Analysing Kauffman Boolean Networks

PAVEL EMELYANOV

† Institute of Informatics Systems and Novosibirsk State University

May 31, 2016

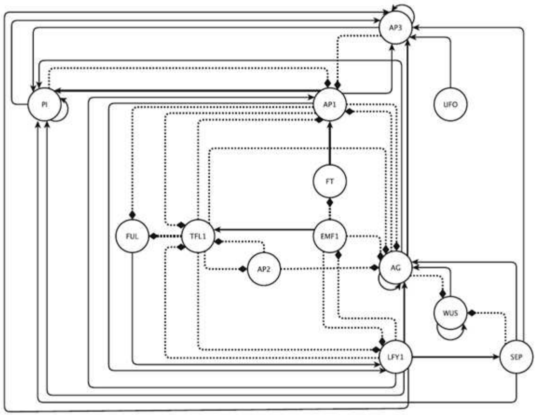
Kauffman's Boolean Networks

Boolean networks (BNs) originated in

- S. A. Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224(215):177–178, 1969.
- S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.

are well-known in the scope of modeling complex systems of different kinds: regulatory networks, cell differentiation, evolution, immune response, neural networks, social networks, interactions over WWW.

Boolean Network for Flowers of Arabidopsis Thaliana



Y.-E. Sanchez-Corrales, E.R. Alvarez-Buylla, L. Mendoza. The Arabidopsis Thaliana flower organ specification gene regulatory network determines a robust differentiation process. *J. Theor. Biol.*, 264:971-983, 2010.

AG=(not EMF1 and not AP2 and not TFL1)
or (not EMF1 and not AP1 and LFY)
or (not EMF1 and not AP2 and LFY)
or (not EMF1 and not TFL1 and LFY
and (AG and SEP))
or (not EMF1 and (LFY and WUS))

AP1=(not AG and not TFL1)
or (FT and LFY and not AG)
or (FT and not AG and not PI)
or (LFY and not AG and not PI)
or (FT and not AG and not AP3)
or (LFY and not AG and not AP3)

AP2=not TFL1

AP3=(LFY and UFO)
or (PI and SEP and AP3 and (AG or AP1))

EMF1=not LFY

FT=not EMF1

FUL=not AP1 and not TFL1

LFY=(not EMF1)
or (not TFL1)

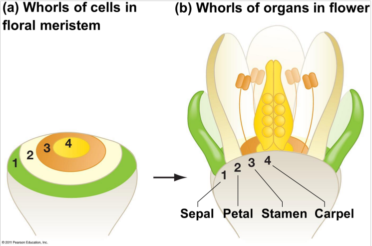
PI=(LFY and (AG o AP3))
or (PI and SEP and AP3 and (AG or AP1))

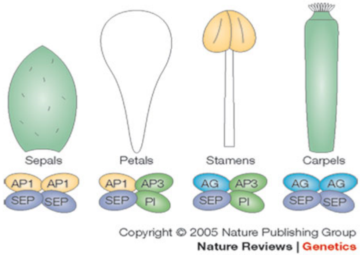
SEP=LFY

TFL1=not AP1 and (EMF1 and not LFY)

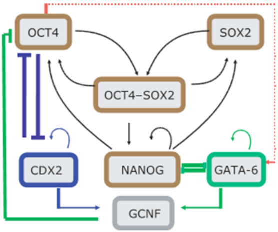
WUS=WUS and (not AG or not SEP)

