W. H. Sommer, R. Braren, V. Heinemann, and B. H. Menze. "Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks." In: *Submitted to PLOS One* (2018).

Peer-reviewed Conference Proceedings

- **M. Rempfler**, J.-H. Lange, F. Jug, C. Blasse, E. W. Myers, B. H. Menze, and B. Andres. "Efficient Algorithms for Moral Lineage Trac-

ing." In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 4705–4714. DOI: 10.1109/ICCV.2017.503.

- **M. Rempfler**, S. Kumar, V. Stierle, P. Paulitschke, B. Andres, and B. H. Menze. "Cell Lineage Tracing in Lens-Free Microscopy Videos." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part II*. Ed. by M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne. Cham: Springer International Publishing, 2017, pp. 3–11. DOI: 10.1007/978-3-319-66185-8_1.
- **M. Rempfler**, B. Andres, and B. H. Menze. "The Minimum Cost Connected Subgraph Problem in Medical Image Analysis." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III*. Ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Cham: Springer International Publishing, 2016, pp. 397–405. DOI: 10.1007/978-3-319-46726-9_46.
- P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, **M. Rempfler**, M. Armbruster, F. Hofmann, M. D'Anastasi, W. H. Sommer, S.-A. Ahmadi, and B. H. Menze. "Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II*. Ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Cham: Springer International Publishing, 2016, pp. 415–423. DOI: 10.1007/978-3-319-46723-8_48.
- E. Alberts, **M. Rempfler**, G. Alber, T. Huber, J. Kirschke, C. Zimmer, and B. H. Menze. "Uncertainty quantification in brain tumor segmentation using CRFs and random perturbation models." In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. Apr. 2016, pp. 428–431. DOI: 10.1109/ISBI.2016.7493299.
- U. Sivalingam, M. Wels, **M. Rempfler**, S. Grosskopf, M. Suehling, and B. H. Menze. "Inner and outer coronary vessel wall segmentation

from CCTA using an active contour model with machine learning-based 3D voxel context-aware image force." In: *Proc. SPIE Medical Imaging*. Vol. 9785. 2016, DOI: 10.1117/12.2216200.

Peer-reviewed Workshop Proceedings

- **M. Rempfler**, B. Andres, and B. H. Menze. "Uncertainty Estimation in Vascular Networks." In: *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*. Ed. by M. J. Cardoso, T. Arbel, E. Ferrante, X. Pennec, A. V. Dalca, S. Parisot, S. Joshi, N. K. Batmanghelich, A. Sotiras, M. Nielsen, M. R. Sabuncu, T. Fletcher, L. Shen, S. Durrleman, and S. Sommer. Cham: Springer International Publishing, 2017, pp. 42–52. DOI: 10.1007/978-3-319-67675-3_5.
- V. Zografos, A. Valentinitsch, **M. Rempfler**, F. Tombari, and B. Menze. "Hierarchical Multi-Organ Segmentation Without Registration in 3D Abdominal CT Images." In: *Medical Computer Vision: Algorithms for Big Data: International Workshop, MCV 2015, Held in Conjunction with MICCAI 2015, Munich, Germany, October 9, 2015, Revised Selected Papers*. Ed. by B. Menze, G. Langs, A. Montillo, M. Kelm, H. Müller, S. Zhang, W. Cai, and D. Metaxas. Cham: Springer International Publishing, 2016, pp. 37–46. DOI: 10.1007/978-3-319-42016-5_4.
- **M. Rempfler**, M. Schneider, G. D. Ielacqua, X. Xiao, S. R. Stock, J. Klohs, G. Székely, B. Andres, and B. H. Menze. "Rekonstruktion zerebraler Gefässnetzwerke aus in-vivo μMRA mittels physiologischem Vorwissen zur lokalen Gefässgeometrie." In: *Bildverarbeitung für die Medizin 2015: Algorithmen - Systeme - Anwendungen. Proceedings des Workshops vom 15. bis 17. März 2015 in Lübeck*. Ed. by H. Handels, T. M. Deserno, H.-P. Meinzer, and T. Tolxdorff. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 161–166. DOI: 10.1007/978-3-662-46224-9_29.

Bibliography

[1] H. Gray. *Anatomy of the human body*. Lea & Febiger, 1918.

[2] M. Rubinov and O. Sporns. "Complex network measures of brain connectivity: Uses and interpretations." In: *NeuroImage* 52.3 (2010). Computational Models of the Brain, pp. 1059–1069.

[3] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. "The large-scale organization of metabolic networks." In: *Nature* 407.6804 (2000), pp. 651–654.

