Fast, Flexible, and Exact Minimum Flow Decompositions via ILP

Fernando H.C. Dias[1], Lucia Williams[2], Brendan Mumey[2], Alexandru I. Tomescu[1]

[1] Department of Computer Science, University of Helsinki, Finland
[2] School of Computing, Montana State University, Bozeman, MT, USA

June 29, 2022

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Introduction

Background and Motivation

UNIVERSITY OF HELSINKI

Flow decomposition (FD), the problem of decomposing a **network flow** into a set of source-to-sink **paths** and associated **weights** that perfectly explain the flow values on the edges, is a classical and well-studied concept in Computer Science.

Motivation

- The main bioinformatics motivation for this paper is *multiassembly* [1], reconstruct multiple genomic sequences from mixed samples using short substrings (called *reads*) generated cheaply and accurately from next-generation sequencing technology;
- One example is the **reconstruction of RNA transcripts** from sequencing reads, which is essential to characterize gene regulation and function, development and diseases such as cancer;
- Another is the **reconstruction of viral quasispecies** which can identify different strains of a virus in a samples.

Methodology

UNIVERSITY OF HELSINKI

Definition (Flow network)

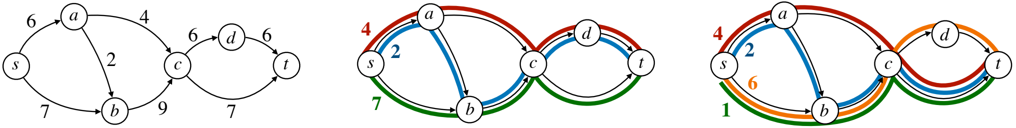
A tuple $G = (V, E, f)$ is said to be a *flow network* if (V, E) is a DAG with unique source s and unique sink t, where for every edge $(u, v) \in E$ we have an associated positive integer *flow value* f_{uv}, satisfying *conservation of flow* for every $v \in V \setminus \{s, t\}$, namely:

$$\sum_{(u,v)\in E} f_{uv} = \sum_{(v,w)\in E} f_{vw}. \tag{1}$$

Definition (k-Flow Decomposition)

A *k-flow decomposition* $(\mathcal{P}, w)$ for a flow network $G = (V, E, f)$ is a set of k s-t flow paths $\mathcal{P} = (P_1, \ldots, P_k)$ and associated weights $w = (w_1, \ldots, w_k)$, with each $w_i \in \mathbb{Z}^+$, such that for each edge $(u, v) \in E$ it holds that:

$$\sum_{\substack{i \in \{1,\ldots,k\} \text{ s.t.} \\ (u,v) \in P_i}} w_i = f_{uv}. \tag{2}$$



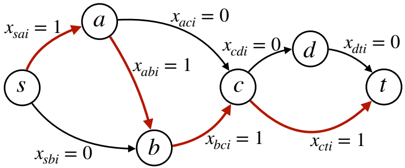
ILP Formulation

Flow Conservation

$$\sum_{(s,v)\in E} x_{svi} = 1, \qquad \forall i \in \{1,\dots,k\}, \tag{3a}$$

$$\sum_{(u,t)\in E} x_{uti} = 1, \qquad \forall i \in \{1,\dots,k\}, \tag{3b}$$

$$\sum_{(u,v)\in E} x_{uvi} - \sum_{(v,w)\in E} x_{vwi} = 0, \qquad \forall i \in \{1,\dots,k\}, \forall v \in V \setminus \{s,t\}. \tag{3c}$$



Flow Superposition

$$\sum_{i\in\{1,\dots,k\}} x_{uvi} w_i = f_{uv}, \qquad \forall (u,v) \in E. \tag{4}$$

Linearized as:

$$f_{uv} = \sum_{i\in\{1,\dots,k\}} \pi_{uvi}, \qquad \forall (u,v) \in E, \tag{5a}$$

$$\pi_{uvi} \le \overline{w} x_{uvi}, \qquad \forall (u,v) \in E, \forall i \in \{1,\dots,k\}, \tag{5b}$$

$$\pi_{uvi} \le w_i, \qquad \forall (u,v) \in E, \forall i \in \{1,\dots,k\}, \tag{5c}$$

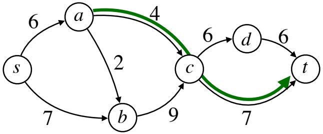
$$\pi_{uvi} \ge w_i - (1 - x_{uvi})\overline{w}, \qquad \forall (u,v) \in E, \forall i \in \{1,\dots,k\}. \tag{5d}$$

Subpath Constraints

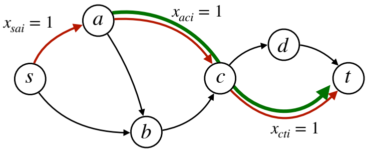
Definition (Flow decomposition with subpath constraints)

Let $G = (V, E, f)$ be a flow network. *Subpath constraints* are defined to be a set of simple paths $\mathcal{R} = \{R_1, \ldots, R_\ell\}$ in G (not necessarily s-t paths). A flow decomposition $(\mathcal{P}, w)$ *satisfies* the subpath constraints if and only if

$$\forall R_j \in \mathcal{R}, \exists P_i \in \mathcal{P} \text{ such that } R_j \text{ is a subpath of } P_i. \tag{6}$$



(a) A flow network with a single subpath constraint $R_1 = (a, c, t)$.



