Algorithms for analyzing and controlling Boolean networks as biological networks

International Workshop on Boolean Networks
Jan. 7-10, 2020

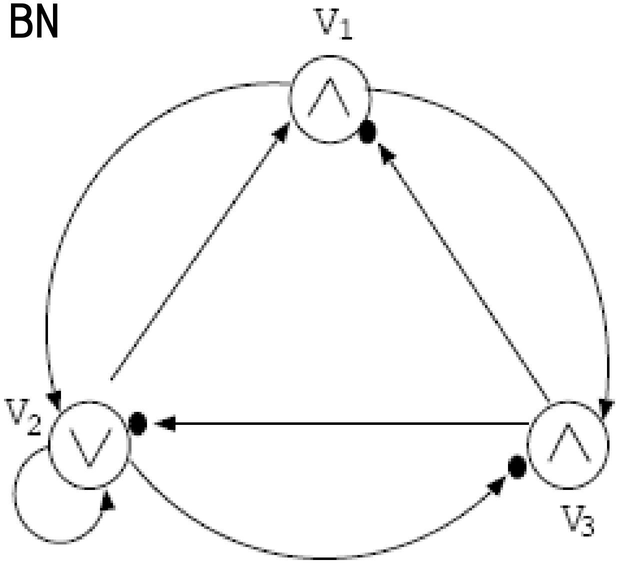
Takeyuki Tamura

Bioinformatics Center,
Institute for Chemical Research
Kyoto University

AND/OR Boolean network (AND/OR BN)

- Mathematical model of genetic network
- Very simple model
  - Each node takes either 0 or 1.
    - Node → gene
    - 1 → active, 0 → inactive
  - States of nodes change synchronously
    - According to regulation rules (= Boolean functions)

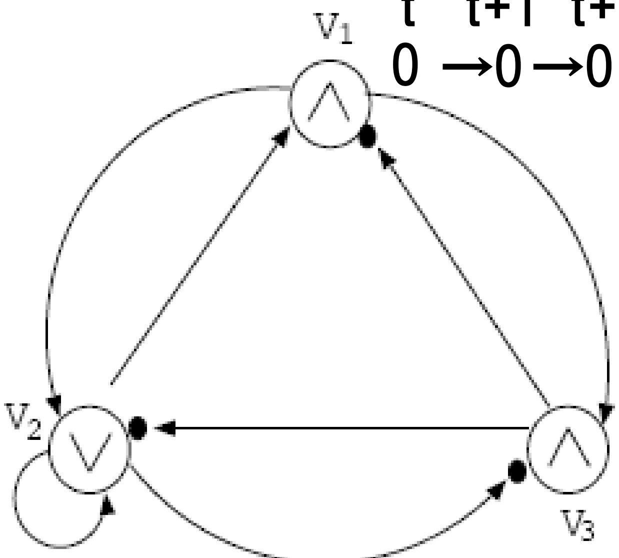
AND/OR BN



AND/OR BN
Regulation rules are limited to disjunction or conjunction of parent nodes.

$$v_1(t+1) = v_2(t) \wedge \overline{v_3(t)}$$
$$v_2(t+1) = v_1(t) \vee v_2(t) \vee \overline{v_3(t)}$$
$$v_3(t+1) = v_1(t) \wedge \overline{v_2(t)}$$

Example of AND/OR BN



t t+1 t+2 t+3
0 →0→0→1

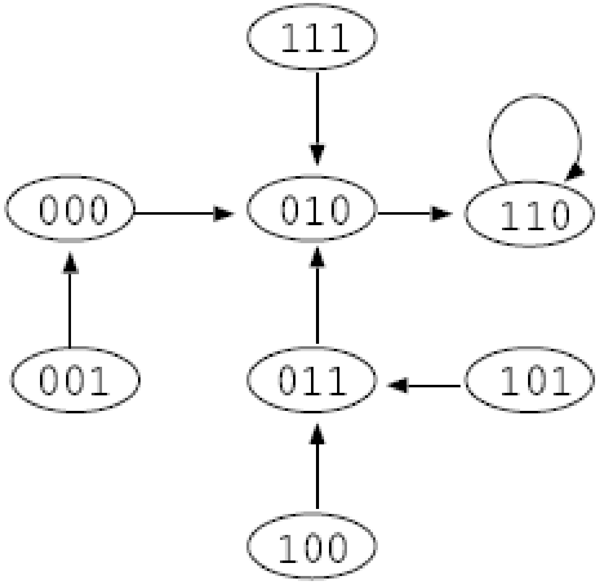
0→0→1→1
t t+1 t+2 t+3

1→0→0→0
t t+1 t+2 t+3

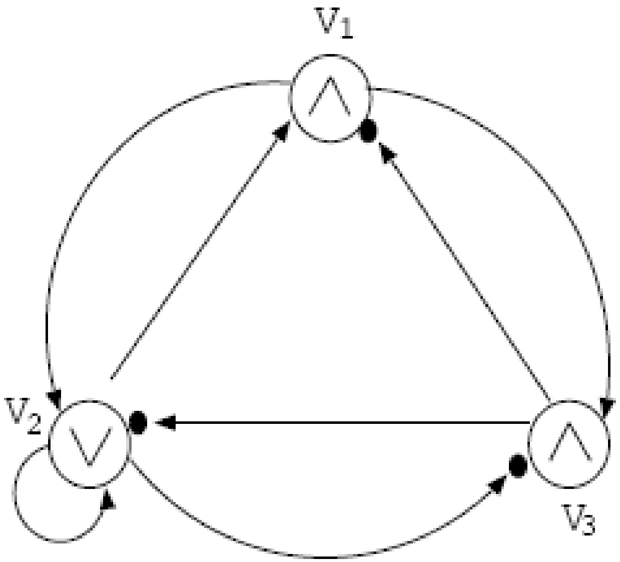
Gene Activity Profile (GAP)

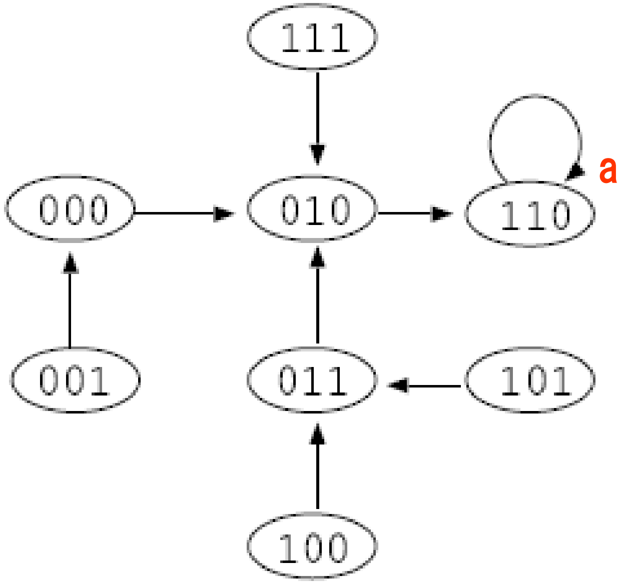
$[v_1, v_2, v_3]$ =[0,0,1]
↓
[0,0,0]
↓
[0,1,0]
↓
[1,1,0]

$$v_1(t+1) = v_2(t) \wedge \overline{v_3(t)}$$
$$v_2(t+1) = v_1(t) \vee v_2(t) \vee \overline{v_3(t)}$$
$$v_3(t+1) = v_1(t) \wedge \overline{v_2(t)}$$



What is a singleton attractor?



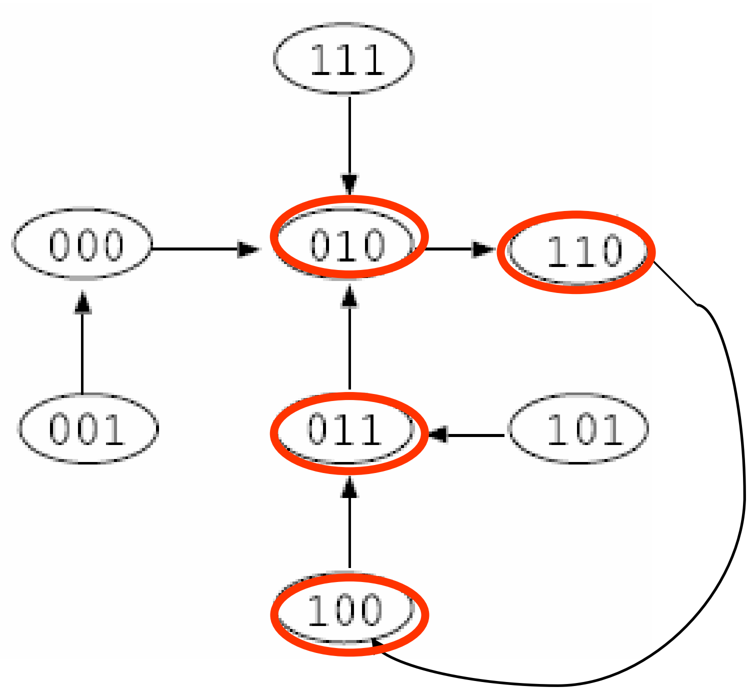


a singleton attractor

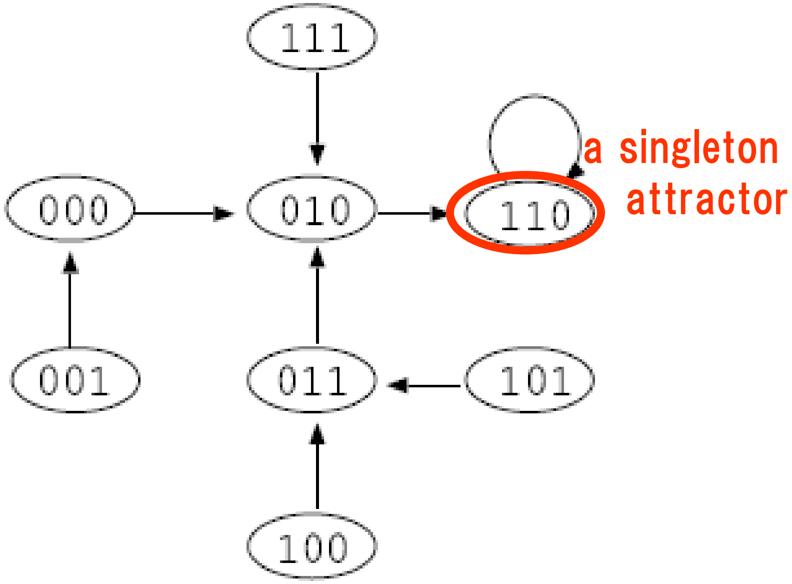
- $[V_1, V_2, V_3]$=[1, 1, 0] → a singleton attractor
  - The state of [1,1,0] never changes.
  - [1,1,0] has a self-loop in the state-transition.