[4] J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles. "Metabolic network structure determines key aspects of functionality and regulation." In: *Nature* 420.6912 (2002), pp. 190–193.

[5] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. "Using Bayesian networks to analyze expression data." In: *Journal of computational biology* 7.3-4 (2000), pp. 601–620.

[6] N. Guelzim, S. Bottani, P. Bourgine, and F. Képès. "Topological and causal structure of the yeast transcriptional regulatory network." In: *Nature genetics* 31.1 (2002), pp. 60–63.

[7] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, et al. "A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae." In: *Nature* 403.6770 (2000), pp. 623–627.

[8] S. Maslov and K. Sneppen. "Specificity and stability in topology of protein networks." In: *Science* 296.5569 (2002), pp. 910–913.

[9] R. Diestel. *Graph theory*. 5th ed. Springer-Verlag Heidelberg, 2000.

[10] B. Bollobás. *Modern graph theory*. Vol. 184. Springer Science & Business Media, 2013.

[11] L. Euler. "Solutio problematis ad geometriam situs pertinensis." In: *Comm. Acad. Sci. Imper. Petropol.* 8 (1736), pp. 128–140.

[12] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press, 1994.

[13] M. G. Bell and Y. Iida. *Transportation network analysis*. Wiley Online Library, 1997.

[14] The Internet Engineering Task Force (IETF). *Request for Comments (RFC) pages*. 2018. URL: https://www.ietf.org/rfc.html (visited on 01/01/2018).

[15] M. Faloutsos, P. Faloutsos, and C. Faloutsos. "On power-law relationships of the internet topology." In: *ACM SIGCOMM computer communication review*. Vol. 29. 4. ACM. 1999, pp. 251–262.

[16] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. "Network Motifs: Simple Building Blocks of Complex Networks." In: *Science* 298.5594 (2002), pp. 824–827.

[17] M. E. J. Newman. "The Structure and Function of Complex Networks." In: *SIAM Review* 45.2 (2003), pp. 167–256.

[18] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. "Complex networks: Structure and dynamics." In: *Physics reports* 424.4 (2006), pp. 175–308.

[19] U. Alon. "Network motifs: theory and experimental approaches." In: *Nature Reviews Genetics* 8.6 (2007), pp. 450–461.

[20] 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." In: *Medical Image Analysis* 25.1 (2015), pp. 86–94.

[21] M. Rempfler, S. Kumar, V. Stierle, P. Paulitschke, B. Andres, and B. H. Menze. "Cell Lineage Tracing in Lens-Free Microscopy Videos." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part II*. Ed. by M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne. Cham: Springer International Publishing, 2017, pp. 3–11.

[22] J.-P. Vert and Y. Yamanishi. "Supervised Graph Inference." In: *Advances in Neural Information Processing Systems 17*. Ed. by L. K. Saul, Y. Weiss, and L. Bottou. MIT Press, 2005, pp. 1433–1440.

[23] Y. Yamanishi. "Supervised Bipartite Graph Inference." In: *Advances in Neural Information Processing Systems 21*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Curran Associates, Inc., 2009, pp. 1841–1848.

[24] M. Fiori. "Graph inference and graph matching problems: theory and algorithms." Dissertation. UR. FI-IIE, 2015.

[25] C. E. Priebe, D. L. Sussman, M. Tang, and J. T. Vogelstein. "Statistical inference on errorfully observed graphs." In: *Journal of Computational and Graphical Statistics* 24.4 (2015), pp. 930–953.

[26] E. J. Candès and B. Recht. "Exact matrix completion via convex optimization." In: *Foundations of Computational mathematics* 9.6 (2009), p. 717.

[27] M. Fiori, P. Musé, and G. Sapiro. "Topology Constraints in Graphical Models." In: *Advances in Neural Information Processing Systems 25*. 2012, pp. 791–799.

[28] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst. "Matrix completion on graphs." In: *arXiv preprint arXiv:1408.1717* (2014).

[29] Y. Koren, R. Bell, and C. Volinsky. "Matrix Factorization Techniques for Recommender Systems." In: *Computer* 42.8 (Aug. 2009), pp. 30–37.

[30] E. Türetken, F. Benmansour, B. Andres, P. Głowacki, H. Pfister, and P. Fua. "Reconstructing Curvilinear Networks Using Path Classifiers and Integer Programming." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.12 (Dec. 2016), pp. 2515–2530.

[31] D. Robben, E. Türetken, S. Sunaert, V. Thijs, G. Wilms, P. Fua, F. Maes, and P. Suetens. "Simultaneous segmentation and anatomical labeling of the cerebral vasculature." In: *Medical Image Analysis* 32 (2016), pp. 201–215.