Examples of Differentiation for Arabidopsis Thaliana





Examples of Differentiation for Stem Cells



OS	CDX2	NANOG	GCNF	Result:OCT4
0	0	1	0	1
1	0	0	0	1
1	0	1	0	1

OS	NANOG	Result:SOX2
0	1	1
1	0	1
1	1	1

CDX2	GCNF	Result:GCNF
0	1	1
1	0	1
1	1	1

OCT4	SOX2	Result:OS
1	1	1

OCT4	CDX2	Result:CDX2
0	1	1

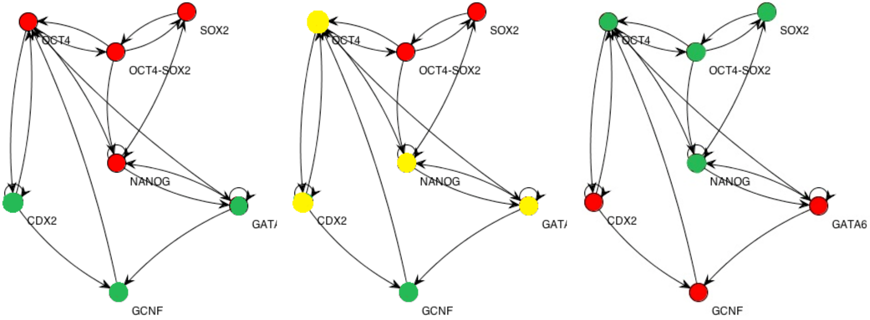
OS	NANOG	GATA6	Result:NANOG
0	1	0	1
1	0	0	1
1	1	0	1

OS	NANOG	GATA6	Result:GATA6
0	0	1	1
1	0	0	1
1	0	1	1

Fixpoints of Differentiation for Stem Cells

OCT4	SOX2	OS	NANOG	GATA6	CDX2	GCNF
0	0	0	0	0	0	0
1	1	1	0	1	0	0
0	1	0	1	1	0	0
0	0	0	0	1	1	1
0	0	0	0	0	1	1
0	0	0	0	1	0	1
1	1	1	1	0	0	0

One Examples of Differentiation for Stem Cells



Problems of Interest

Given a boolean map $F : \{0,1\}^n \to \{0,1\}^n$. We are interested in:

- finding fixpoints of the map (also called singleton attractors), i. e. finding points $x \in \{0,1\}^n$ such that $x = F(x)$;

Problems of Interest

Given a boolean map $F : \{0,1\}^n \to \{0,1\}^n$. We are interested in:

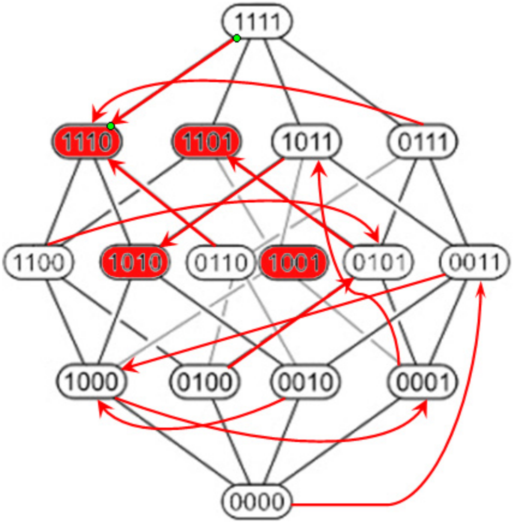
- finding fixpoints of the map (also called singleton attractors), i. e. finding points $x \in \{0,1\}^n$ such that $x = F(x)$;
- finding k-cycles of the map (also called cyclic attractors), i. e. finding points $x \in \{0,1\}^n$ such that $x = F^k(x)$ and $k > 1$ obeying this identity is minimal, where $F^{k+1} = F \circ F^k, F^1 = F$;

Problems of Interest

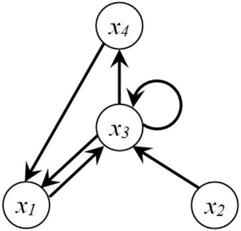
Given a boolean map $F : \{0,1\}^n \to \{0,1\}^n$. We are interested in:

- finding fixpoints of the map (also called singleton attractors), i. e. finding points $x \in \{0,1\}^n$ such that $x = F(x)$;
- finding k-cycles of the map (also called cyclic attractors), i. e. finding points $x \in \{0,1\}^n$ such that $x = F^k(x)$ and $k > 1$ obeying this identity is minimal, where $F^{k+1} = F \circ F^k, F^1 = F$;
- finding basins of attractors, both singleton and cyclic, i. e. finding sets of points $x \in \{0,1\}^n$ such that after some number of iterations of the map F it falls into the corresponding attractor.

Boolean Map as Functional Network



$$\begin{cases} x_1 = x_3 \vee x4 \\ x_2 = x_2 \\ x_3 = x_3 \oplus (\overline{x_1} \cdot \overline{x_2}) \\ x_4 = \overline{x_3} \end{cases}$$



Problem Complexity

The NP-hardness of BNs fixpoint problem was independently established in

- T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. *Genome Informatics*, 9:151–160, 1998.
- M. Milano and A. Roli. Solving the satisfiability problem through boolean networks. In *Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence*, volume 1792 of *Lecture Notes in Artificial Intelligence*, pages 72–83. Springer-Verlag, 1999.

