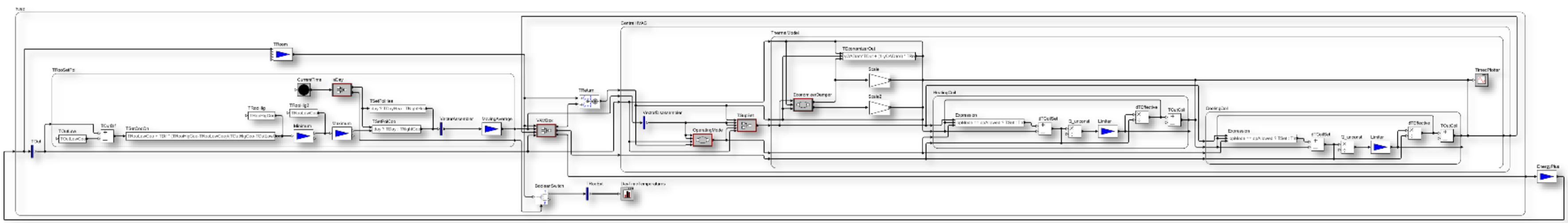
A Generalization of the Directed Graph Layering Problem

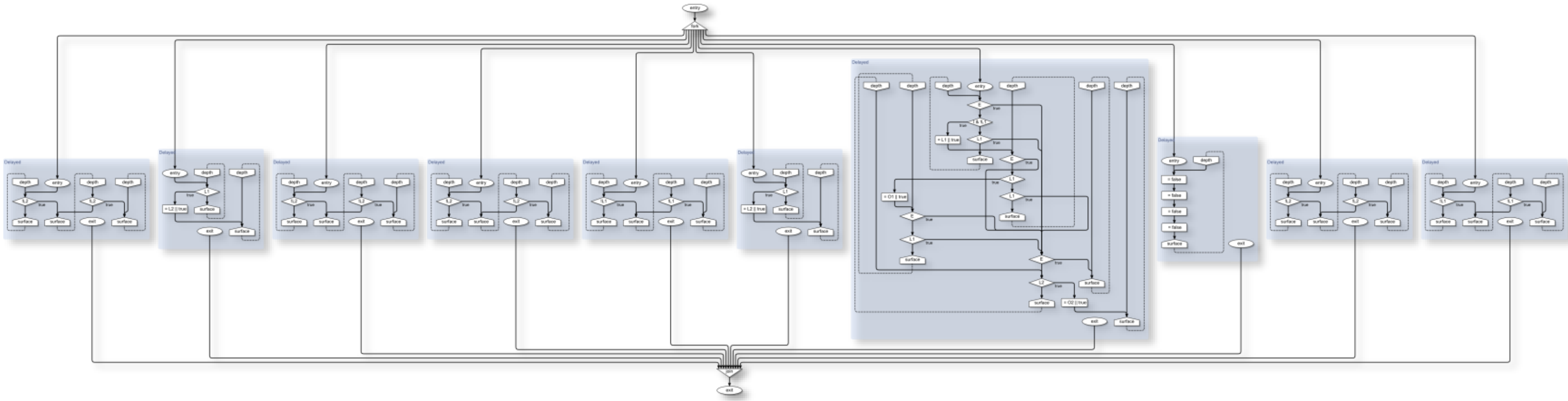
Ulf Rüegg, Thorsten Ehlers,
Miro Spönemann, and Reinhard von Hanxleden