  - One of the most stable states
    - play an important role in biological systems

Cyclic attractor



a cycle of length 4

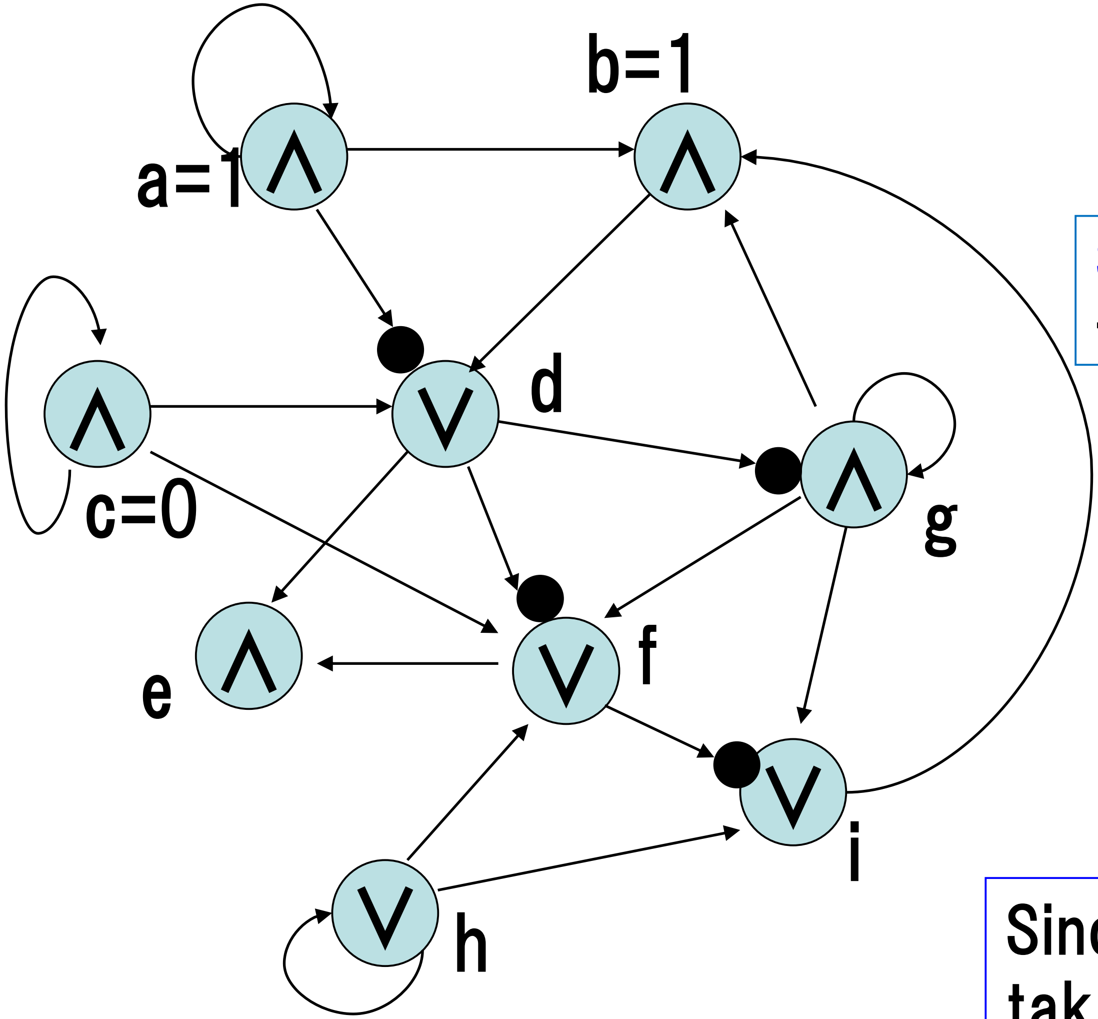


- [0,1,0]→[1,1,0]→[1,0,0]→[0,1,1]
  - An attractor with period 4 (cyclic attractor)

- [1,1,0]
  - An attractor with period 1 (singleton attractor)

- In this talk, we deal with only singleton attractors.

$O(1.787^n)$ time algorithm (Tamura and Akutsu, 2009)



① assign values to all nodes
② consistency checking

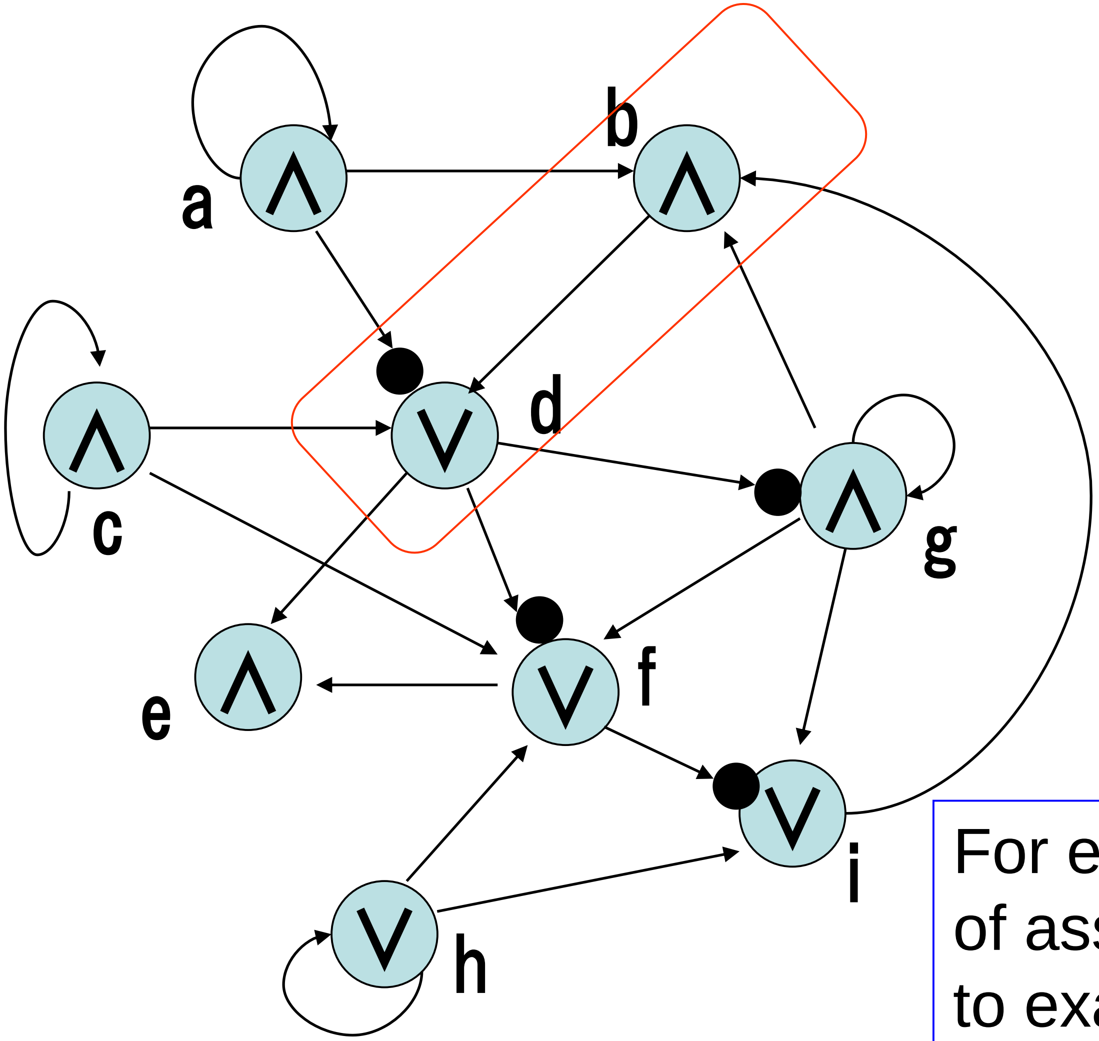
Singleton attractor
→values of nodes never change.

The consistency checking can be done in $O(n^2)$ time.

Consistency checking for node d
-d=1 → OK
-d=0 → contradiction

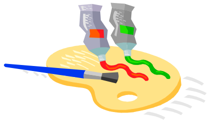
Since the main algorithm takes exponential time, we can ignore the time for consistency checking.

$O(1.787^n)$ time algorithm (Tamura and Akutsu, 2009)



① assign values to all nodes
② consistency checking

If all assignment are examined, it takes $O(2^n)$ time.