Solving BNs Fixpoint Problems

To solve the fixpoint problem different techniques were developed (non-exhaustive reference list):

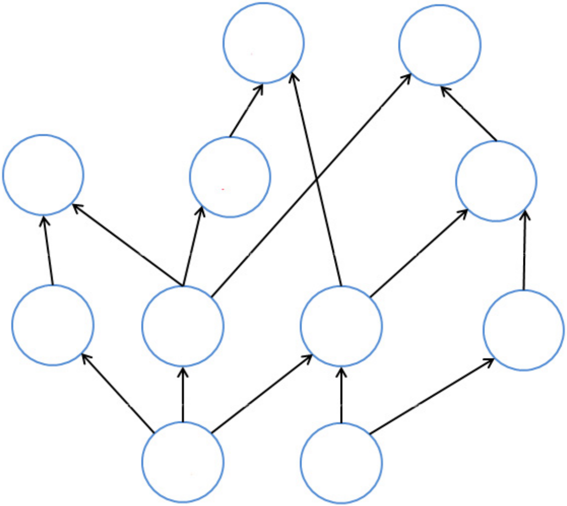
- boolean formulae satisfiability with SAT-solvers [Dubrova, Teslenko, 2011];
- abstract interpretation of dynamical systems [Paulevé, Magnin, Roux, 2012];
- Petri nets modeling [Steggles, Banks, Shaw, Wipat, 2007];
- matrix algebras computations [Cheng, Qi, Zhao, 2012];
- graph-theoretical decompositions [Zhang, Hayashida, Akutsu, Ching, Ng, 2007; Soranzo, Iacono, Ramezani, Altafini, 2012].

What We Do

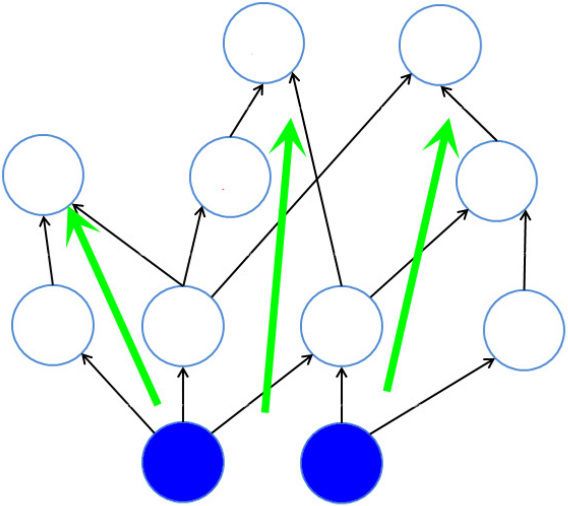
Our approach consists of:

- decomposition of an original network into smaller networks (this talk);
- solving Fixpoint Problem for each small network;
- reconstruction solution for the entire network from "small" sub-solutions.