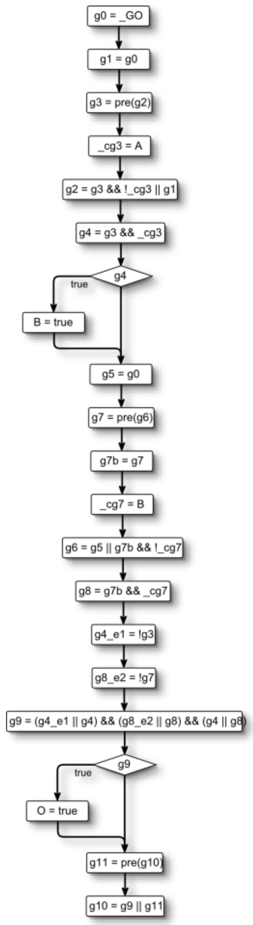
Department of Computer Science

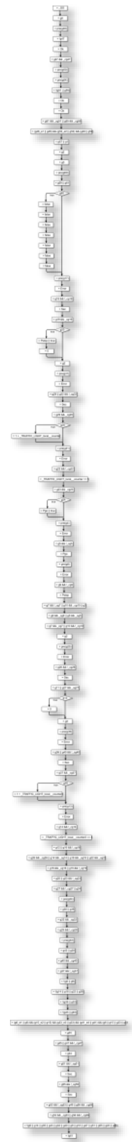
Kiel University

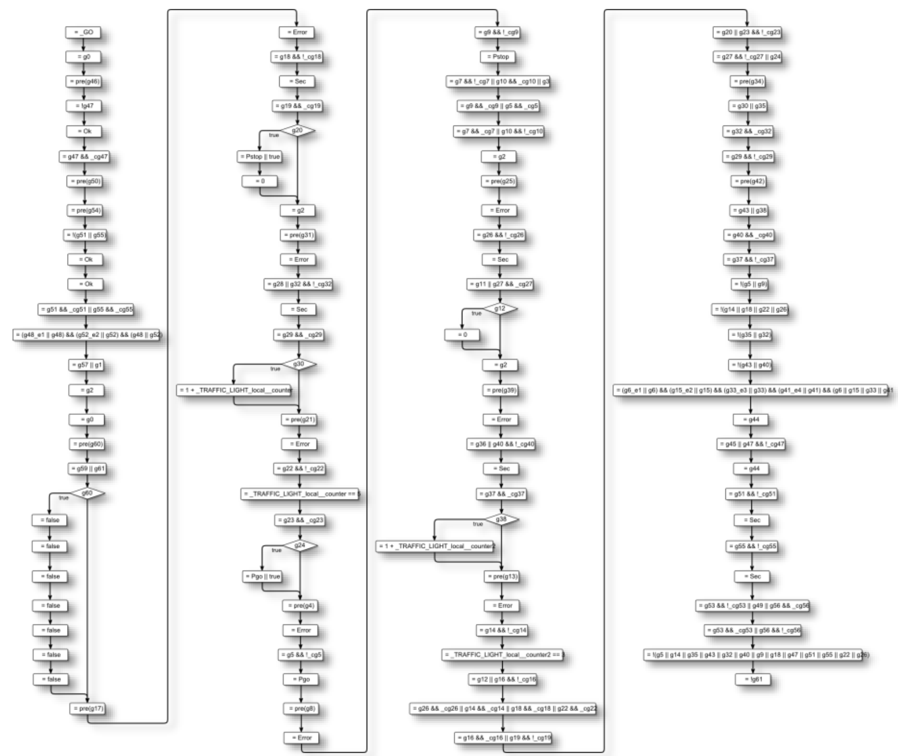


The Motivation


entry
fork
Delayed
depth
surface
true
exit


g0 = _GO
g1 = g0
g3 = pre(g2)
_cg3 = A
g2 = g3 && !_cg3 || g1
g4 = g3 && _cg3
g4
true
B = true
g5 = g0
g7 = pre(g6)
g7b = g7
_cg7 = B
g6 = g5 || g7b && !_cg7
g8 = g7b && _cg7
g4_e1 = !g3
g8_e2 = !g7
g9 = (g4_e1 || g4) && (g8_e2 || g8) && (g4 || g8)
g9
true
O = true
g11 = pre(g10)
g10 = g9 || g11





Problem

Diagrams in practice often have

an unfortunate aspect ratio

resulting in

poor utilization of the drawing area

small diagram elements

a lot of whitespace

C. Gutwenger, R. von Hanxleden, P. Mutzel, U. Rüegg, and M. Spönemann
Examining the compactness of automatic layout algorithms for practical diagrams, GraphViP 2014

SUGIYAMA LAYOUT

P1 Cycle Removal

P2 Layer Assignment

P3 Crossing Minimization

Topological Phases

P4 Node Placement

P5 Edge Routing

Geometrical Phases

K. Sugiyama, S. Tagawa, M. Toda
Methods for visual understanding of hierarchical system structures, IEEE Trans. on Syst., Man and Cybernetics, 1981.

C. D. Schulze, M. Spönemann, R. v. Hanxleden
Drawing layered graphs with port constraints
Journal of Visual Languages and Computing, 2013

ALTERING THE ASPECT RATIO

Which phase has influence?

P3 Crossing Minimization

(almost) nothing to do with it

P4 Node Placement

straight edges may increase width

P5 Edge Routing

edges may occupy space between layers

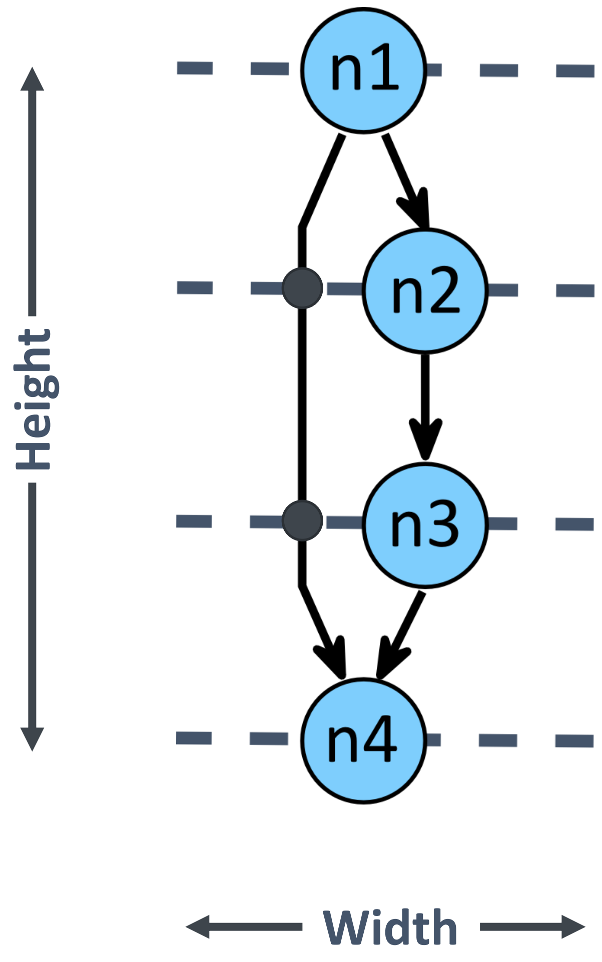
certainly have an impact,
still, the topology is already fixed

This leaves us with ...