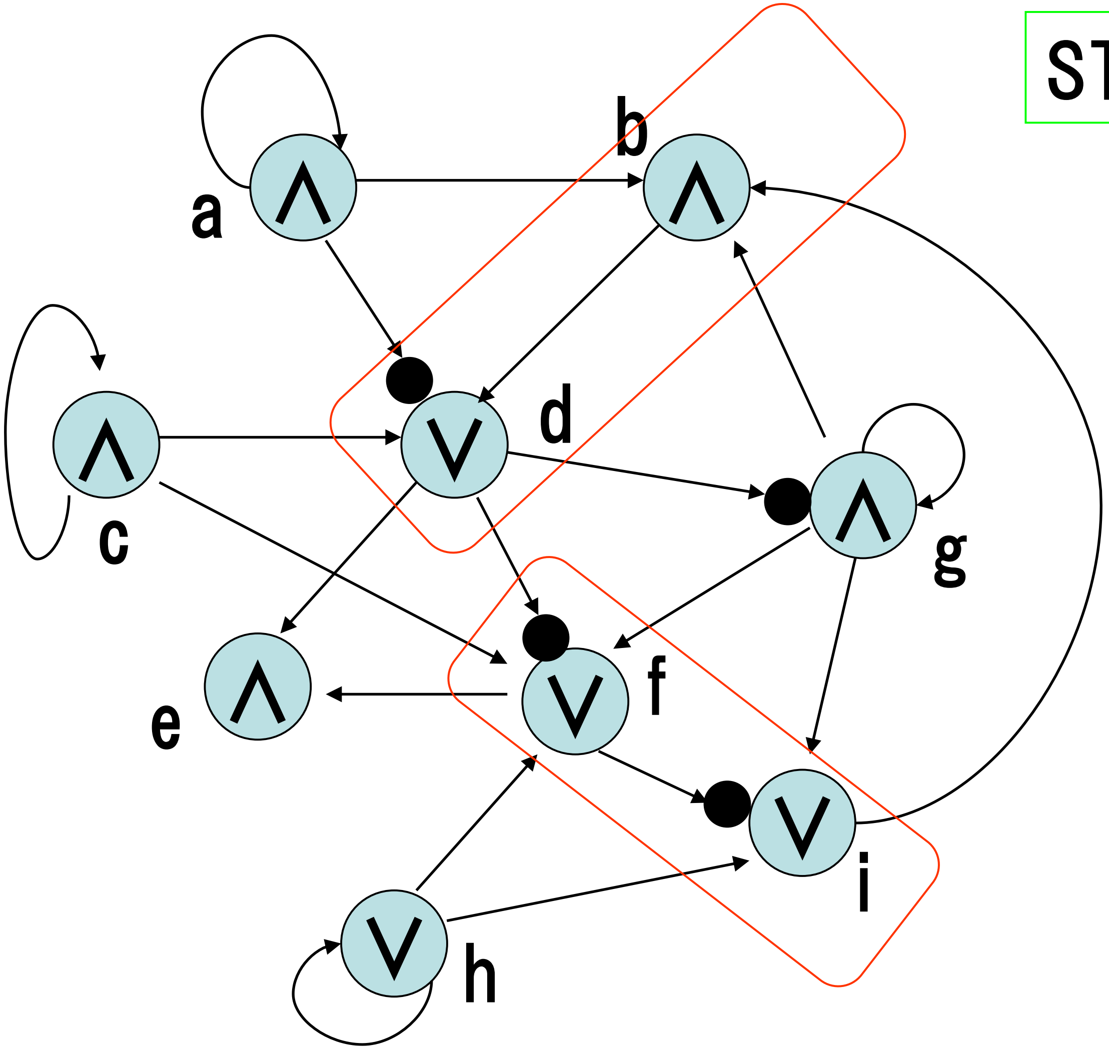


If (b,d)=[1,0], the value of d changes from 0 to 1. It contradicts the condition of a singleton attractor.

For every node pair, the number of assignments which we have to examine is at most 3 of 4 assignments

By using this fact, we can reduce the computational time.

STEP 1 of the proposed algorithm



Initial state:
All nodes are non-assigned

While there exists a non-assigned edge (u,v), examine all possible 3 assignments on (u,v).

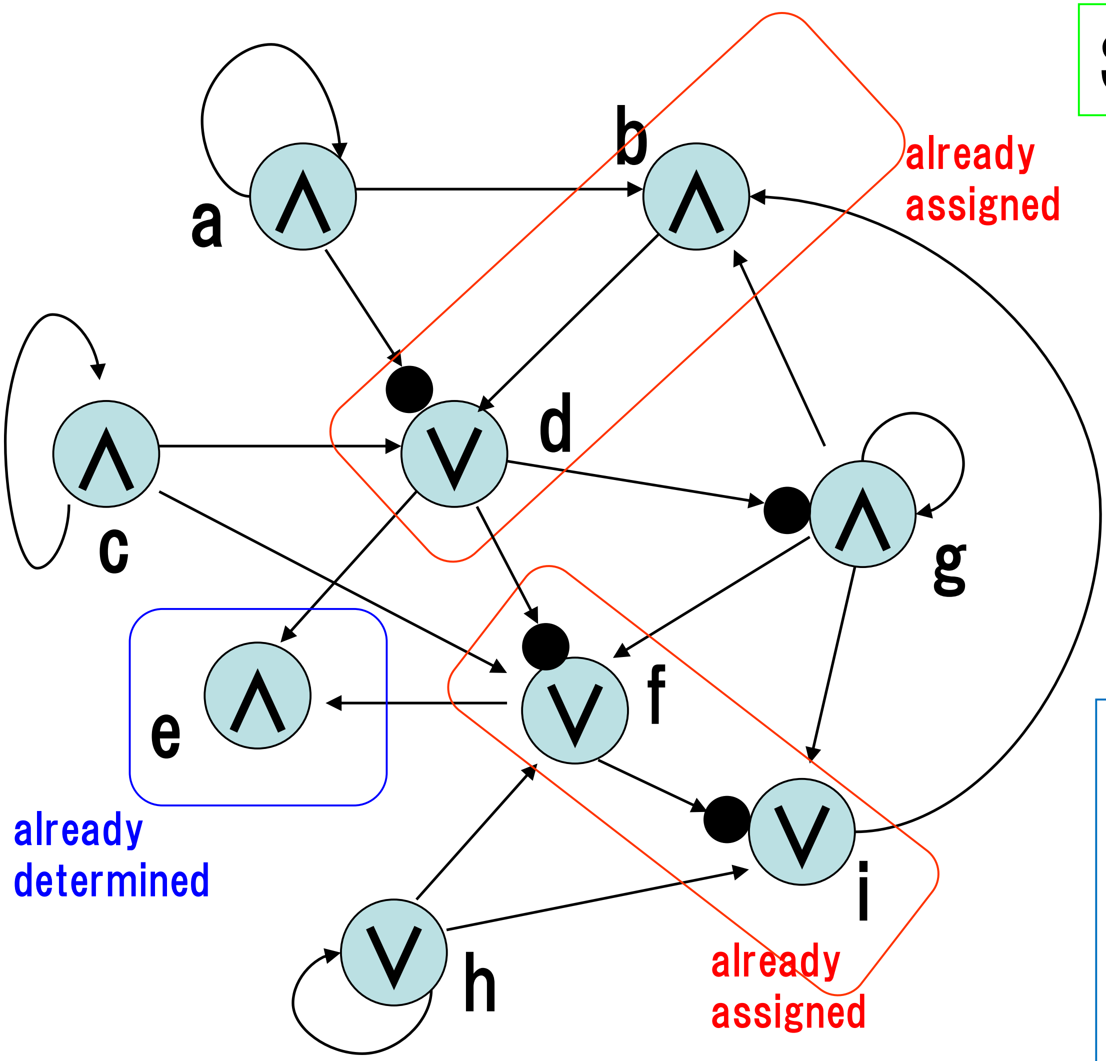
Possible assignments for (b,d) are [0,0], [0,1] and [1,1]. Note that [1,0] is not allowed.

Possible assignments for (f,i) are [0,1], [1,0] and [1,1]. Note that [0,0] is not allowed.

When K nodes are assigned, the number of cases are bounded by $f(K)=3 \cdot f(K-2)$, $f(2)=3$.

Then, $f(K)$ is $O(3^{K/2})$, which is at most $O(1.733^K)$.

STEP 2



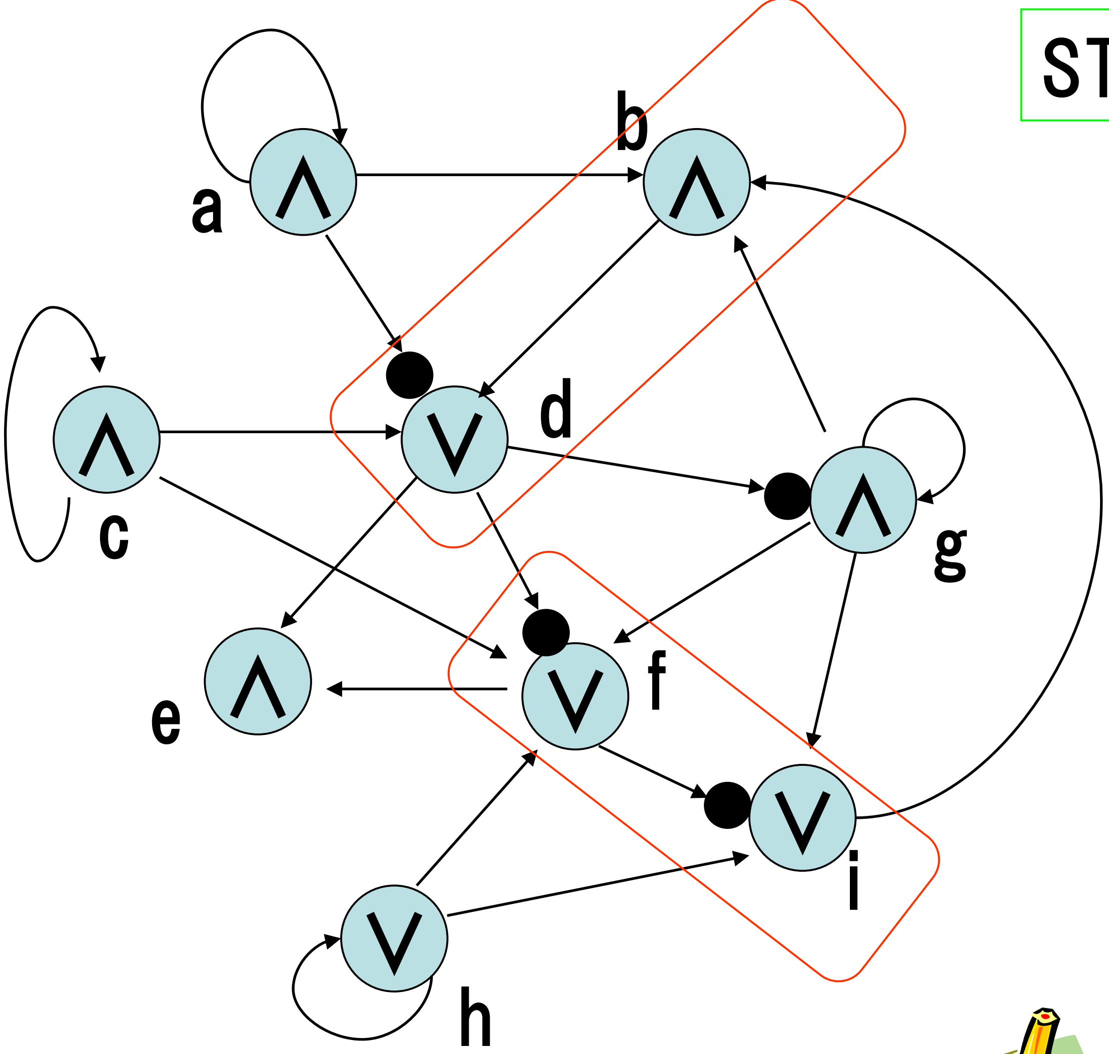
Let W be nodes whose values have not been determined yet.

