Linedigraph edit distance

GRIF

Dominique Barth[1] Quentin Japhet[1] Dimitri Watel[3,4]
Marc-Antoine Weisser[2]

[1]DAVID, UVSQ, Versailles

[2]LISN, Centrale Supélec, Gif sur Yvette

[3]ensIIE, Evry

[4]Cedric-Cnam, Paris

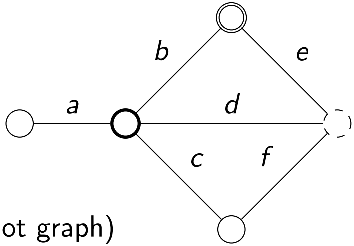
Friday, 2026 June 5th

Linegraph

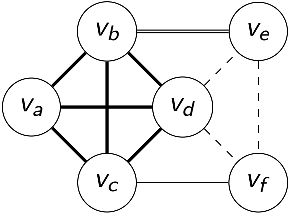
Definition

Let $G = (V, E)$ be an undirected graph, the linegraph H of G is

- $V_H = E \qquad e \in E \leftrightarrow v_e \in V_H$
- $(v_e, v_f) \in E_H \Leftrightarrow$ if e and f are incident to the same node.



G (Root graph)



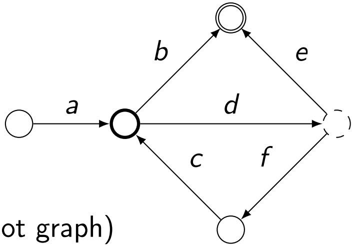
H (Linegraph)

Line-digraph

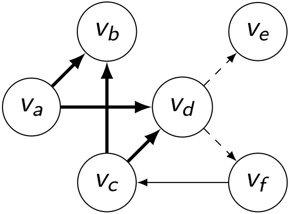
Definition

Let $G = (V, A)$ be a directed graph, the line-digraph H of G is

- $V_H = A$ $\qquad a \in A \leftrightarrow v_a \in V_H$
- $(v_a, v_b) \in A_H \Leftrightarrow$ if b follows a in G.



G (Root graph)



H (Line-digraph)

Recognition

First problems : line(-di)graph recognition

- Given a (di)graph H, is H a line(-di)graph ?

Property of linegraph

(We consider simple graphs.)

Linegraph characterization

H is a linegraph iff it contains none of the following graph as induced subgraph

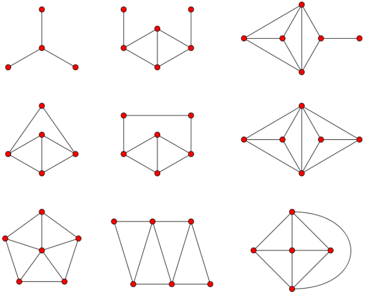


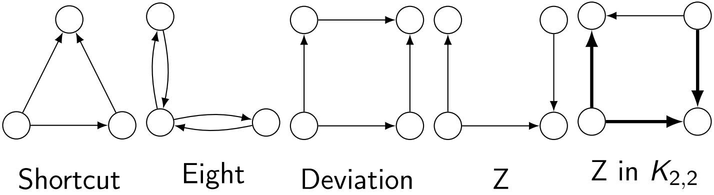
Image source : wikipedia

Property of line-digraph

(We consider digraphs with symmetric arcs but no parallel arc or loops.)

Line-digraph characterization

A graph H is a line-digraph if and only if it does not contain the Shortcut, the Eight or the Deviation as a subgraph and where every Z is contained in a $K_{2,2}$.



In what follows, we refer to them as patterns.

Root graph reconstruction

Build the root graph

Given a line-(di)graph H, find the root graph G ?

Answer: Polynomial problems.

Undirected: *P. G. H. Lehot. An Optimal Algorithm to Detect a Line Graph and Output Its Root Graph. Journal of the ACM, 21(4):569–575, oct 1974*

Directed: *J. S. Bagga1, L. W. Beineke, A survey of line digraphs and generalizations, Discrete Math. Lett. 6 (2021) 68–83*

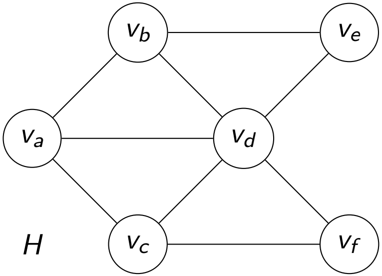
Edit distance problem

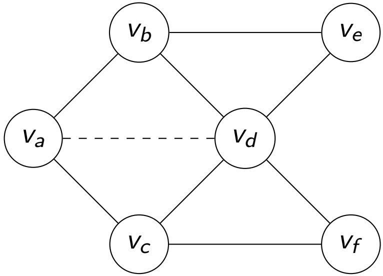
Edit distance between G and H:

- Shortest way to transform G into H by adding or removing edges/arcs.

EDL(d)G

Given a (di)graph H, find a line(-di)graph G minimizing the edit distance between G and H.