[32] C. Payer, M. Pienn, Z. Bálint, A. Shekhovtsov, E. Talakic, E. Nagy, A. Olschewski, H. Olschewski, and M. Urschler. "Automated integer programming based separation of arteries and veins from thoracic {CT} images." In: *Medical Image Analysis* 34 (2016). Special Issue on the 2015 Conference on Medical Image Computing and Computer Assisted Intervention, pp. 109–122.

[33] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. "Multiple object tracking using k-shortest paths optimization." In: *IEEE transactions on pattern analysis and machine intelligence* 33.9 (2011), pp. 1806–1819.

[34] S. Tang, B. Andres, M. Andriluka, and B. Schiele. "Subgraph decomposition for multi-target tracking." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5033–5041.

[35] X. Wang, E. Türetken, F. Fleuret, and P. Fua. "Tracking Interacting Objects Optimally Using Integer Programming." In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 17–32.

[36] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. "ArtTrack: Articulated Multi-person Tracking in the Wild." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[37] S. Tang, M. Andriluka, B. Andres, and B. Schiele. "Multiple People Tracking by Lifted Multicut and Person Re-Identification." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[38] J. Qian, V. Saligrama, and Y. Chen. "Connected Sub-graph Detection." In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. Ed. by S. Kaski and J. Corander. Vol. 33. Proceedings of Machine Learning Research. Reykjavik, Iceland: PMLR, 22–25 Apr 2014, pp. 796–804.

[39] J. Qian and V. Saligrama. "Efficient Minimax Signal Detection on Graphs." In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2708–2716.

[40] D. Wassermann, D. Mazauric, G. Gallardo-Diez, and R. Deriche. "Extracting the Core Structural Connectivity Network: Guaranteeing Network Connectedness Through a Graph-Theoretical Approach." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part I*. Ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Cham: Springer International Publishing, 2016, pp. 89–96.

[41] J. Klohs, C. Baltes, F. Princz-Kranz, D. Ratering, R. M. Nitsch, I. Knuesel, and M. Rudin. "Contrast-Enhanced Magnetic Resonance Microangiography Reveals Remodeling of the Cerebral Microvasculature in Transgenic ArcAβ Mice." In: *Journal of Neuroscience* 32.5 (2012), pp. 1705–1713.

[42] J. M. Hunter, J. Kwan, M. Malek-Ahmadi, C. L. Maarouf, T. A. Kokjohn, C. Belden, M. N. Sabbagh, T. G. Beach, and A. E. Roher. "Morphological and pathological evolution of the brain microcirculation in aging and Alzheimer's disease." In: *PloS one* 7.5 (2012), e36893.

[43] L. D. Cohen and R. Kimmel. "Global Minimum for Active Contour Models: A Minimal Path Approach." In: *International Journal of Computer Vision* 24.1 (Aug. 1997), pp. 57–78.

[44] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. "Multiscale vessel enhancement filtering." In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI'98: First International Conference Cambridge, MA, USA, October 11–13, 1998 Proceedings*. Ed. by W. M. Wells, A. Colchester, and S. Delp. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 130–137.

[45] S. R. Aylward and E. Bullitt. "Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction." In: *IEEE Transactions on Medical Imaging* 21.2 (Feb. 2002), pp. 61–75.

[46] C. Kirbas and F. Quek. "A Review of Vessel Extraction Techniques and Algorithms." In: *ACM Comput. Surv.* 36.2 (2004), pp. 81–121.

[47] M. A. Gülsün and H. Tek. "Robust Vessel Tree Modeling." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008: 11th International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part I*. Ed. by D. Metaxas, L. Axel, G. Fichtinger, and G. Székely. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 602–611.

[48] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. "A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes." In: *Medical Image Analysis* 13.6 (2009), pp. 819–845.

[49] F. Benmansour and L. D. Cohen. "Tubular Structure Segmentation Based on Minimal Path Method and Anisotropic Enhancement." In: *International Journal of Computer Vision* 92.2 (Apr. 2011), pp. 192–210.

[50] M. Schneider, S. Hirsch, B. Weber, G. Székely, and B. H. Menze. "Joint 3-D vessel segmentation and centerline extraction using oblique Hough forests with steerable filters." In: *Medical Image Analysis* 19.1 (2015), pp. 220–249.