If $|W| \leqq n - \alpha n$, examine all possible assignments on W

For example,
a,c,g,h ∈W

All 2^4 assignments for a,c,g,h are examined if STEP2 is executed.

If STEP 2 is executed, the computational time is at most $O(2^{n-K} \cdot 1.733^K)$.



STEP 3

If |W|>n-αn,
solve a SAT problem.

If (b,d)=[0,1] is assigned,
$(\bar{a}\vee\bar{g})(\bar{a}\vee c)=1$
must be satisfied.

If (f,i)=[1,1] is assigned,
$(c\vee g\vee h)(g\vee h)=1$
must be satisfied.

When K nodes are assigned, the condition of a singleton attractor can be represented by at most K clauses.

SAT problem with K clauses can be solved in $\tilde{O}(1.234^K)$ time.
where $\tilde{O}(f(m))$ means $O(f(m)poly(m,n))$. (Yamamoto, 2005).

→the overall computational time is bounded by $O(1.234^K \cdot 1.733^K)$.

Theorem

The detection of a singleton attractor can be done in $O(1.792^n)$-time for AND/OR BNs. (worst case)

After STEP1

if |w|≦n-αn,

then STEP 2 is executed.

the computational time is $O(2^{n-K} \cdot 1.733^K)$.

else, STEP 3 is executed.

the computational time is $O(1.234^K \cdot 1.733^K)$.

By setting K=0.767n （α=0.767),

$$2^{n-0.767n} \cdot 1.733^{0.767n} < 1.792^n$$

$$1.234^{0.767n} \cdot 1.733^{0.767n} < 1.792^n$$

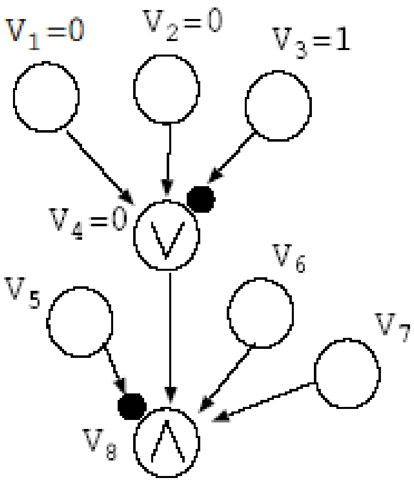
are obtained.

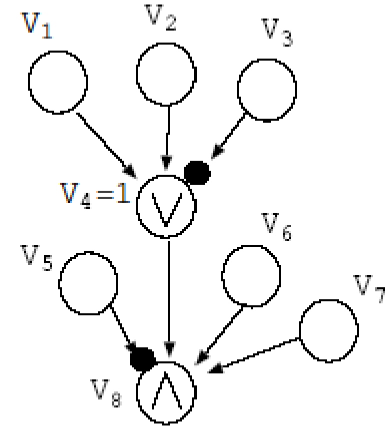
Improved analysis

In the previous analysis,
the number of SAT clauses constructed in STEP 1
is estimated as same as
the number of assigned nodes in STEP 1.

However, there are cases in which SAT clauses are not constructed.

example



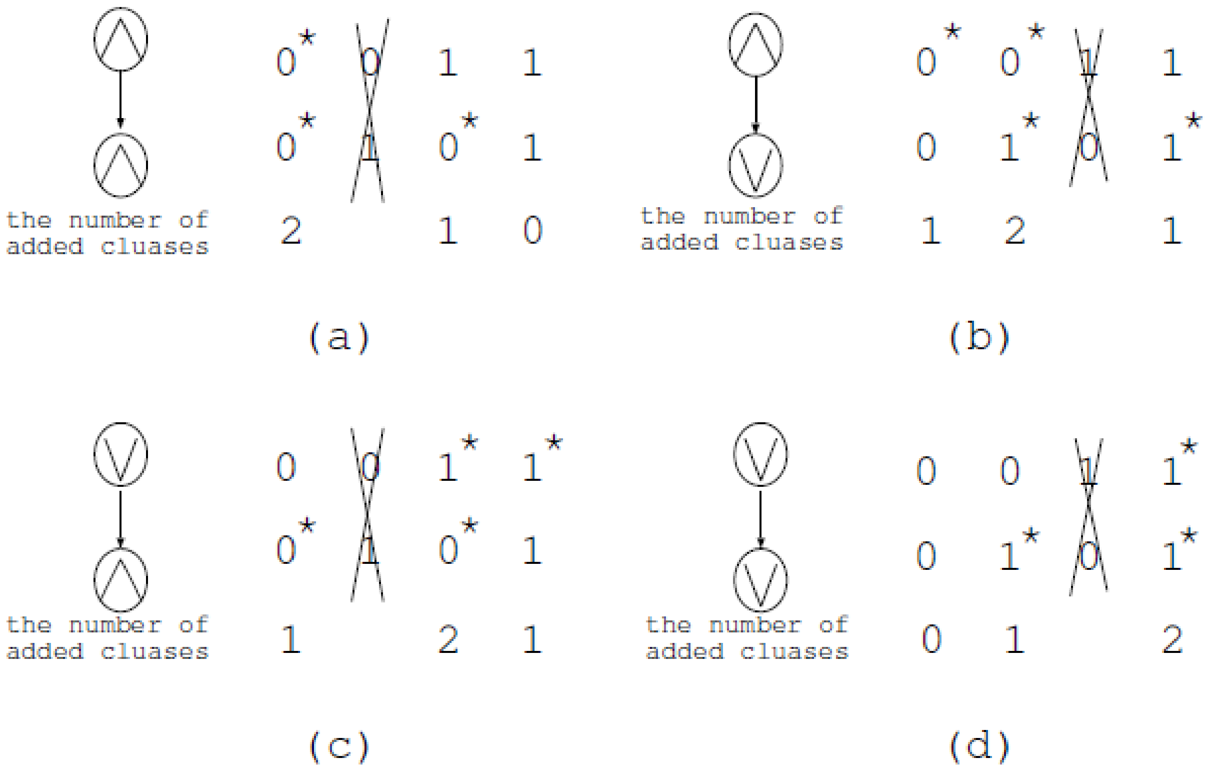


$(V_1 \vee V_2 \vee \overline{V_3})$
is added.

When 0 is assigned to v4,
no SAT clauses are constructed

When 1 is assigned to v4,
a SAT clause is constructed.

Improved analysis



By examining all cases, it can be observed that the worst case for the number of constructed SAT clause is

- One of the three assignments add 2 clauses.
- Two of the three assignments add 1 caluse.

Theorem

Detection of a singleton attractor can be done in $O(1.787^n)$ -time for AND/OR BNs.

After STEP1

if |w|≦n-αn,

then STEP 2 is executed.

the computational time is $O(2^{n-K} \cdot 1.733^K)$.

else, STEP 3 is executed.

the computational time is $O(2.089^K)$.

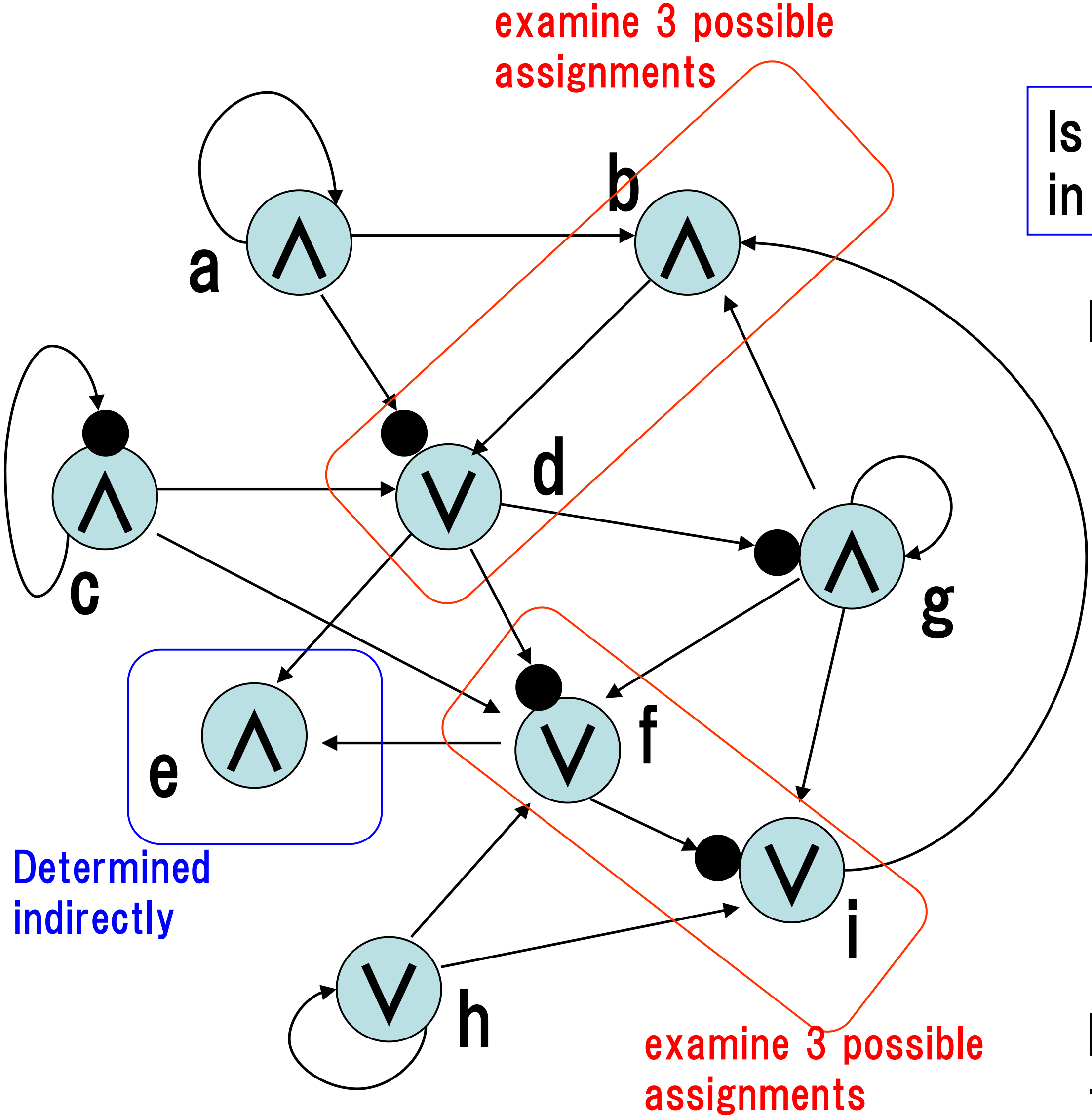
By setting K=0.7877n （α=0.7877),

$$2^{n-0.7877n} \cdot 1.733^{0.7877n} < 1.7866^n$$

$$2.089^{0.7877n} < 1.7866^n$$

are obtained.