P1 Cycle Removal

P2 Layer Assignment

Layering



estimated

vs

effective

Existing Work

Layering

P. Eades and K. Sugiyama,
How to draw a directed graph, Journal of Information Processing, 1990.

E. R. Gansner, E. Koutsofios, S. C. North, and K.-P. Vo,
A technique for drawing directed graphs, *IEEE Transactions on Software Engineering*, 1993.

E. G. Coffman and R. L. Graham,
Optimal scheduling for two-processor systems, Acta Informatica, 1972.

"Minimum Width"

P. Healy and N. Nikolov,
How to layer a directed acyclic graph, Graph Drawing, 2001.

P. Healy and N. Nikolov,
A branch-and-cut approach to the directed acyclic graph layering problem, Graph Drawing, 2002.

N. Nikolov, A. Tarassov, and J. Branke,
In search for efficient heuristics for minimum-width graph layering with consideration of dummy nodes, Journal of Experimental Algorithmics, 2005.

Aspect Ratio

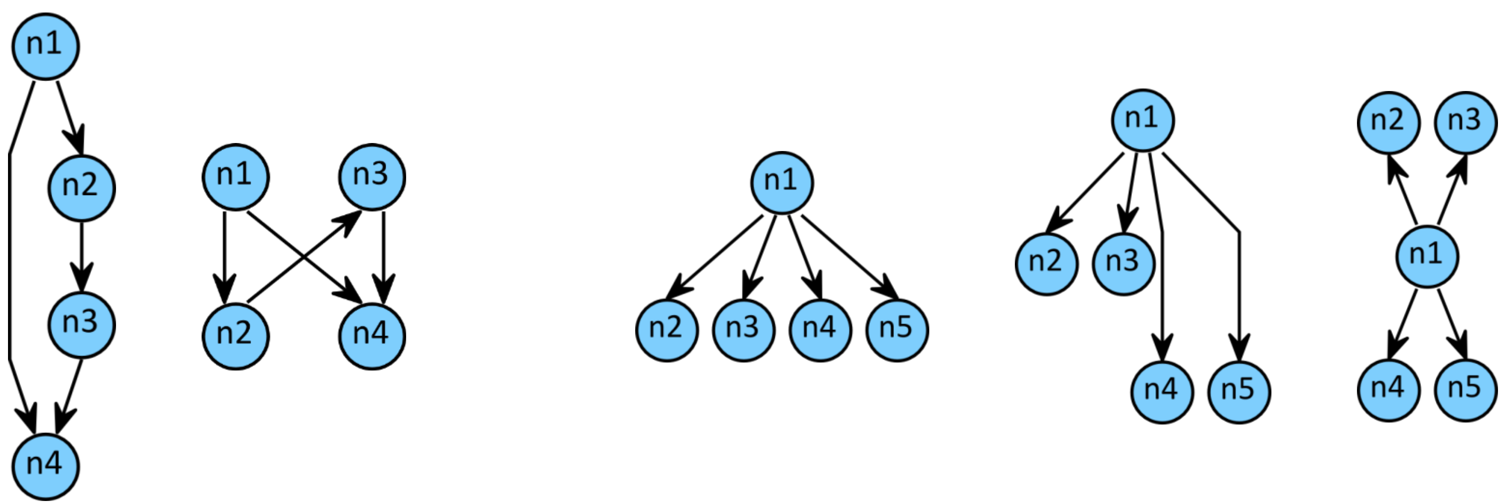
L. Nachmanson, G. Robertson, and B. Lee,
Drawing graphs with GLEE, Graph Drawing, 2008.

Not Addressed so Far

Input graph must be acyclic

1 Height bound by the longest path

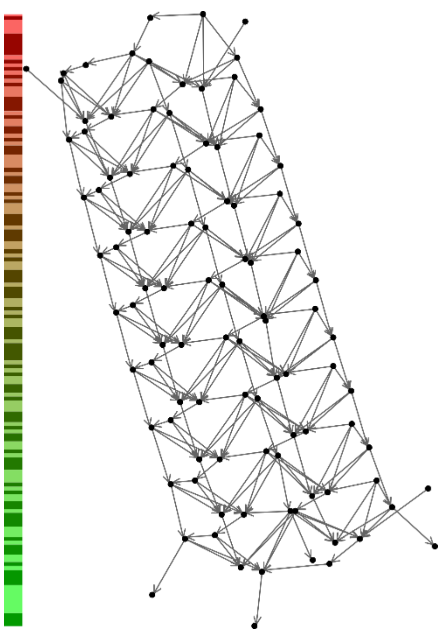
2 Width bound by highest node degree



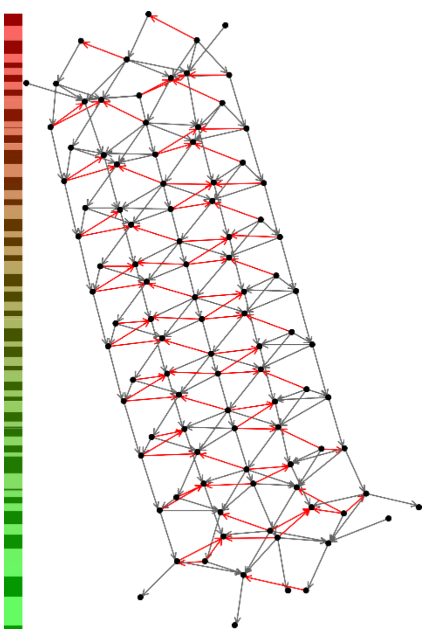
DIG-COLA

"We have found that allowing small deviation sometimes gives us the necessary freedom for overcoming local inefficiencies in the layout, without affecting visualization of the overall directionality."