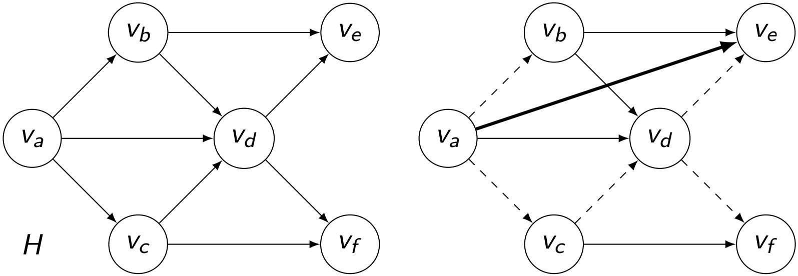
Edit distance problem

Edit distance between G and H:

- Shortest way to transform G into H by adding or removing edges/arcs.

EDL(d)G

Given a (di)graph H, find a line(-di)graph G minimizing the edit distance between G and H.



Undirected case

Complexity

EDLG is NP-Hard even if the only forbidden subgraph of H is the claw. [a]

- EDLG remains NP-Hard even if only deletion of edges is allowed. [b]
- EDLG remains NP-Hard even if only addition of edges is allowed. [c]

[a] Barth, Watel, Weisser. (2025)
[b] Yannakakis, M. (1978)
[c] Ehounou, Barth, de Moissac, Watel, Weisser. (2020)

EDLG is FPT by treewidth of the input graph H.[a]

[a] Barth, Watel, Weisser. (2025)

Directed case

Complexity (our results)

- EDLdG is NP-Hard even if H is Shortcut, Eight and Deviation free (the Z is the only forbidden pattern)
- EDLdG is NP-Hard even if every Z in H in a $K_{2,2}$ and is Shortcut and Eight free.
- These results remains even if only deletion of arcs is allowed.

- EDLdG is polynomial if only addition of arcs is allowed
- EDLdG is FPT by the number of Z in H.

Directed case

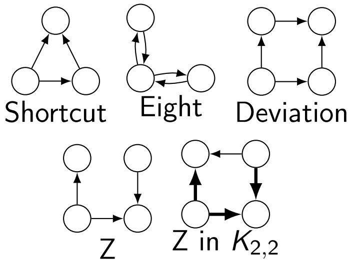
Complexity (our results)

- **EDLdG is NP-Hard even if H is Shortcut, Eight and Deviation free (the Z is the only forbidden pattern)**
- EDLdG is NP-Hard even if every Z in H in a $K_{2,2}$ and is Shortcut and Eight free.
- These results remains even if only deletion of arcs is allowed.

- **EDLdG is polynomial if only addition of arcs is allowed**
- **EDLdG is FPT by the number of Z in H.**

Polynomial case

EDLdG is polynomial if only addition of arcs is allowed



- Presence of Shortcut, Eight or Deviation $\Rightarrow$ no feasible correction.
- Presence of Z $\Rightarrow$ only one possible choice.

Add the missing arcs of the Zs until the procedure fails or succeeds.

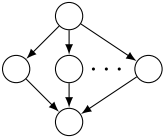
FPT algorithm

Parameterized complexity case

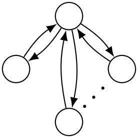
EDLdG is FPT by the number of Z in H.

Easy case : no Z

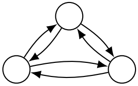
If no Z occurs in H, then the Shortcuts, Eights and Deviation may intersect in H only with the following configurations.



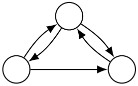
Deviations



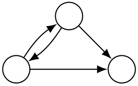
Eights



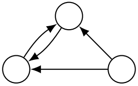
Eights Shortcuts



Eight Shortcuts



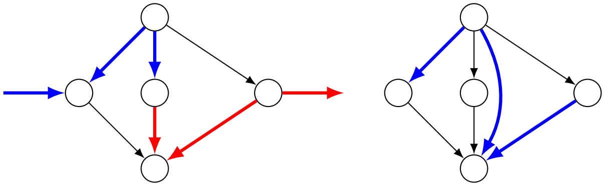
Shortcuts



Shortcuts

Easy case : no Z

Key idea : structural constraints because no Z in H



Structural constraints $\Rightarrow$ All possible configurations have independent corrections.
Each correction can be find in polynomial time.

Generalization

If a Z occurs k times in H

Branching algorithm

- Find a Z
  - Either remove one of the three arcs
  - or add the missing one (if it does not exists), lock the four arcs
  - or if the Z is already in a $K_{2,2}$, lock the four arcs. (locked arcs cannot be removed)
- Continue until all Zs are corrected.
- Process each leaf of the branching tree (see next slide)

At most $(4k)^4$ Zs can be found during the process $\Rightarrow$ number of nodes in the branching tree is FPT by k.

Generalization

Process a leaf

Each leaf is a graph where every Z is in a locked $K_{2,2}$.

Generalization of the *no Z* proof

- enumerate all the possible intersections of the Shortcuts, Eights and Deviation (arround 60-70 possible configurations)
- find the structural constraints
- most of the configurations may be solved independently.

Build an optimal correction of a leaf can be done un polynomial time. $\Rightarrow$ EDLdG is FPT by the number of Z in H.