$O(1.787^n)$ time algorithm (Tamura and Akutsu,2009)



examine 3 possible assignments

Determined indirectly

examine 3 possible assignments

The consistency checking can be done in polynomial time.

Is there a singleton attractor in a given Boolean network?

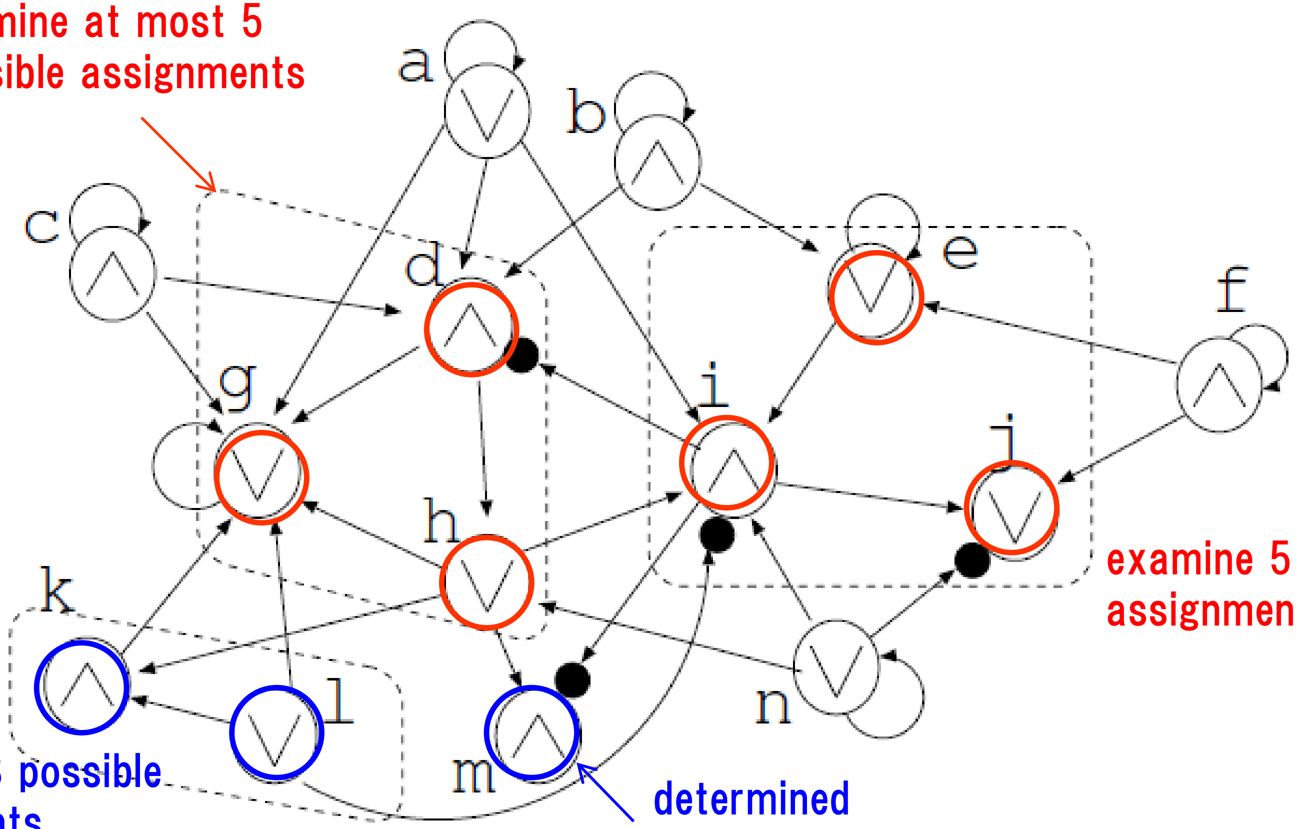
If all assignment are examined, it takes $O(2^n)$ time.

The consistency checking can be done in polynomial time.

If (b,d)=[1,0], the value of d changes from 0 to 1. It contradicts the condition of a singleton attractor.

By using this fact, we reduced the computational time in the previous algorithm.

More improved algorithm



examine at most 5 possible assignments

examine 5 possible assignments

examine 3 possible assignments

determined indirectly

While there exist non-assigned neighboring edges, examine all possible assignment, which are at most 5.

For example, possible assignments for (e,i,j) are
[0,0,0],[0,0,1],[1,0,0],[1,0,1],[1,1,1]
since [0,1,0],[0,1,1],[1,1,0] are impossible assignments.

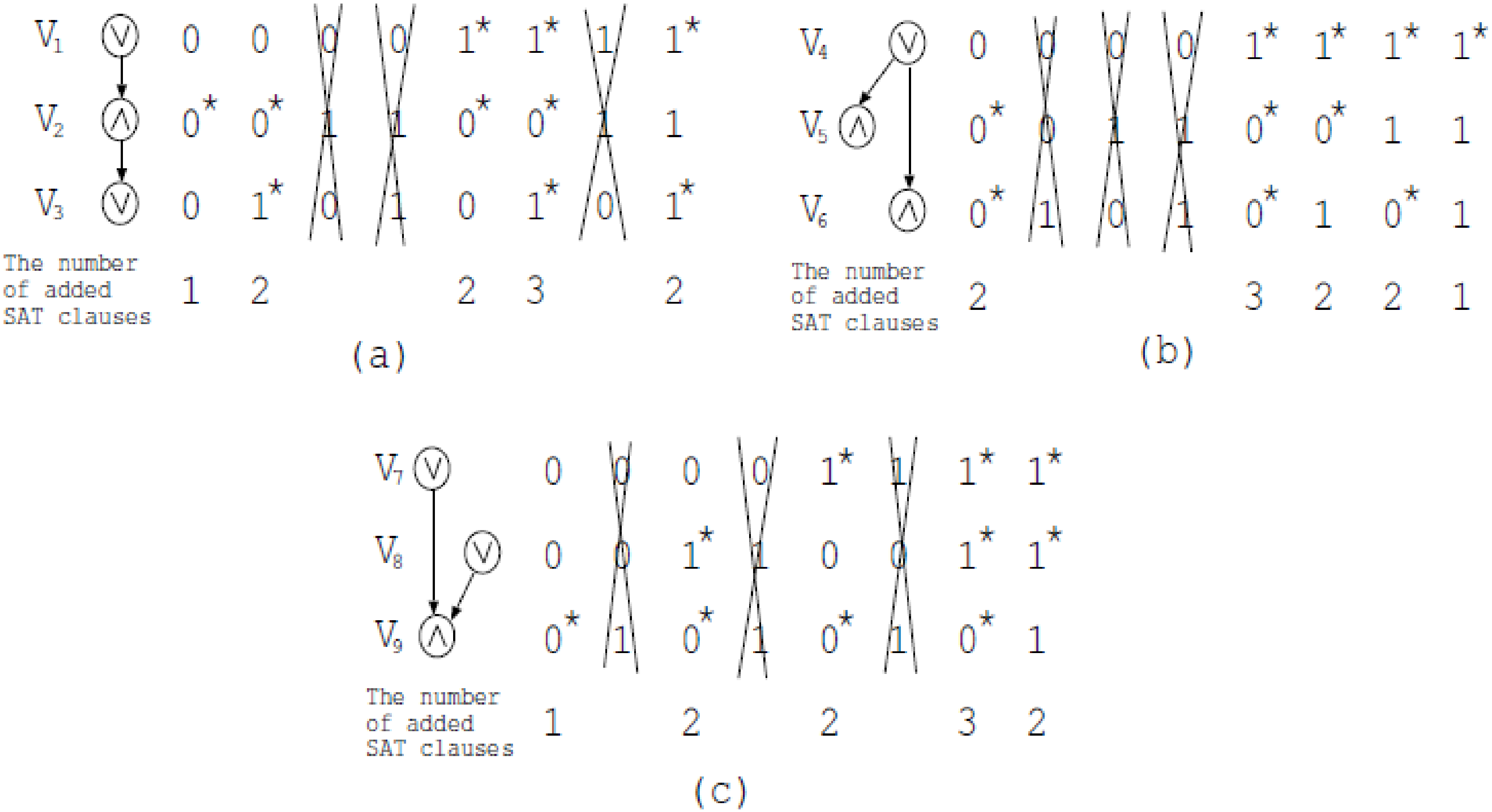
Theorem

The detection of a singleton attractor can be done in $O(1.774^n)$ -time for AND/OR BNs.

```
After STEP1
    if K>0.767(n-L),
        then STEP 3 is executed.

        the computational time is $\underline{O(2^{n-K-L} \cdot 1.71^K \cdot 1.733^L)}$

    else if STEP 4 is executed.

        the computational time is $\underline{O(1.234^K \cdot 1.71^K \cdot 1.733^L)}$
```

Improved analysis

There are cases where SAT clauses are not constructed.



V_1	∨	0	0	~~0~~	~~0~~	1^*	1^*	~~1~~	1^*
V_2	∧	0^*	0^*	~~1~~	~~1~~	0^*	0^*	~~1~~	1
V_3	∨	0	1^*	~~0~~	~~1~~	0	1^*	~~0~~	1^*
The number of added SAT clauses		1	2			2	3		2

(a)

V_4	∨	0	~~0~~	~~0~~	~~0~~	1^*	1^*	1^*	1^*
V_5	∧	0^*	~~0~~	~~1~~	~~1~~	0^*	0^*	1	1
V_6	∧	0^*	~~1~~	~~0~~	~~1~~	0^*	1	0^*	1
The number of added SAT clauses		2				3	2	2	1

(b)

V_7	∨	0	~~0~~	0	~~0~~	1^*	~~1~~	1^*	1^*
V_8	∨	0	~~0~~	1^*	~~1~~	0	~~0~~	1^*	1^*
V_9	∧	0^*	~~1~~	0^*	~~1~~	0^*	~~1~~	0^*	1
The number of added SAT clauses		1		2		2		3	2

(c)

The worst case is as follows:

(1) One of the five assignments adds one clause.
(2) Three of the five assignments add two clauses.
(3) One of the five assignments adds three clauses.