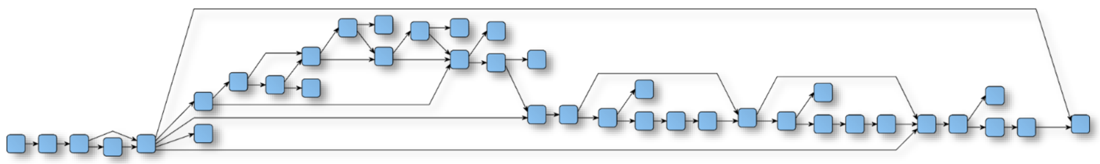
T. Dwyer and Y. Koren,
DIG-COLA: Directed Graph Layout through Constrained Energy Minimization, INFOVIS, 2005.

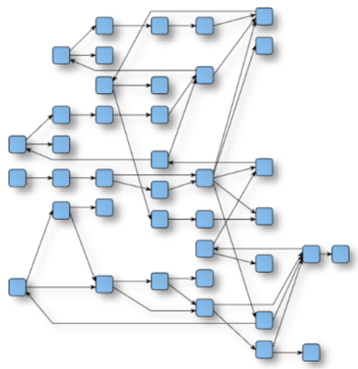


all edges point downwards



some edges point upwards
less crossings





SOMETHING NEW

Warning

From now on, drawing are **left to right** instead of **top down**.

Hence ...

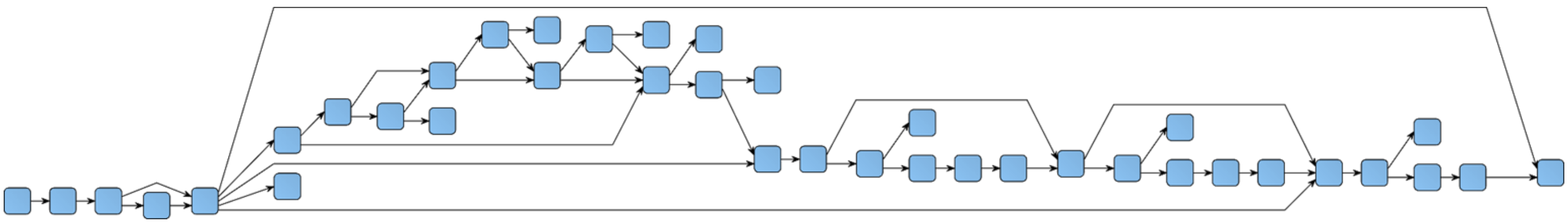
Height is **Width**

and

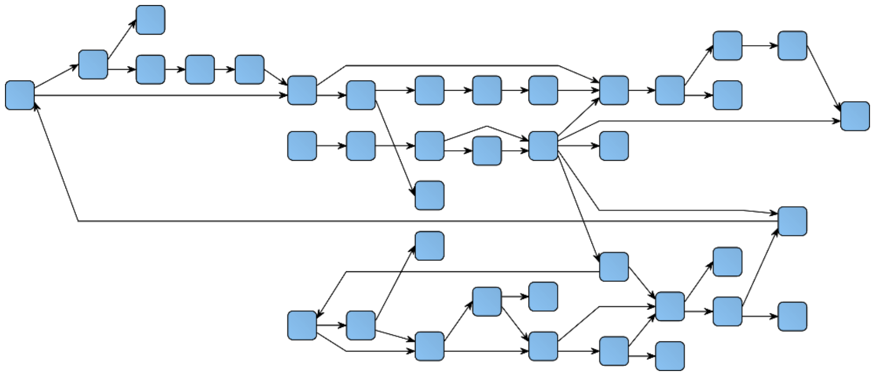
Width is **Height**

(I will most definitely use them wrongly at some point ...)

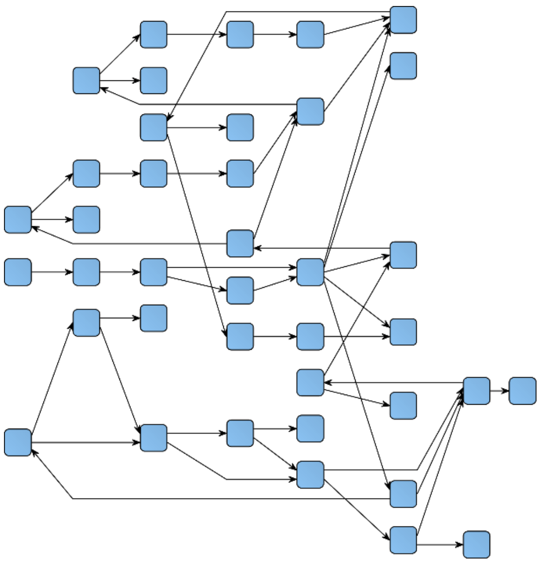
OBSERVATION



0 reversed edges, 71 edge length



2 reversed edges, 35 edge length



6 reversed edges, 16 edge length

DIRECTED LAYERING (DLP)

Given a directed acyclic graph $G = (V, E)$,
find a *valid* layering $L: V \to \mathbb{N}$.