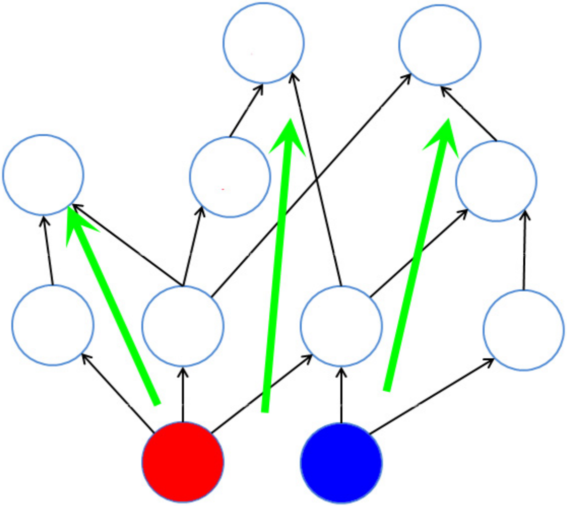
Idea: Acyclic Case



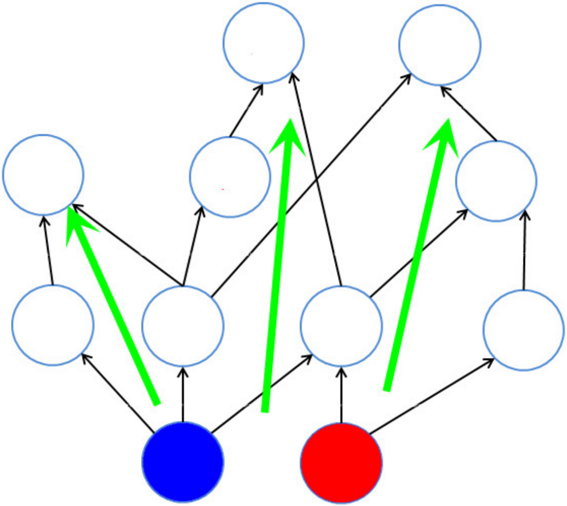
Idea: Acyclic Case



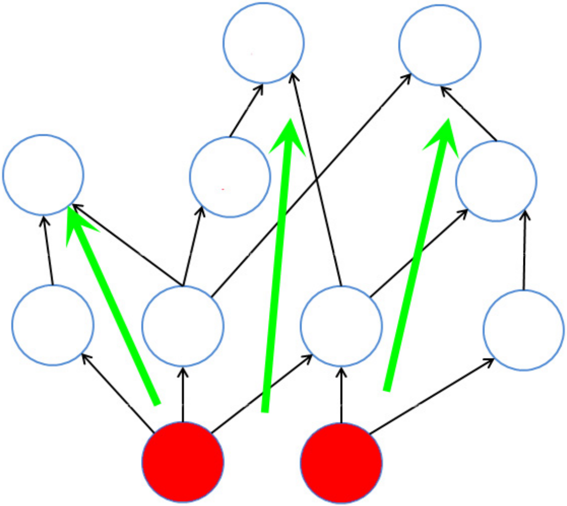
Idea: Acyclic Case



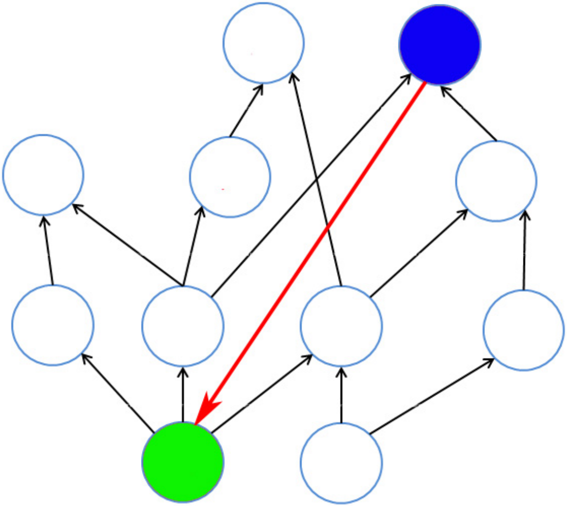
Idea: Acyclic Case



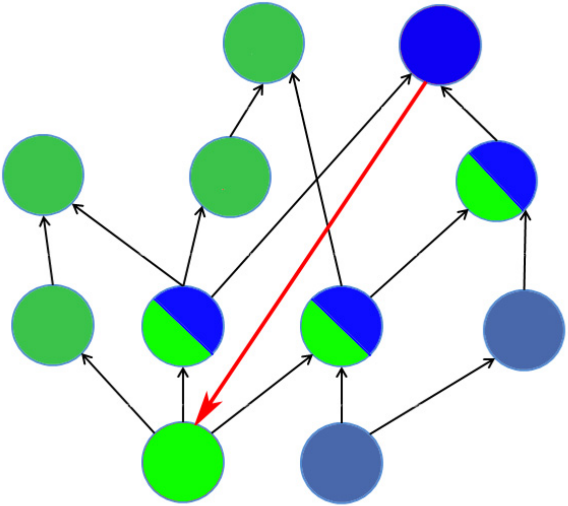
Idea: Acyclic Case



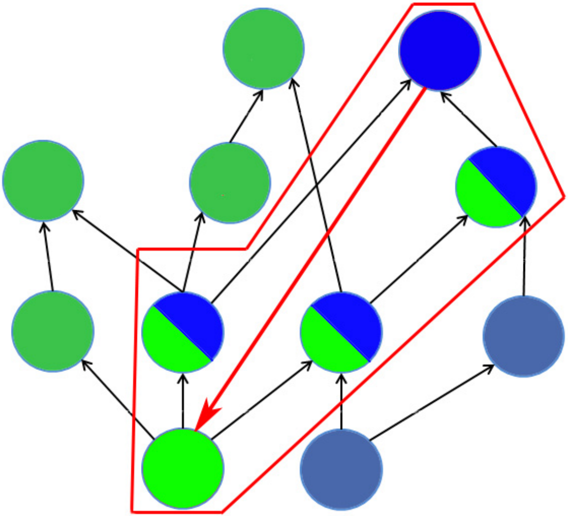
Idea: Add One Feedback Arc



Idea: Add One Feedback Arc



Idea: Add One Feedback Arc

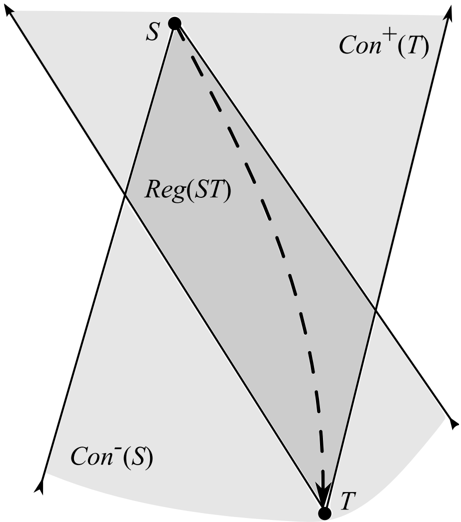


Feedback (Arc) Region

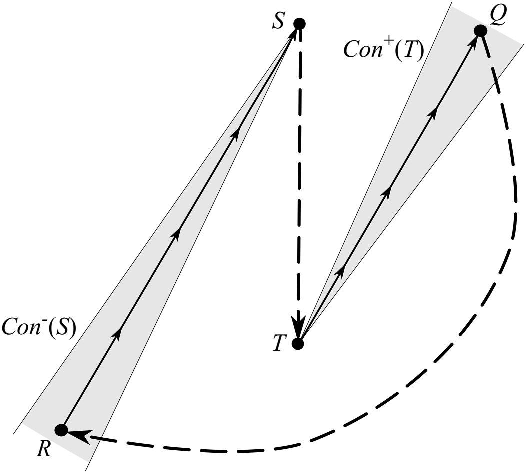