[51] U. Sivalingam, M. Wels, M. Rempfler, S. Grosskopf, M. Suehling, and B. H. Menze. "Inner and outer coronary vessel wall segmentation from CCTA using an active contour model with machine learning-based 3D voxel context-aware image force." In: *Proc. SPIE Medical Imaging*. Vol. 9785. 2016,

[52] T. Zhao, J. Xie, F. Amat, N. Clack, P. Ahammad, H. Peng, F. Long, and E. Myers. "Automated reconstruction of neuronal morphology based on local geometrical and global structural models." In: *Neuroinformatics* 9.2-3 (2011), pp. 247–261.

[53] K. M. Brown, G. Barrionuevo, A. J. Canty, V. De Paola, J. A. Hirsch, G. S. X. E. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G. A. Ascoli. "The DIADEM Data Sets: Representative Light Microscopy Images of Neuronal Morphology to Advance Automation of Digital Reconstructions." In: *Neuroinformatics* 9.2 (2011), pp. 143–157.

[54] L. Acciai, P. Soda, and G. Iannello. "Automated Neuron Tracing Methods: An Updated Account." In: *Neuroinformatics* 14.4 (Oct. 2016), pp. 353–367.

[55] P. Dollár and C. L. Zitnick. "Fast edge detection using structured forests." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.8 (2015), pp. 1558–1570.

[56] Y. Ganin and V. Lempitsky. "N^4-Fields: Neural Network Nearest Neighbor Fields for Image Transforms." In: *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II*. Ed. by D. Cremers, I. Reid, H. Saito, and M.-H. Yang. Cham: Springer International Publishing, 2015, pp. 536–551.

[57] A. Sironi, E. Türetken, V. Lepetit, and P. Fua. "Multiscale centerline detection." In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 1 (2015), pp. 1–14.

[58] S. Xie and Z. Tu. "Holistically-Nested Edge Detection." In: *The IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 1395–1403.

[59] Y. Jiang, Z. Zhuang, A. J. Sinusas, and X. Papademetris. "Vascular tree reconstruction by minimizing a physiological functional cost." In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. June 2010, pp. 178–185.

[60] Y. Jiang, Z. W. Zhuang, A. J. Sinusas, L. H. Staib, and X. Papademetris. "Vessel Connectivity Using Murray's Hypothesis." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011: 14th International Conference, Toronto, Canada, September 18-22, 2011, Proceedings, Part III*. Ed. by G. Fichtinger, A. Martel, and T. Peters. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 528–536.

[61] A. Mosinska, J. Tarnawski, and P. Fua. "Active Learning and Proofreading for Delineation of Curvilinear Structures." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, pp. 165–173.

[62] B. Korte and J. Vygen. *Combinatorial optimization: theory and algorithms*. 5th ed. Springer Science & Business Media, 2011.

[63] S. Vicente, V. Kolmogorov, and C. Rother. "Graph cut based image segmentation with connectivity priors." In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. June 2008, pp. 1–8.

[64] S. Nowozin and C. H. Lampert. "Global connectivity potentials for random field models." In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 818–825.

[65] C. Chen, D. Freedman, and C. H. Lampert. "Enforcing topological constraints in random field image segmentation." In: *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2011, pp. 2089–2096.

[66] J. Stühmer, P. Schröder, and D. Cremers. "Tree Shape Priors with Connectivity Constraints Using Convex Relaxation on General Graphs." In: *2013 IEEE International Conference on Computer Vision*. Dec. 2013, pp. 2336–2343.

[67] J. Stühmer and D. Cremers. "A Fast Projection Method for Connectivity Constraints in Image Segmentation." In: *Energy Minimization Methods in Computer Vision and Pattern Recognition: 10th International Conference, EMMCVPR 2015, Hong Kong, China, January 13-16, 2015. Proceedings*. Ed. by X.-C. Tai, E. Bae, T. F. Chan, and M. Lysaker. Cham: Springer International Publishing, 2015, pp. 183–196.

[68] J. Stühmer. "A Convex Optimization Framework for Connectivity Constraints in Image Segmentation and 3D Reconstruction." Dissertation. München: Technische Universität München, 2016.

[69] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. "Ridge-based vessel segmentation in color images of the retina." In: *IEEE Transactions on Medical Imaging* 23.4 (Apr. 2004), pp. 501–509.

[70] 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." In: *Nature methods* 7.8 (2010), pp. 637–642.

[71] P. J. Keller, A. D. Schmidt, J. Wittbrodt, and E. H. Stelzer. "Reconstruction of Zebrafish Early Embryonic Development by Scanned Light Sheet Microscopy." In: *Science* 322.5904 (2008), pp. 1065–1069.