Theorem

The detection of a singleton attractor can be done in $O(1.757^n)$ -time for AND/OR BNs.

After STEP1

if K>0.8286(n-L),

then STEP 3 is executed.

the computational time is $O(2^{n-K-L} \cdot 1.71^K \cdot 1.733^L)$

else if STEP 4 is executed.

the computational time is $O(1.234^K \cdot 1.71^K \cdot 1.733^L)$

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Determining a singleton attractor of an AND/OR Boolean network in $O(1.587^n)$ time

Avraham A. Melkman[a], Takeyuki Tamura[b,*], Tatsuya Akutsu[b]

[a] *Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel 84105*
[b] *Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto, Japan 611-0011*

ARTICLE INFO

Article history:
Received 31 January 2010
Received in revised form 16 April 2010
Accepted 3 May 2010
Available online 8 May 2010
Communicated by M. Yamashita

Keywords:
Algorithms
SAT
Boolean network
Singleton attractor

Integer linear programming–based methods for controlling Boolean metabolic networks

Takeyuki Tamura

Bioinformatics Center,
Institute for Chemical Research
Kyoto University

Models of metabolic networks

- Mathematical model
  - Ordinary differential equation (ODE) model

    high explanatory power, but needs many parameters, often used for small models
  - Flux balance analysis (FBA) model

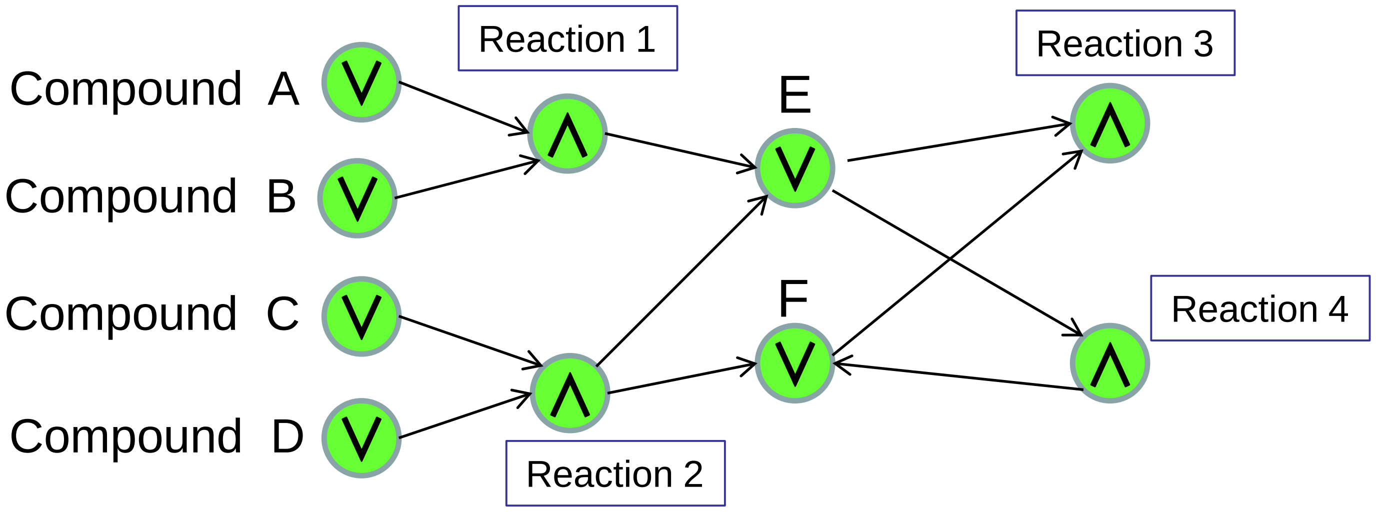
    assumes a steady state, often used for genome-scale model, good for optimizing production of biomass
  - Elementary mode (EM) model,

    less explanatory power, good for checking the producibility of biomass
  - Boolean model

    Every node is assigned either 0 or 1.

    Simple model, but good for logical analysis

Metabolic network on Boolean model

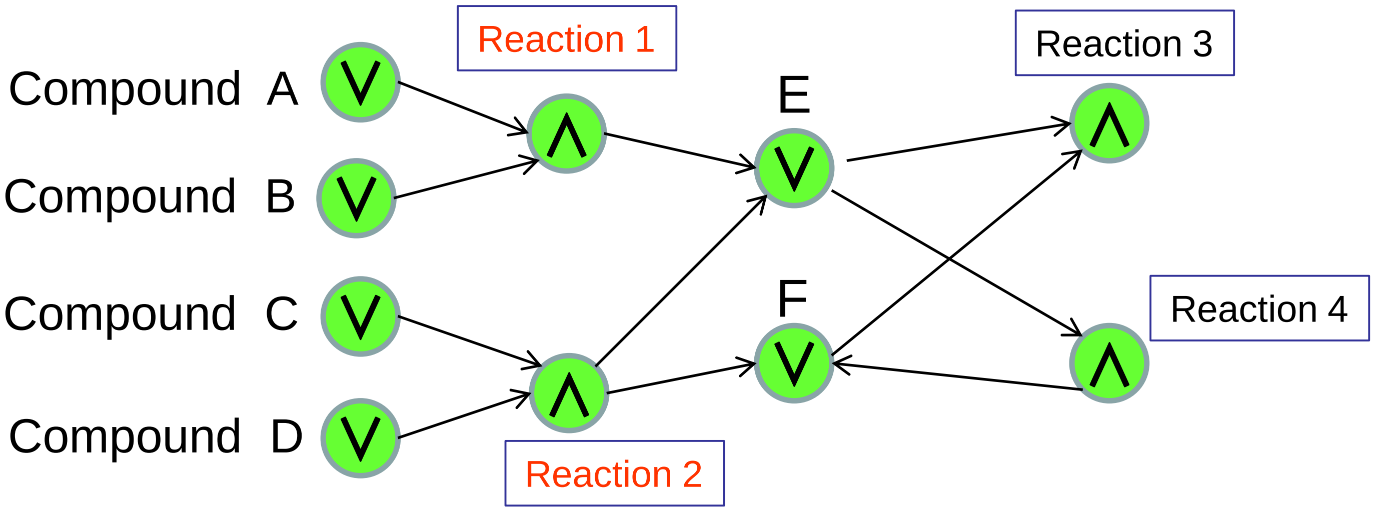


- <span>Every node is assigned either 0 or 1.</span>
- For reactions,

  1: can takes place, 0: cannot take place.
- For compounds,

  1: producible, exist, 0: not producible, not exist

Metabolic network on Boolean model



- For Reaction 1, Compounds A and B are necessary.

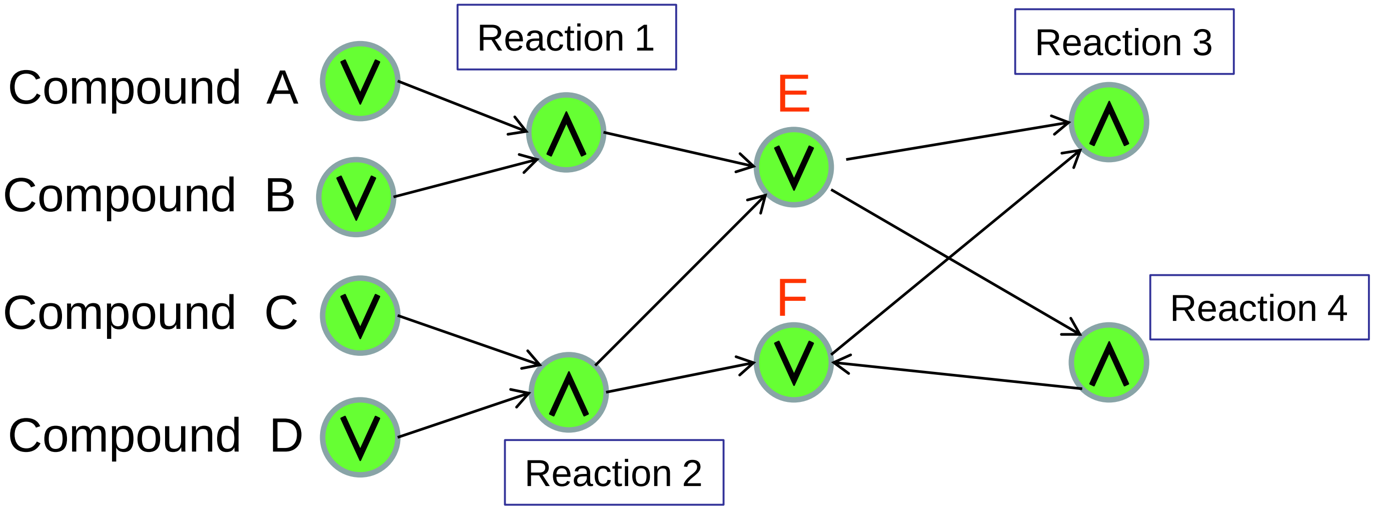
  → $R1 = A \wedge B$

- For Reaction 2, Compounds C and D are necessary.

  → $R2 = C \wedge D$

- Reactions can be represented by "AND" nodes.

Metabolic network on Boolean model



- Compound E is producible if Reaction 1 or 2 occurs.

  → E = R1 ∨ R2

- Compound F is producible if Reaction 2 or 4 occurs.