For each feedback arc ST let us consider next vertices belonging to the graph G without all feedback arcs:

- *upper cone* of the arc end $Con^{+}(T)$ is a set of all vertices being reachable from the end of the feedback arc;
- *lower cone* of the arc start $Con^{-}(S)$ is a set of all vertices reaching the start of the feedback arc.
- $Reg(ST) = Con^{+}(T) \cap Con^{-}(S)$.

Feedback Region

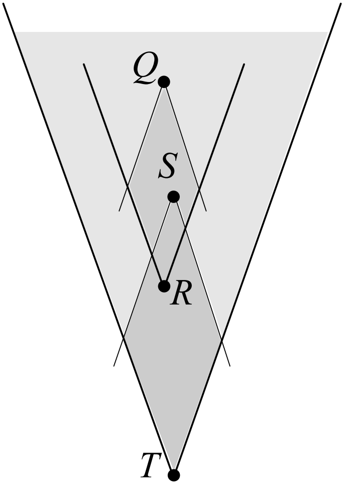


Big Region



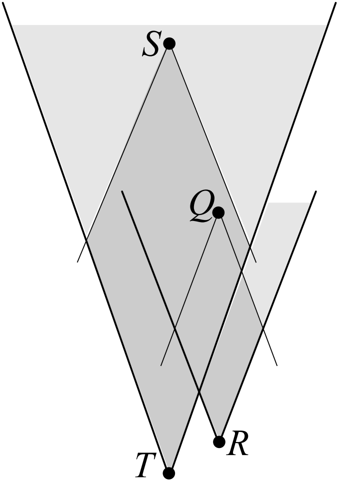
Big Region includes all vertices of feedback upper and lower cones which are disjoint.