[72] C. Guillot and T. Lecuit. "Mechanics of epithelial tissue homeostasis and morphogenesis." In: *Science* 340.6137 (2013), pp. 1185–1189.

[73] 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." In: *Proceedings of the National Academy of Sciences* 109.34 (2012), pp. 13515–13520.

[74] 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." In: *Nature methods* (2014).

[75] F. Amat, E. W. Myers, and P. J. Keller. "Fast and robust optical flow for time-lapse microscopy using super-voxels." In: *Bioinformatics* 29.3 (2013), pp. 373–380.

[76] 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." In: *Nature Methods* 11.3 (2014), pp. 281–289.

[77] 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." In: *Medical image analysis* 12.5 (2008), pp. 546–566.

[78] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. Balak, et al. "A benchmark for comparison of cell tracking algorithms." In: *Bioinformatics* 30.11 (2014), pp. 1609–1617.

[79] E. Meijering, O. Dzyubachyk, and I. Smal. "Methods for cell and particle tracking." In: *Methods in Enzymology* 504.9 (2012), pp. 183–200.

[80] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer Science & Business Media, 2002.

[81] 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*. Springer, 2014, pp. 25–36.

[82] 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: *Computer Vision – ECCV 2012*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 144–157.

[83] D. Padfield, J. Rittscher, and B. Roysam. "Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis." In: *Medical image analysis* 15.4 (2011), pp. 650–668.

[84] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht. "Conservation tracking." In: *2013 IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 2928–2935.

[85] 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." In: *Bioinformatics* 31 (2015), pp. 948–956.

[86] F. Jug, E. Levinkov, C. Blasse, E. W. Myers, and B. Andres. "Moral Lineage Tracing." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[87] B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. "Probabilistic Image Segmentation with Closedness Constraints." In: *2011 IEEE International Conference on Computer Vision (ICCV)*. 2011.

[88] B. Andres, T. Kroeger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Koethe, and F. A. Hamprecht. "Globally Optimal Closed-Surface Segmentation for Connectomics." In: *Computer Vision – ECCV 2012*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 778–791.

[89] 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.

[90] S. Bagon and M. Galun. "Large Scale Correlation Clustering Optimization." In: *CoRR* abs/1112.2903 (2011).

[91] T. Beier, B. Andres, U. Köthe, and F. A. Hamprecht. "An Efficient Fusion Move Algorithm for the Minimum Cost Lifted Multicut Problem." In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Cham: Springer International Publishing, 2016, pp. 715–730.

[92] T. Beier, F. A. Hamprecht, and J. H. Kappes. "Fusion moves for correlation clustering." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3507–3516.

[93] T. Beier, T. Kroeger, J. H. Kappes, U. Köthe, and F. A. Hamprecht. "Cut, Glue, & Cut: A Fast, Approximate Solver for Multicut Partitioning." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 73–80.

[94] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnörr. "Higher-order segmentation via multicuts." In: *Computer Vision and Image Understanding* 143 (2016), pp. 104–119.

[95] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. "Efficient Decomposition of Image and Mesh Graphs by Lifted Multicuts." In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1751–1759.

[96] S. Kim, C. D. Yoo, S. Nowozin, and P. Kohli. "Image Segmentation UsingHigher-Order Correlation Clustering." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.9 (2014), pp. 1761–1774.

[97] J. Yarkony, A. Ihler, and C. C. Fowlkes. "Fast Planar Correlation Clustering for Image Segmentation." In: *Computer Vision – ECCV 2012*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 568–581.

[98] J. E. Yarkony and C. Fowlkes. "Planar Ultrametrics for Image Segmentation." In: *NIPS*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 64–72.

[99] O. Mudanyali, D. Tseng, C. Oh, S. O. Isikman, I. Sencan, W. Bishara, C. Oztoprak, S. Seo, B. Khademhosseini, and A. Ozcan. "Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications." In: *Lab on a Chip* 10.11 (2010), pp. 1417–1428.

[100] 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." In: *Nature methods* 9.9 (2012), pp. 889–895.

[101] S. V. Kesavan, F. Momey, O. Cioni, B. David-Watine, N. Dubrulle, S. Shorte, E. Sulpice, D. Freida, B. Chalmond, J. Dinten, et al. "High-throughput monitoring of major cell functions by means of lensfree video microscopy." In: *Scientific reports* 4 (2014), p. 5942.

[102] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[103] P. Kainz, M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit. "You Should Use Regression to Detect Cells." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Cham: Springer International Publishing, 2015, pp. 276–283.