MINIMIZE

$$\sum_{(v,w)\in E} L(w) - L(v)$$

SUBJECT TO

$$L(w) - L(v) \geq 1 \qquad \forall (v, w) \in E$$

E. R. Gansner, E. Koutsofios, S. C. North, and K.-P. Vo,
A technique for drawing directed graphs,
IEEE Transactions on Software Engineering, 1993.

Generalized Layering (GLP)

Given a directed ~~acyclic~~ graph $G = (V, E)$,
find a *feasible* layering $L: V \rightarrow \mathbb{N}$.

Minimize

$$w_{len}\left(\sum_{(v,w)\in E} |L(w) - L(v)|\right) + w_{rev}|\{(v,w) \in E: L(v) > L(w)\}|$$

Subject to

$$|\boldsymbol{L}(\boldsymbol{w}) - \boldsymbol{L}(\boldsymbol{v})| \geq 1 \quad \forall (v,w) \in E$$

IP Model

INPUT

Graph $G = (V, E)$ — $\omega_{rev}, \omega_{len}$ weighting constants

INTEGER VARIABLES

$l(v)$ — $\forall\, v \in V$ — takes values in $\{1, \dots, n\}$ ($|V| = n$) indicating node v is placed in layer $l(v)$

BOOLEAN VARIABLES

$r(u, v)$ — $\forall\, (u, v) \in E$ — 1 iff $e(u, v) \in E$ reversed, 0 otherwise

MINIMIZE

$$\omega_{len} \sum_{(u,v)\in E} |l(u) - l(v)| \;+\; \omega_{rev} \sum_{(u,v)\in E} r(u, v)$$

SUBJECT TO

$$1 \le l(v) \le n \qquad \forall\, v \in V$$

$$|l(u) - l(v)| \ge 1 \qquad \forall\, (u, v) \in E$$

$$n \cdot r(u, v) + l(v) \ge l(u) + 1 \qquad \forall\, (u, v) \in E$$

HEURISTIC

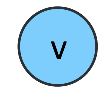
1. Remove leaf nodes iteratively
2. Construct initial feasible layering, deduce edge directions
3. Run DLP
4. Greedily improve result
5. Reattach leaf nodes

LAYERING CONSTRUCTION

STEP 2

IDEA

Place nodes on a line (assign indexes),
try to keep edges short
and the number of left edges small



u, a_i, a_o

u unassigned neighbors
a_i assigned, incoming
a_o assigned, outgoing

LOOP

select candidate with smallest |**u**| (start with random node)
if |**a_i**| < |**a_o**|
  assign index to the left of currently placed nodes
else
  assign to the right

A. J. McAllister,
A new heuristic algorithm for the linear arrangement problem, Technical Report, University of New Brunswick, 1999.

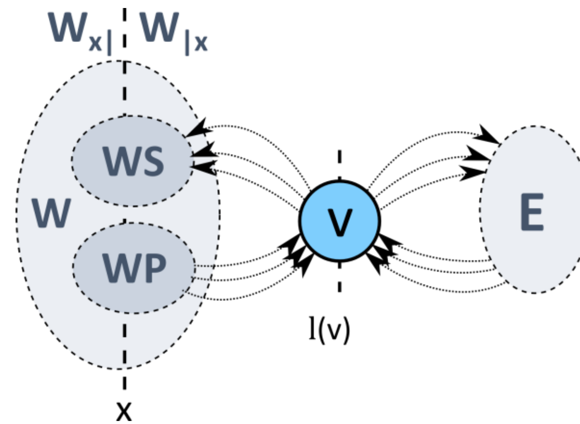
J. Pantrigo, R. Mart, A. Duarte, and E. Pardo,
Scatter search for the cutwidth minimization problem, Annals of Operations Research, 2012.

Layering Improvement

Step 4

Idea

Identify badly placed nodes
and move them to a different layer

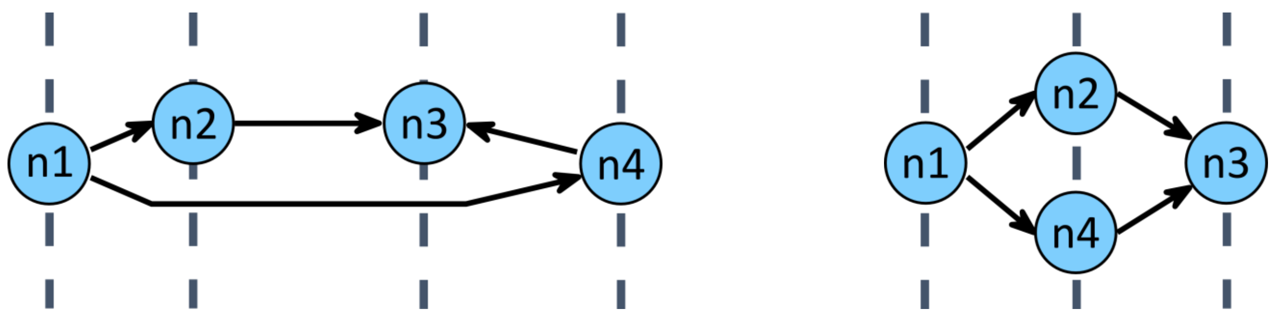


Move

$$\text{move}(v) = \begin{cases} 0, & \text{if } \mathbf{WS} = \emptyset \\ l(v) - \min\big(l(\mathbf{WS})\big) + 1, & \text{if } \mathbf{WP} = \emptyset \\ l(v) - \max\big(l(\mathbf{WP})\big) - 1, & \text{else} \end{cases}$$