Region Interaction: Simple Influence



One region is contained within a zone of influence (upper cone) of another region:
$R \in Con^{+}(T) \quad \vee$
$T \in Con^{+}(R)$

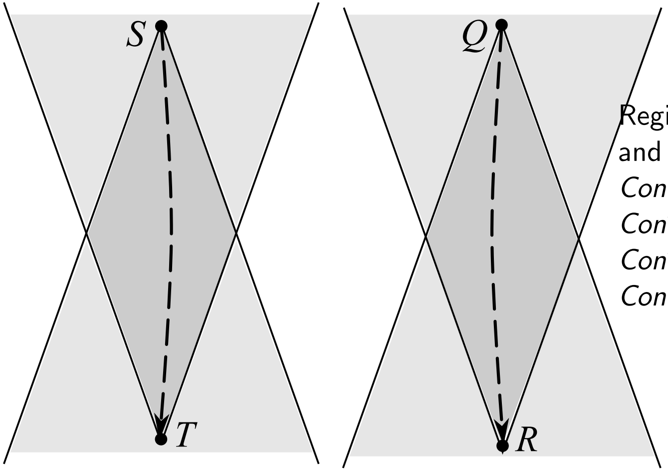
Region Interaction: Tangled



One region is *partially* influenced by another (only part of a region lies within upper cone of another):

$R \notin Con^{+}(T) \ \wedge$
$T \notin Con^{+}(R) \ \wedge$
$(Q \in Con^{+}(T) \ \vee$
$S \in Con^{+}(R))$

Region Interaction: Disjoint



Regions are disjoint and do not interact:
$Con^{-}(S) \cap Con^{+}(R) = \emptyset \quad \wedge$
$Con^{+}(T) \cap Con^{-}(Q) = \emptyset$

Feedback Arc Set Problem

- Given a directed graph $G = (V, A)$.

Feedback Arc Set Problem

- Given a directed graph $G = (V, A)$.
- We need to find its *maximum acyclic spanning subgraph* $G_1 = (V, A_0)$.

Feedback Arc Set Problem

- Given a directed graph $G = (V, A)$.
- We need to find its *maximum acyclic spanning subgraph* $G_1 = (V, A_0)$.
- Arcs excluded from this subgraph ($A_1 = A \setminus A_0$) are called feedback arcs. This gives a name for the complementary problem: finding a *minimum feedback arc set* (**MinFAS**). It is not unique.

Feedback Arc Set Problem

- Given a directed graph $G = (V, A)$.
- We need to find its *maximum acyclic spanning subgraph* $G_1 = (V, A_0)$.
- Arcs excluded from this subgraph ($A_1 = A \setminus A_0$) are called feedback arcs. This gives a name for the complementary problem: finding a *minimum feedback arc set* (**MinFAS**). It is not unique.
- In general this problem is hard to solve. Therefore we need an efficient algorithm finding a correct (upper) approximation of FAS.

FAS Algorithmics

- R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

FAS Algorithmics

- R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- B. Berger and P. W. Shor. Approximation algorithms for the maximum acyclic subgraph problem. In *Proceedings of First ACM–SIAM Symposium on Discrete Algorithms*, pages 236–243. ACM Press, 1990.

FAS Algorithmics

- R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- B. Berger and P. W. Shor. Approximation algorithms for the maximum acyclic subgraph problem. In *Proceedings of First ACM–SIAM Symposium on Discrete Algorithms*, pages 236–243. ACM Press, 1990.
- P. Eades, X. Lin, and W. F. Smyth. A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323, 1993.