[104] M. Rempfler, B. Andres, and B. H. Menze. "The Minimum Cost Connected Subgraph Problem in Medical Image Analysis." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III*. Ed. by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells. Cham: Springer International Publishing, 2016, pp. 397–405.

[105] M. Rempfler, J.-H. Lange, F. Jug, C. Blasse, E. W. Myers, B. H. Menze, and B. Andres. "Efficient Algorithms for Moral Lineage Tracing." In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 4705–4714.

[106] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[107] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[108] C. M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag New York, 2006.

[109] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009.

[110] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[111] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[112] R. M. Karp. "Reducibility among combinatorial problems." In: *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[113] N. Karmarkar. "A new polynomial-time algorithm for linear programming." In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM. 1984, pp. 302–311.

[114] G. B. Dantzig. "Maximization of a Linear Function of Variables Subject to Linear Inequalities, in Activity Analysis of Production and Allocation." In: (1951), pp. 339–347.

[115] R. E. Gomory. "Outline of an algorithm for integer solutions to linear programs." In: *Bull. Amer. Math. Soc.* 64.5 (Sept. 1958), pp. 275–278.

[116] A. H. Land and A. G. Doig. "An automatic method of solving discrete programming problems." In: *Econometrica: Journal of the Econometric Society* (1960), pp. 497–520.

[117] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual*. 2015. URL: http://www.gurobi.com.

[118] IBM ILOG CPLEX optimizer. 2014. URL: http://www.ibm.com/software/integration/optimization/cplex-optimizer/.

[119] G. Dantzig, R. Fulkerson, and S. Johnson. "Solution of a large-scale traveling-salesman problem." In: *Journal of the operations research society of America* 2.4 (1954), pp. 393–410.

[120] S. Chopra and M. R. Rao. "The partition problem." In: *Mathematical Programming* 59.1 (1993), pp. 87–115.

[121] C. Cortes and V. Vapnik. "Support-vector networks." In: *Machine learning* 20.3 (1995), pp. 273–297.

[122] L. Breiman. "Random forests." In: *Machine learning* 45.1 (2001), pp. 5–32.

[123] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[124] G. Cybenko. "Approximation by superpositions of a sigmoidal function." In: *Mathematics of Control, Signals, and Systems (MCSS)* 2.4 (1989), pp. 303–314.

[125] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[126] S. Hochreiter and J. Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[127] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." In: *Journal of Machine Learning Research* 11.Dec (2010), pp. 3371–3408.

[128] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, et al. "The multimodal brain tumor image segmentation benchmark (BRATS)." In: *IEEE transactions on medical imaging* 34.10 (2015), pp. 1993–2024.

[129] *LiTS – Liver Tumor Segmentation Challenge*. 2018. URL: https://competitions.codalab.org/competitions/17094 (visited on 03/01/2018).

[130] O. Maier, B. H. Menze, J. von der Gablentz, L. Häni, M. P. Heinrich, M. Liebrand, S. Winzeck, A. Basit, P. Bentley, L. Chen, et al. "ISLES 2015-A public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI." In: *Medical image analysis* 35 (2017), pp. 250–269.

[131] R. Kohavi et al. “A study of cross-validation and bootstrap for accuracy estimation and model selection.” In: *Ijcai*. Vol. 14. 2. Stanford, CA. 1995, pp. 1137–1145.

[132] S. Nowozin, C. H. Lampert, et al. “Structured learning and prediction in computer vision.” In: *Foundations and Trends® in Computer Graphics and Vision* 6.3–4 (2011), pp. 185–365.

[133] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. “A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems.” In: *International Journal of Computer Vision* 115.2 (2015), pp. 155–184.

[134] M. Rempfler, M. Schneider, G. D. Ielacqua, X. Xiao, S. R. Stock, J. Klohs, G. Székely, B. Andres, and B. H. Menze. “Extracting Vascular Networks under Physiological Constraints via Integer Programming.” In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part II*. Ed. by P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe. Cham: Springer International Publishing, 2014, pp. 505–512.

[135] C. Pudney. “Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images.” In: *Computer Vision and Image Understanding* 72.3 (1998), pp. 404–413.

[136] T. Lee, R. Kashyap, and C. Chu. “Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms.” In: *CVGIP: Graphical Models and Image Processing* 56.6 (1994), pp. 462–478.

[137] L. Lu, J. Bi, S. Yu, Z. Peng, A. Krishnan, and X. S. Zhou. “Hierarchical learning for tubular structure parsing in medical imaging: A study on coronary arteries using 3D CT Angiography.” In: *2009 IEEE 12th International Conference on Computer Vision (ICCV)*. Sept. 2009, pp. 2021–2028.