(b) Constraint R_1 is satisfied because for the i^{th} path we can set $r_{i1} = 1$ so that $x_{aci} + x_{cti} \geq 2r_{i1}$ holds .

Subpath Constraints - Formulation

$$\forall R_j \in \mathcal{R}, \exists P_i \in \mathcal{P} \text{ such that } R_j \text{ is a subpath of } P_i. \tag{7}$$

can be written:

$$\sum_{(u,v) \in R_j} x_{uvi} \geq |R_j| r_{ij}, \qquad \forall i \in \{1, \ldots, k\}, \forall R_j \in \mathcal{R}, \tag{8a}$$

$$\sum_{i \in \{1,\ldots,k\}} r_{ij} \geq 1, \qquad \forall R_j \in \mathcal{R}. \tag{8b}$$

Inexact Flow

UNIVERSITY OF HELSINKI

FD - ILP

Fernando H.C. Dias

Definition (Inexact flow network)

A tuple $G = (V, E, \underline{f}, \overline{f})$ is said to be an *inexact flow network* if (V, E) is a DAG with unique source s and unique sink t, where for every edge $(u, v) \in E$ we have associated two positive integer values $\underline{f_{uv}}$ and $\overline{f_{uv}}$, satisfying $\underline{f_{uv}} \leq \overline{f_{uv}}$.

Given an inexact flow network $G = (V, E, \underline{f}, \overline{f})$ the *minimum inexact flow decomposition* problem is to determine if there exists, and if so, find a minimum-size set of s-t paths $\mathcal{P} = (P_1, \ldots, P_k)$ and associated weights $w = (w_1, \ldots, w_k)$ with $w_i \in \mathbb{Z}^+$ such that for each edge $(u, v) \in E$ it holds that:

$$\underline{f_{uv}} \leq \sum_{\substack{i \in \{1,\ldots,k\} \text{ s.t.} \\ (u,v) \in P_i}} w_i \leq \overline{f_{uv}}. \quad (9)$$

Inexact Flow - Formulation

UNIVERSITY OF HELSINKI

$$\underline{f_{uv}} \leq \sum_{i \in \{1,\ldots,k\}} \pi_{uvi} \leq \overline{f_{uv}}, \qquad \forall (u,v) \in E. \tag{10a}$$

Results

Experiments Design

1. Time limit of 60 seconds;
2. Comparison of *STANDARD* with Toboggan, the implementation by [2] for their exact FPT algorithm for MFD;
3. Comparison of *SUBPATH* with Coaster, the implementation by [3] for MFDSC, which is an exact FPT algorithm extending Toboggan.
4. Comparison of *INEXACT* with IFDSolver, which is an implementation of a heuristic algorithm for MIFD by [4];
5. Three different datasets composed of RNA transcripts with range of nodes between 4 and 50;

Results

- For the *STANDARD* formulation, we could solve all instances within **20 seconds.** While the *TOBOGGAN* required at least **1 minute** for instances up to 10 flow-paths. For instances with more than 10 flow-paths, it did **not** solve within the runtime limit;
- For the *SUBPATH* version, the runtime for our formulation is below **30 seconds**, while *COASTER* cannot solve most instances;
- For the *INEXACT*, the runtime of the heuristic is **faster**, but the solution is not optimal (overestimate the number of paths by 2, in average).

Conclusions

Conclusions

- Fast: all instances were solved in under 20 seconds while state-of-the-start method requires a few minutes;
- Flexible: capable to be easily adjusted to incorporate new behaviour (as done with subpath constraints and inexact constraint) without compromising performance;
- Future Work: Cycles, improvement in inexact flow with Robust Optimization;
- `https://github.com/algbio/MFD-ILP`

Acknowlegement

UNIVERSITY OF HELSINKI

- Alexandru Tomescu[1];
- Lucia Williams,
- Brendan Mummey.

[1]This work was partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 851093, SAFEBIO), partially by the Academy of Finland (grants No. 322595, 328877, 308030), and partially by the US NSF (award 1759522).

References I

- Y. Xing, A. Resch, and C. Lee, "The multiassembly problem: reconstructing multiple transcript isoforms from est fragment mixtures," *Genome research*, vol. 14, no. 3, pp. 426–441, 2004.
- K. Kloster, P. Kuinke, M. P. O'Brien, F. Reidl, F. S. Villaamil, B. D. Sullivan, and A. van der Poel, "A practical fpt algorithm for flow decomposition and transcript assembly," in *2018 Proceedings of the Twentieth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 75–86, SIAM, 2018.
- L. Williams, A. Tomescu, B. M. Mumey, *et al.*, "Flow decomposition with subpath constraints," in *21st International Workshop on Algorithms in Bioinformatics (WABI 2021)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

References II

L. Williams, G. Reynolds, and B. Mumey, "RNA Transcript Assembly Using Inexact Flows," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1907–1914, IEEE, 2019.