Profit

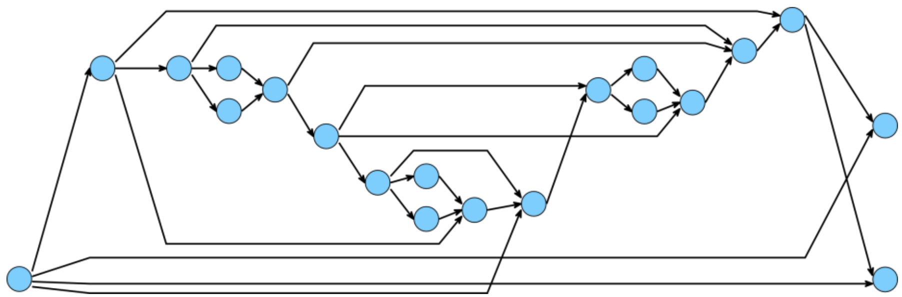
$$\text{profit}(v, m, x) = \begin{cases} 0, & \text{if } m \leq 1 \\ m \cdot (|\mathbf{W}_{x|}| - |\mathbf{E}|) + |\mathbf{WS}_{|x}|, & \text{else} \end{cases}$$



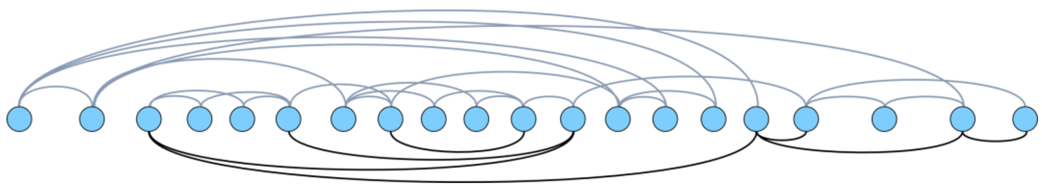
$\text{move}(\text{n4}) = 2$ $\text{profit}(\text{n4}, 2, 2) = 2 + 1 = 3$