[138] J. P. Kaufhold, P. S. Tsai, P. Blinder, and D. Kleinfeld. “Vectorization of optically sectioned brain microvasculature: Learning aids completion of vascular graphs by connecting gaps and deleting open-ended segments.” In: *Medical Image Analysis* 16.6 (2012), pp. 1241–1258.

[139] M. Schneider, S. Hirsch, B. Weber, G. Székely, and B. H. Menze. "TGIF: Topological Gap In-Fill for Vascular Networks." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part II*. Ed. by P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe. Cham: Springer International Publishing, 2014, pp. 89–96.

[140] C. D. Murray. "The physiological principle of minimum work I. The vascular system and the cost of blood volume." In: *Proceedings of the National Academy of Sciences* 12.3 (1926), pp. 207–214.

[141] E. Turetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua. "Reconstructing Loopy Curvilinear Structures Using Integer Programming." In: *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2013, pp. 1822–1829.

[142] M. Schneider, S. Hirsch, G. Székely, B. Weber, and B. H. Menze. "Oblique Random Forests for 3-D Vessel Detection Using Steerable Filters and Orthogonal Subspace Filtering." In: *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging: Second International MICCAI Workshop, MCV 2012, Nice, France, October 5, 2012, Revised Selected Papers*. Ed. by B. H. Menze, G. Langs, L. Lu, A. Montillo, Z. Tu, and A. Criminisi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 142–154.

[143] B. H. Menze, B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht. "On Oblique Random Forests." In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II*. Ed. by D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 453–469.

[144] L. Qu and B. Andres. "Estimating Maximally Probable Constrained Relations by Mathematical Programming." In: *CoRR* abs/1408.0838 (2014).

[145] P. Miliotis. "Integer programming approaches to the travelling salesman problem." In: *Mathematical Programming* 10.1 (Dec. 1976), pp. 367–378.

[146] S. Heinzer, T. Krucker, M. Stampanoni, R. Abela, E. P. Meyer, A. Schuler, P. Schneider, and R. Müller. "Hierarchical microimaging for multiscale analysis of large vascular networks." In: *NeuroImage* 32.2 (2006), pp. 626–636.

[147] B. H. Menze, K. van Leemput, D. Lashkari, M.-A. Weber, N. Ayache, and P. Golland. "A Generative Model for Brain Tumor Segmentation in Multi-Modal Images." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010: 13th International Conference, Beijing, China, September 20-24, 2010, Proceedings, Part II*. Ed. by T. Jiang, N. Navab, J. P. W. Pluim, and M. A. Viergever. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 151–159.

[148] T. Riklin-Raviv, K. V. Leemput, B. H. Menze, W. M. Wells, and P. Golland. "Segmentation of image ensembles via latent atlases." In: *Medical Image Analysis* 14.5 (2010). Special Issue on the 12th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2009, pp. 654–665.

[149] 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." In: *Science* 346.6208 (2014), p. 1257998.

[150] R. Tomer, K. Khairy, F. Amat, and P. J. Keller. "Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy." In: *Nature methods* 9.7 (2012), pp. 755–763.

[151] J. Funke, B. Andres, F. A. Hamprecht, A. Cardona, and M. Cook. "Efficient automatic 3D-reconstruction of branching neurons from EM data." In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012, pp. 1004–1011.

[152] E. Türetken, G. González, C. Blum, and P. Fua. "Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors." In: *Neuroinformatics* 9.2-3 (2011), pp. 279–302.

[153] 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: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[154] A. Horňáková, J.-H. Lange, and B. Andres. "Analysis and Optimzation of Graph Decompositions by Lifted Multicuts." In: *ICML*. 2017.

[155] B. W. Kernighan and S. Lin. "An efficient heuristic procedure for partitioning graphs." In: *Bell system technical journal* 49.2 (1970), pp. 291–307.

[156] H. W. Kuhn. "The Hungarian method for the assignment problem." In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[157] J. Munkres. "Algorithms for the assignment and transportation problems." In: *Journal of the society for industrial and applied mathematics* 5.1 (1957), pp. 32–38.

[158] 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." In: *Cell* 142.5 (2010), pp. 773–786.

[159] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241.

[160] A. Khan, S. Gould, and M. Salzmann. "Deep Convolutional Neural Networks for Human Embryonic Cell Counting." In: *Computer Vision – ECCV 2016 Workshops*. Ed. by G. Hua and H. Jégou. Cham: Springer International Publishing, 2016, pp. 339–348.