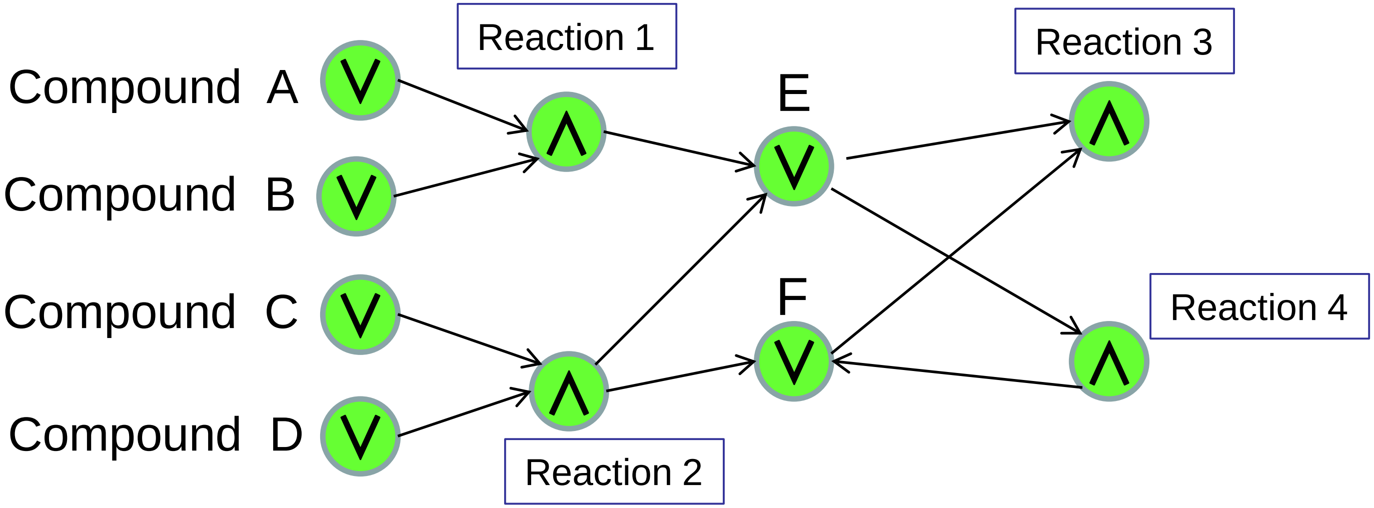
  → F = R2 ∨ R4

- Compounds can be represented by "OR" nodes.

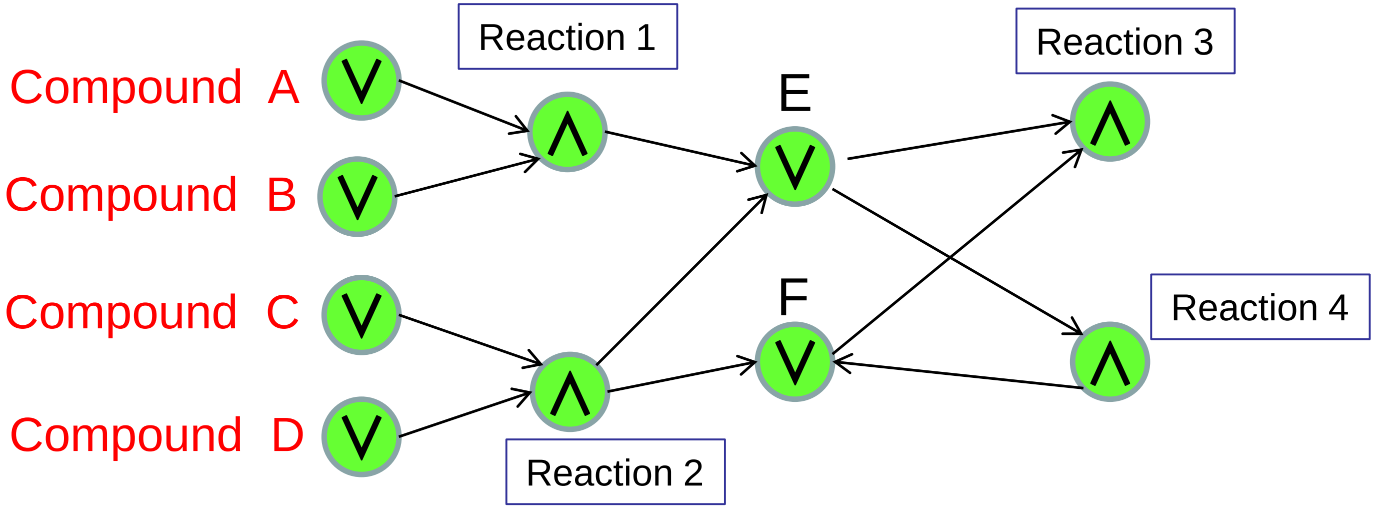
Metabolic network on Boolean model

- Thus, a metabolic network can be represented by a directed graph in which each node is labeled by either "AND(∧) (Reaction)" or "OR(∨) (Compound)".

  - All adjacent nodes of "AND" nodes are "OR" nodes
  - All adjacent nodes of "OR" nodes are "AND" nodes.
  - "Negation"s do not exist.

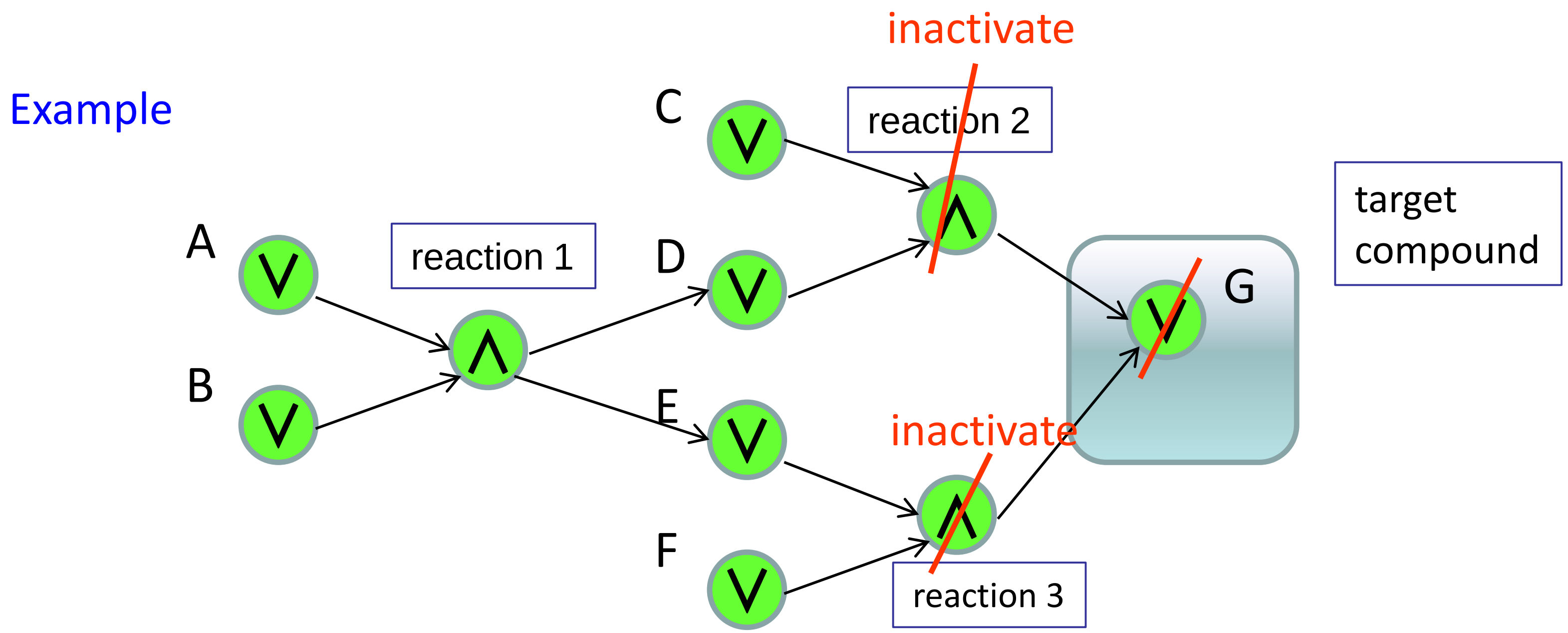


Metabolic network on Boolean model



- Nodes with indegree 0 are called source nodes
- Source nodes are always assigned 1, assuming that they are provided by external environment.

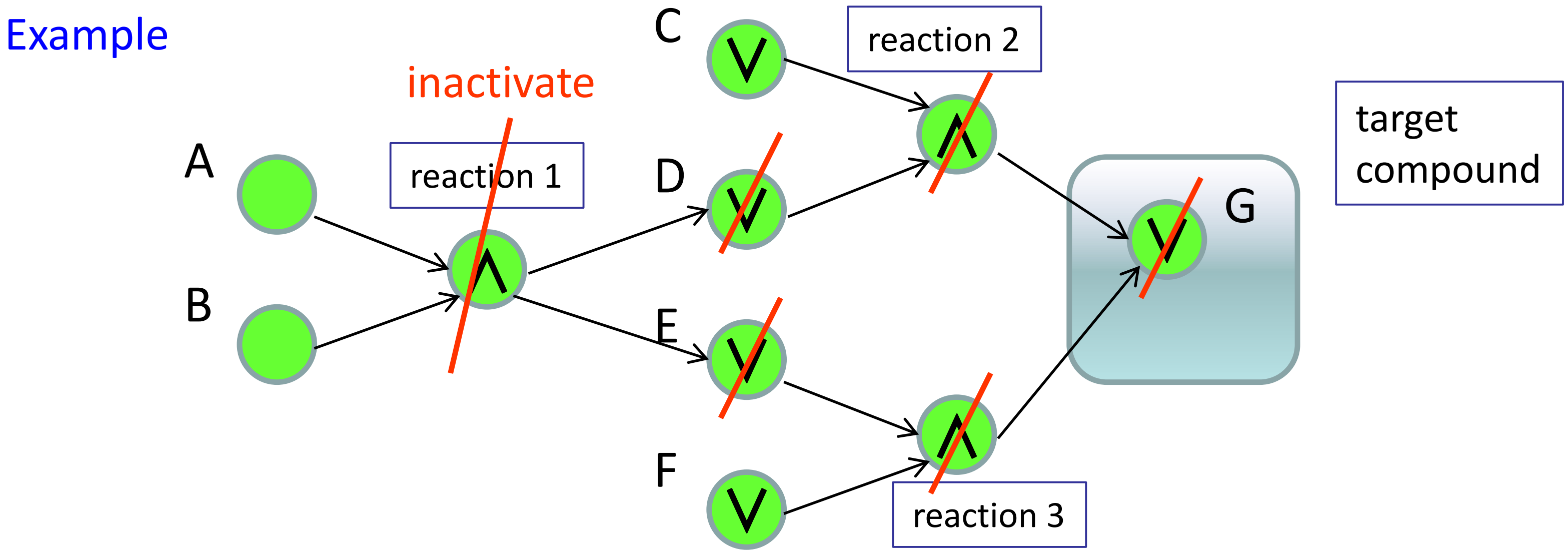
Boolean Reaction Cut Problem



- Which reactions should be deleted so that the target compound becomes unproducible?

（A,B,C, and F are always assigned 1.)