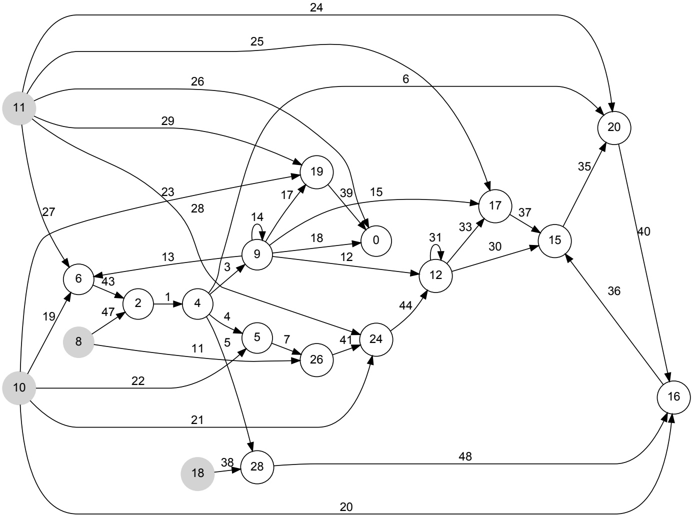
FAS Algorithmics

- R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
- B. Berger and P. W. Shor. Approximation algorithms for the maximum acyclic subgraph problem. In *Proceedings of First ACM–SIAM Symposium on Discrete Algorithms*, pages 236–243. ACM Press, 1990.
- P. Eades, X. Lin, and W. F. Smyth. A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323, 1993.
- G. Even, J. Noar, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20:151–174, 1998.

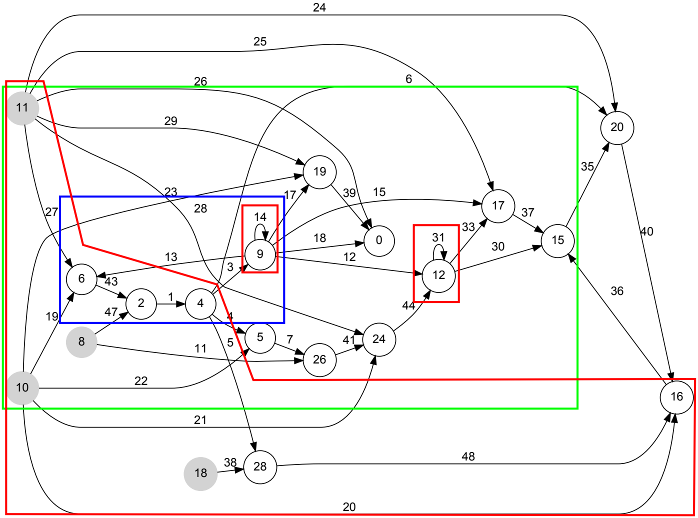
Algorithm Characteristics

- It can solve weighted and unweighted problems. At the moment, its unweighted version is used. The weighted version is investigated in the scope how to take into account information about network properties.
- It consists of relatively simple matrix manipulations being highly parallelizable.
- If n is number of graph vertices, then overall complexity of this method is $O(\epsilon^{-2} n^2 M(n) \log^2 n)$, where ϵ is a parameter relating to approximation quality, $M(n)$ – complexity of an $n \times n$ matrices multiplication.
- It is widely known to be one of the best approximations what is important for successful use of our method.