[161] A. Arbelle, N. Drayman, M. Bray, U. Alon, A. Carpenter, and T. R. Raviv. "Analysis of High-throughput Microscopy Videos: Catching Up with Cell Dynamics." In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Cham: Springer International Publishing, 2015, pp. 218–225.

[162] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser. "Robust single-particle tracking in live-cell time-lapse sequences." In: *Nature methods* 5.8 (2008), pp. 695–702.

[163] G. Flaccavento, V. Lempitsky, I. Pope, P. Barber, A. Zisserman, J. A. Noble, and B. Vojnovic. "Learning to count cells: applications to lens-free imaging of large fields." In: *Microscopic Image Analysis with Applications in Biology* (2011).

[164] K. E. G. Magnusson and J. Jaldén. "A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages." In: *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*. 2012, pp. 382–385.

[165] K. Bernardin and R. Stiefelhagen. "Evaluating multiple object tracking performance: the CLEAR MOT metrics." In: *EURASIP Journal on Image and Video Processing* 2008.1 (2008), pp. 1–10.

[166] A. Sanfeliu and K.-S. Fu. "A distance measure between attributed relational graphs for pattern recognition." In: *IEEE transactions on systems, man, and cybernetics* 3 (1983), pp. 353–362.

[167] T. A. Gillette, K. M. Brown, and G. A. Ascoli. "The DIADEM metric: comparing multiple reconstructions of the same neuron." In: *Neuroinformatics* 9.2-3 (2011), p. 233.

[168] V. Ulman, M. Maška, K. E. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, et al. "An objective comparison of cell-tracking algorithms." In: *Nature methods* 14.12 (2017), p. 1141.

[169] R. Shen, E. Kendinibilir, I. B. Ayed, A. Lodi, A. Tramontani, and G. Reinelt. "An ILP Solver for Multi-label MRFS with Connectivity Constraints." In: *CoRR* abs/1712.06020 (2017).

[170] S. Wolf, L. Schott, U. Kothe, and F. Hamprecht. "Learned Watershed: End-To-End Learning of Seeded Segmentation." In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[171] M. Rempfler, V. Stierle, K. Ditzel, S. Kumar, P. Paulitschke, B. Andres, and B. H. Menze. "Tracing Cell Lineages in Videos of Lens-free Microscopy." In: *Submitted to Medical Image Analysis (invite for MICCAI 2017 Special Issue)* (2018).

[172] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. "Learning by tracking: Siamese CNN for robust target association." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 33–40.

[173] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, et al. "The multimodal brain tumor image segmentation benchmark (BRATS)." In: *IEEE transactions on medical imaging* 34.10 (2015), pp. 1993–2024.

[174] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. "Deep retinal image understanding." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 140–148.

[175] H. Fu, Y. Xu, S. Lin, D. W. K. Wong, and J. Liu. "Deepvessel: Retinal vessel segmentation via deep learning and conditional random field." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 132–139.

[176] J. Funke, F. D. Tschopp, W. Grisaitis, C. Singh, S. Saalfeld, and S. C. Turaga. "A Deep Structured Learning Approach Towards Automating Connectome Reconstruction from 3D Electron Micrographs." In: *arXiv preprint arXiv:1709.02974* (2017).

[177] J. Zung, I. Tartavull, K. Lee, and H. S. Seung. "An Error Detection and Correction Framework for Connectomics." In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 6818–6829.

[178] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. "Geometric Deep Learning: Going beyond Euclidean data." In: *IEEE Signal Processing Magazine* 34.4 (July 2017), pp. 18–42.

[179] M. Henaff, J. Bruna, and Y. LeCun. "Deep convolutional networks on graph-structured data." In: *arXiv preprint arXiv:1506.05163* (2015).

[180] M. Niepert, M. Ahmed, and K. Kutzkov. "Learning convolutional neural networks for graphs." In: *International conference on machine learning*. 2016, pp. 2014–2023.

[181] M. Defferrard, X. Bresson, and P. Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." In: *Advances in Neural Information Processing Systems*. 2016, pp. 3844–3852.

[182] F. Monti, M. M. Bronstein, and X. Bresson. "Geometric Matrix Completion with Recurrent Multi-Graph Neural Networks." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 2017, pp. 3700–3710.

[183] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. "Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 5425–5434.

[184] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. G. Moreno, B. Glocker, and D. Rueckert. "Spectral graph convolutions for population-based disease prediction." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, pp. 177–185.

[185] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert. "Distance metric learning using graph convolutional networks: Application to functional brain networks." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, pp. 469–477.