EXAMPLE

ORIGINAL

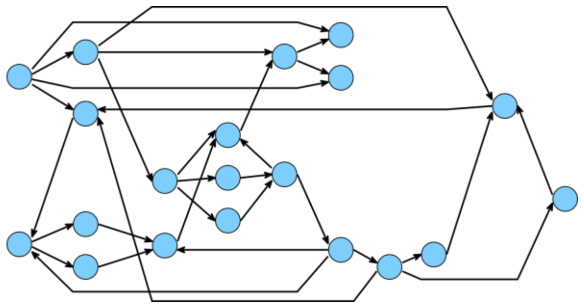


CONSTRUCTION

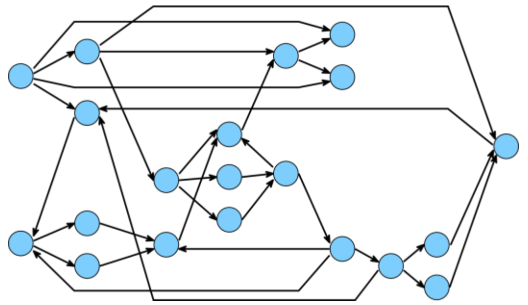


DLP

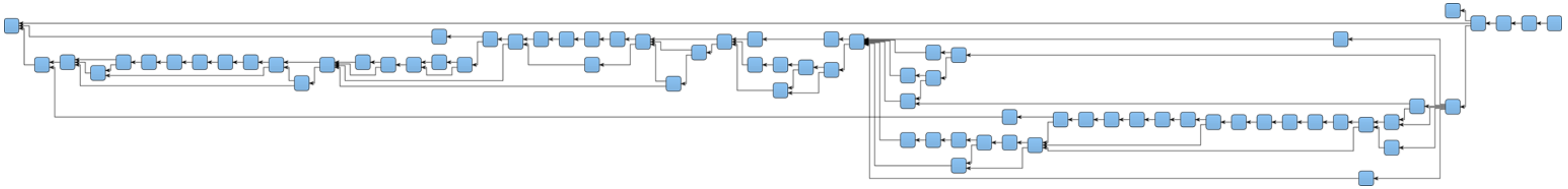
& IMPROVEMENT

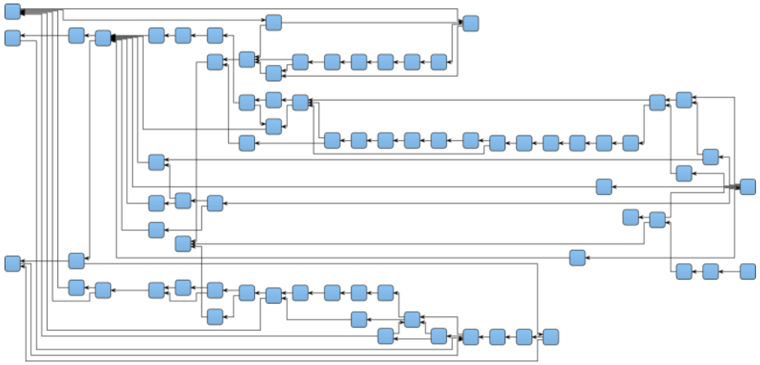


7 left edges, 34 edge length



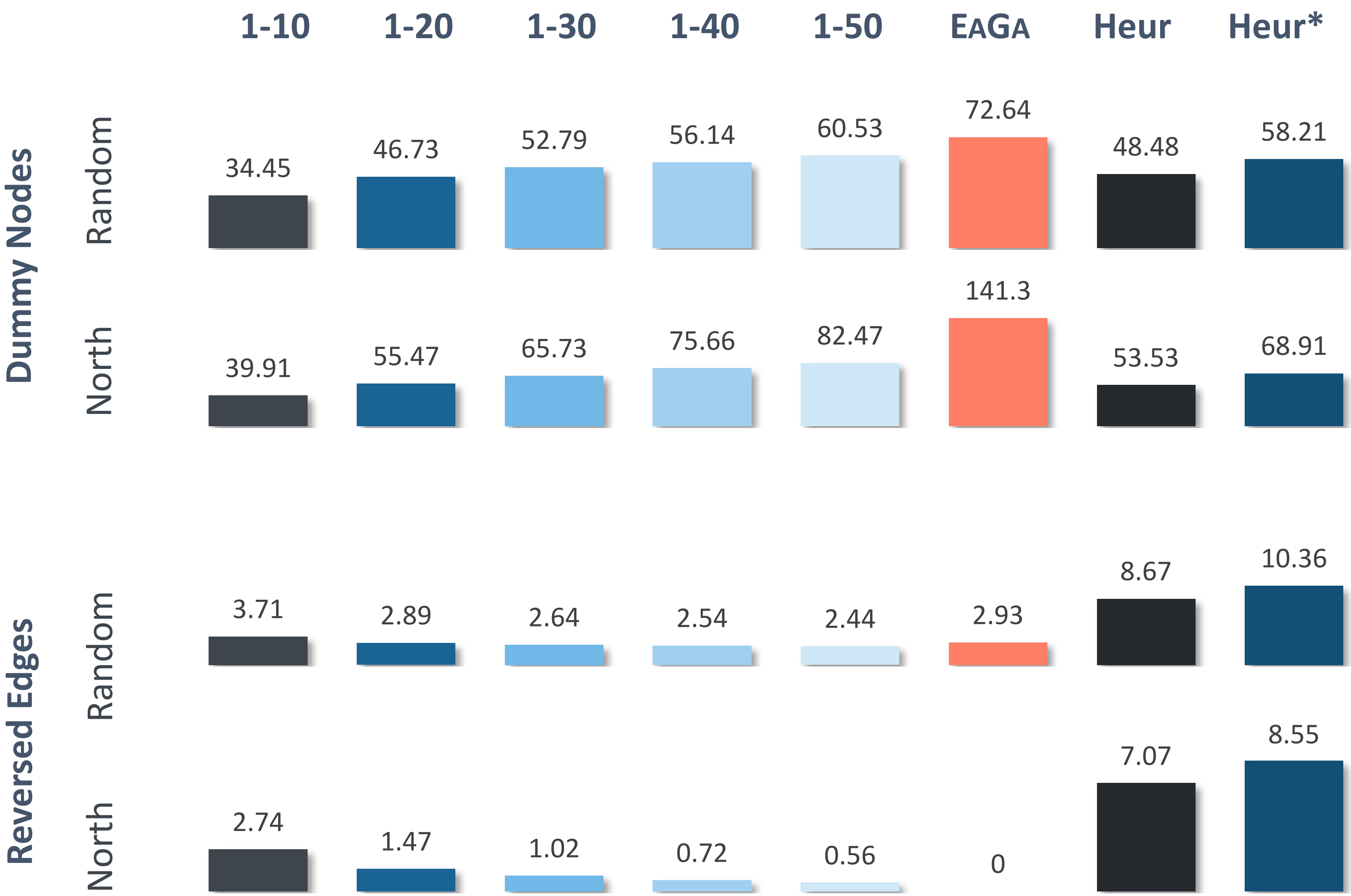
6 left edes, 32 edge length





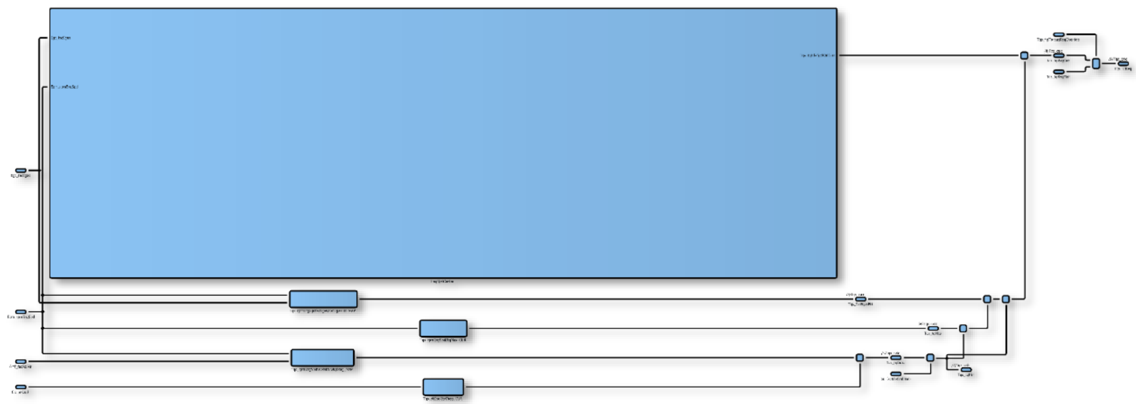
RESULTS

Topological Metrics



Geometrical Metrics

		1-10	1-20	1-30	1-40	1-50	EaGa	Heur	Heur*
Width (px)	Random	843	943	980	1,004	1,025	1,084	930	1,027
	North	1,068	1,224	1,334	1,409	1,469	1,727	1,137	1,216