Conclusion

Summary

- EDLdG has distinct results than EDLG.
- Determining the right correction of the Z may be a good starting point for heuristics or approximation algorithms

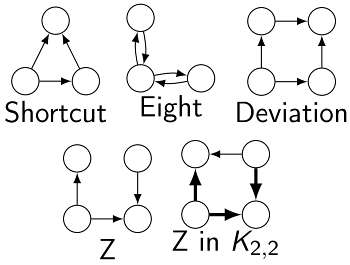
Perspectives

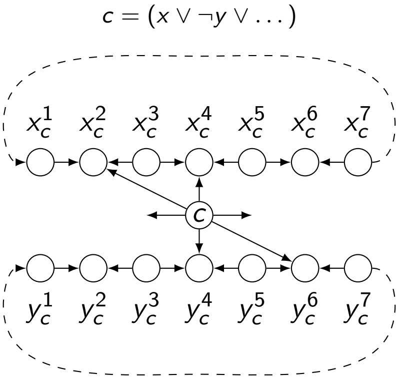
Some open questions:

- If EDLdG NP-Hard if every Z is in a $K_{2,2}$ but no Deviation occurs in H ?
- Can we generalize to non simple graphs ?
- Kernel when the parameter is the number of editions ?

NP-Hardness

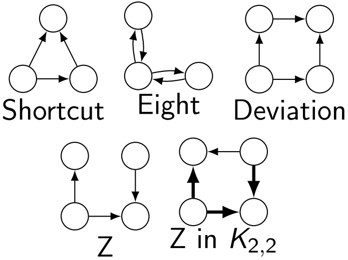
EDLdG is NP-Hard even if H is Shortcut, Eight and Deviation free.



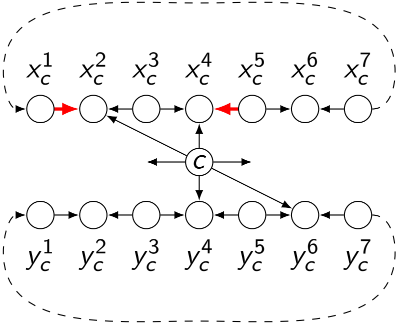


NP-Hardness

Positive correction of x

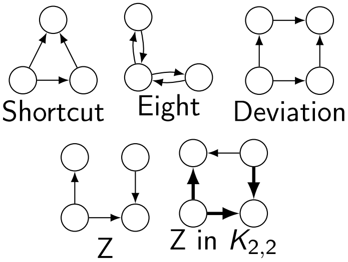


$c = (x \vee \neg y \vee \dots)$

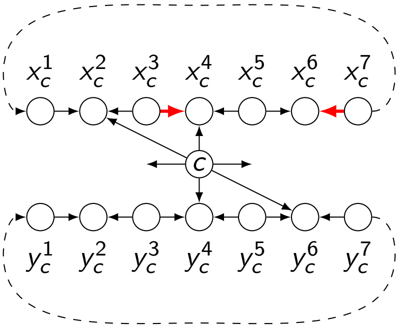


NP-Hardness

Negative correction of x

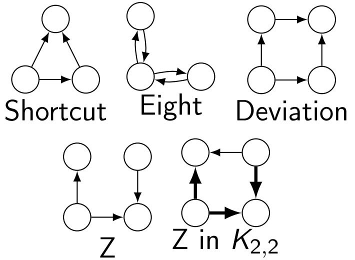


$c = (x \vee \neg y \vee \dots)$

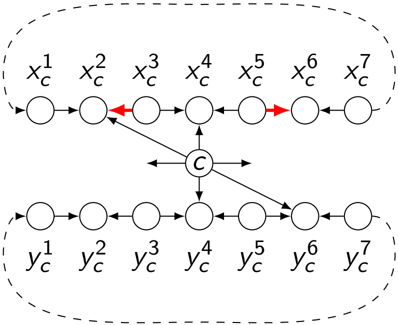


NP-Hardness

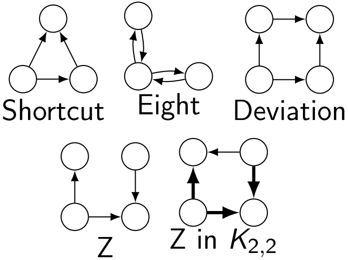
Undefined correction of x



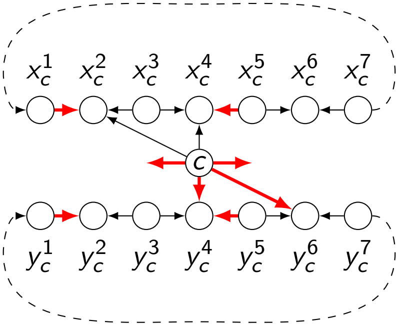
$c = (x \vee \neg y \vee \dots)$



Correction of c (with positive correction of x and y)



$c = (x \vee \neg y \vee \dots)$



NP-Hardness

Key arguments

- Adding an arc is suboptimal
- Having a variable with neither a positive, negative nor undefined correction is suboptimal.
- Having a clause where no positive (resp negative) litteral has a positive (resp negative) correction is suboptimal