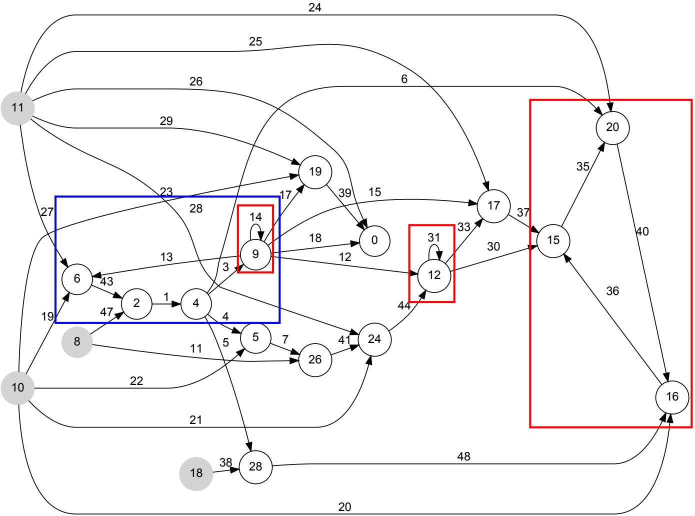
Network Example: 19 vertices



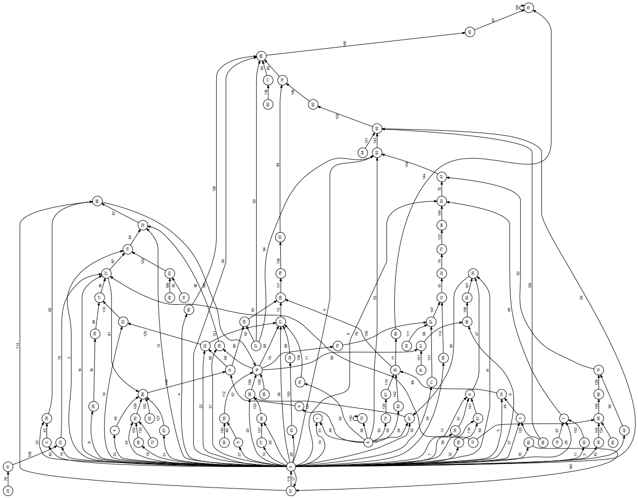
Network Example: 19 vertices, $\epsilon = 1.0$



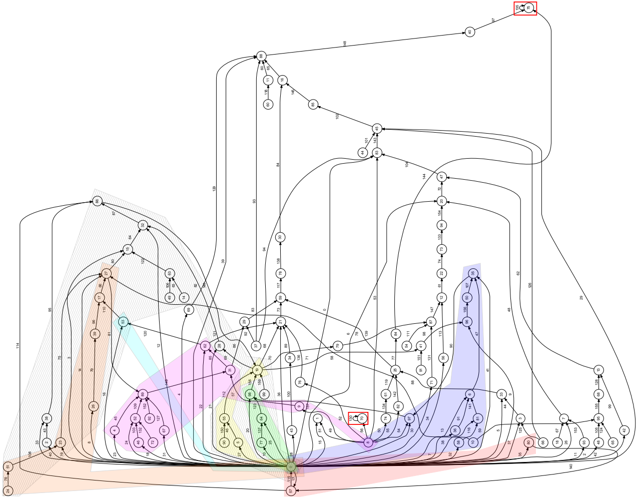
Network Example: 19 vertices, $\epsilon = 0.01$



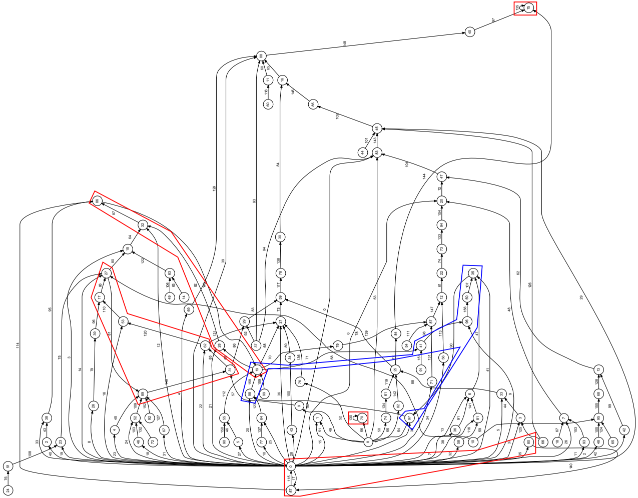
Network Example: 100 vertices



Network Example: 100 vertices, $\epsilon = 1.0$



Network Example: 100 vertices, $\epsilon = 0.01$



Network Statistics Examples

ϵ	#Vert's	#Arcs	#Reg's	Big Reg.	#Vert's per Reg.
0.01	30	45	2	no	4 / 5.50 / 7
0.5	30	45	4	yes	4 / 11.5 / 20
0.01	30	54	4	no	2 / 3.00 / 5
0.5	30	54	6	yes	2 / 7.50 / 25
0.01	100	160	4	no	3 / 5.00 / 7
0.5	100	160	6	yes	2 / 8.67 / 20

Network Statistics Examples: 100 Vertices

ϵ	#Arcs	#Reg's	Big Reg.	#Vert's per Reg.
0.01	144	5	no	3 / 8.40 / 20
0.5	144	7	yes	3 / 16.71 / 56
1	144	7	yes	3 / 16.71 / 56
0.01	153	7	no	2 / 3.86 / 9
0.5	153	10	yes	2 / 12.8 / 41
1	153	13	yes	2 / 10.54 / 28
0.01	179	9	no	3 / 8.78 / 33
0.5	179	12	yes	2 / 10.17 / 51
1	179	13	yes	3 / 10.85 / 32
0.01	185	8	no	2 / 10.50 / 21
0.5	185	10	yes	2 / 10.50 / 23
1	185	11	yes	2 / 12.36 / 23