Area (px)	Random	631,737	672,717	691,216	700,385	708,361	737,159	656,070	720,798
	North	587,727	622,838	641,581	660,842	695,494	874,374	629,778	691,372



Wrap Up

NOTES

GLP does not consider node sizes and is too slow

Heuristic fails for paths and trees

Averages of estimated and effective values show the same tendency

Still, for 64% of the graphs: estimated increased and effective decreased

QUESTIONS

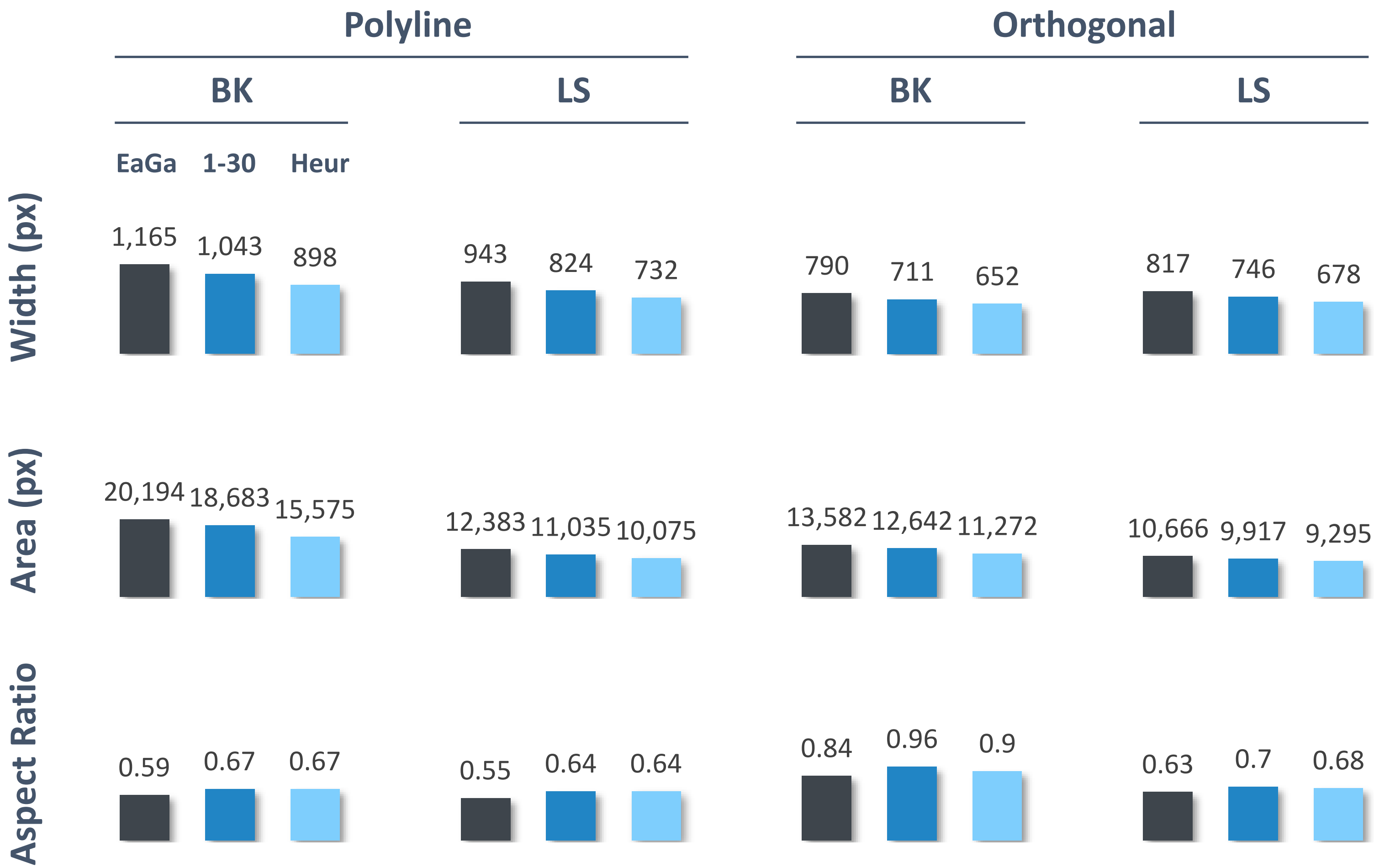
Do people accept reversed edges?

Are there better metrics for compactness?

Does an iterative procedure/portfolio of algorithms help?

APPENDIX

Different Phase Strategies



Random graphs