Minimum Boolean Reaction Cut Problem



- Which reactions should be deleted so that the target compound becomes unproducible?
  - Reaction 2 and 3: two reaction deletion
  - Reaction 1: one reaction deletion

Minimum reaction cut problem

- Detection of such a reaction cut has potential application to drug design.

- This problem is known to be very complex, (NP-complete). (Tamura et al. 2010)

- For this problem, we developed an integer linear programming (ILP)-based method which can handle large scale networks.

Linear Programming (LP)

Example

$$\begin{array}{ll} \text{maximize} & 2x + 5y - 3z \\ \text{subject to} & 3a - x < 2b + 4y \\ & y + 2z = 3x + c \\ & 2x + 5c > 3a \end{array}$$

- An objective function and constraints must be represented by linear function of variables.

- Linear Programming is efficiently solvable.

(Mixed) Integer Linear Programming (ILP, MILP)

Example

$$\begin{aligned} \text{maximize} \quad & 2x + 5y - 3z \\ \text{subject to} \quad & 3a - x < 2b + 4y \\ & y + 2z = 3x + c \\ & 2x + 5c > 3a \end{aligned}$$

x,y,z are integers.

- An objective function and constraints must be represented by linear function of variables.
- Solving ILP or MILP is NP-complete problem.
- CPLEX is an efficient ILP solver.

Linear representation of Boolean AND

$$x_1 = x_2 \wedge x_3 \wedge \ldots \wedge x_k$$

Not applicable

$$\Longleftrightarrow \quad (x_1 \vee \overline{x_2} \vee \ldots \vee \overline{x_k}) \wedge (\overline{x_1} \vee x_2) \wedge \ldots \wedge (\overline{x_1} \vee x_k) = 1$$

Not applicable

$$\Longleftrightarrow \quad \boxed{\begin{array}{r} x_1 + (1 - x_2) + \ldots + (1 - x_k) \geq 1 \\ (1 - x_1) + x_2 \geq 1 \\ \vdots \\ (1 - x_1) + x_k \geq 1 \end{array}}$$

$$x_1, x_2, \ldots, x_k \in \{0,1\}$$

Linear inequalities

Applicable !

To represent BN related problems by ILP

Representing Boolean "AND" by linear constraints

$$y = x_1 \land x_2 \land \ldots \land x_k$$

$$y \geq (x_1 + x_2 + \ldots + x_k) - (k - 1)$$

$$y \leq \frac{1}{k}(x_1 + x_2 + \ldots + x_k)$$

x and y are binary

If all x are 1,

y ≧ 1 and y ≦ 1 must be satisfied → y=1

If some x is 0,

y ≧ 0 and y ≦ 0.zzz must be satisfied → y=0

Linear representation of Boolean OR

$$x_1 = x_2 \vee x_3 \vee \ldots \vee x_k$$

Not applicable

$$\Longleftrightarrow \quad (\overline{x_1} \vee x_2 \vee \ldots \vee x_k) \wedge (x_1 \vee \overline{x_2}) \wedge \ldots \wedge (x_1 \vee \overline{x_k}) = 1$$

Not applicable

$$\Longleftrightarrow \quad \boxed{\begin{array}{r} (1-x_1)+x_2+\ldots+x_k \geq 1 \\ x_1+(1-x_2) \geq 1 \\ \vdots \\ x_1+(1-x_k) \geq 1 \end{array}}$$

Linear inequalities

$$x_1, x_2, \ldots, x_k \in \{0,1\}$$

Applicable !

To represent BN related problems by ILP…

Representing Boolean “OR” by linear constraints

$$y = x_1 \lor x_2 \lor \ldots \lor x_k$$

$$y \leq x_1 + x_2 + \ldots + x_k$$

$$y \geq \frac{1}{k}(x_1 + x_2 + \ldots + x_k)$$

x and y are binary

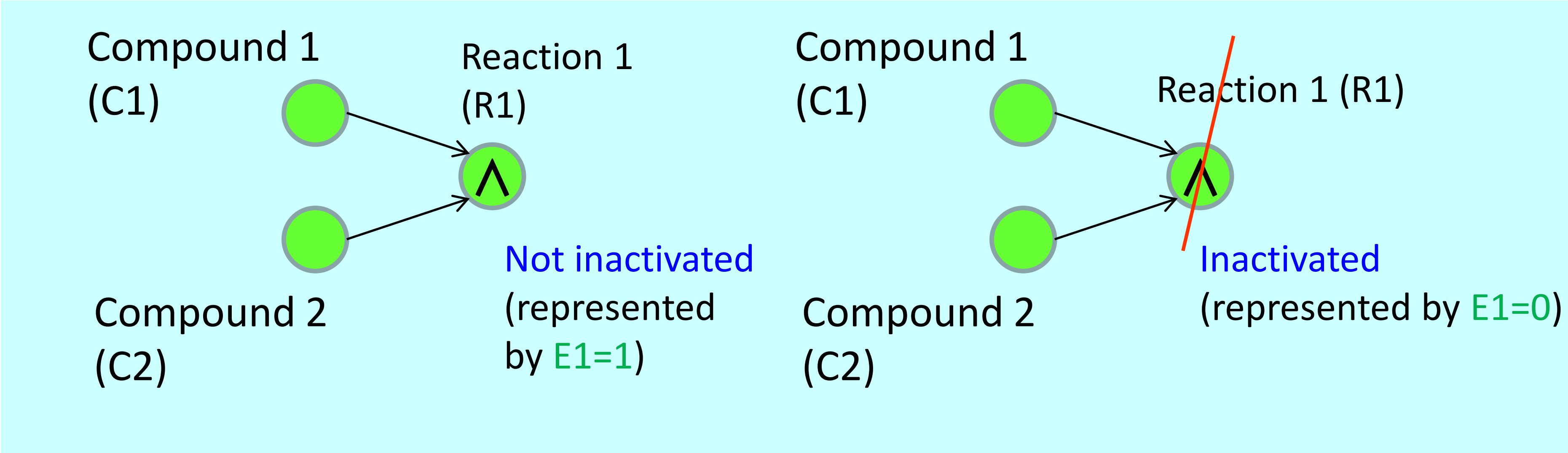
If all x are 0,

$y \leqq 0$ and $y \geqq 0$ must be satisfied → y=0

If some x is 1,

$y \leqq 1$ and $y \geqq 0.zzz$ must be satisfied → y=1

ILP for Minimum Boolean Cut



Compound 1 (C1)

Compound 2 (C2)

Reaction 1 (R1)

Not inactivated (represented by E1=1)

Compound 1 (C1)

Compound 2 (C2)

Reaction 1 (R1)

Inactivated (represented by E1=0)

The reaction R1 can be represented by

$R1 = C1 \wedge C2 \wedge E1,$

and it is transformed into

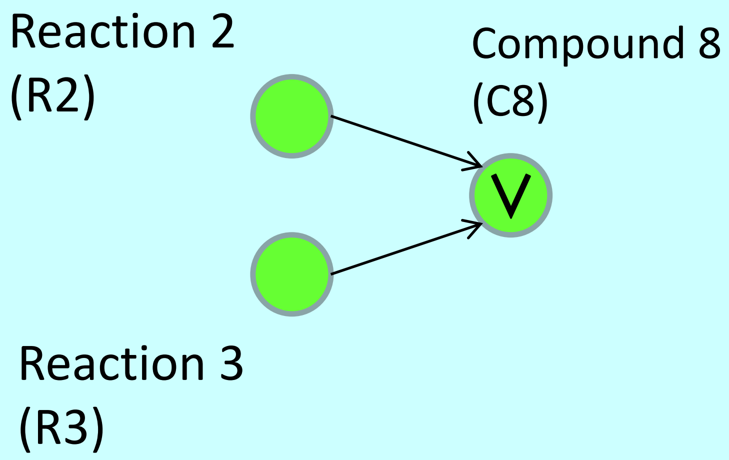
$$R1 + (1-C1) + (1-C2) + (1-E1) \geqq 1$$

$$(1-R1) + C1 \geqq 1$$

$$(1-R1) + C2 \geqq 1$$

$$(1-R1) + E1 \geqq 1.$$

ILP for Minimum Boolean Cut



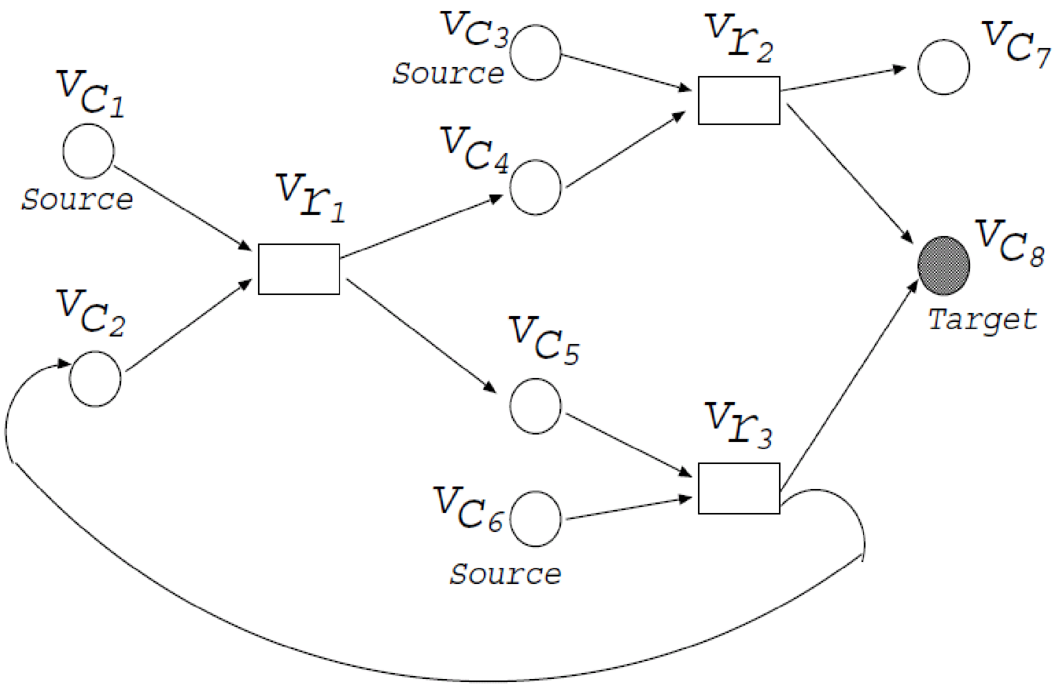
The compound C8 can be represented by

C8 = R2 ⋁ R3,

and it is transformed into

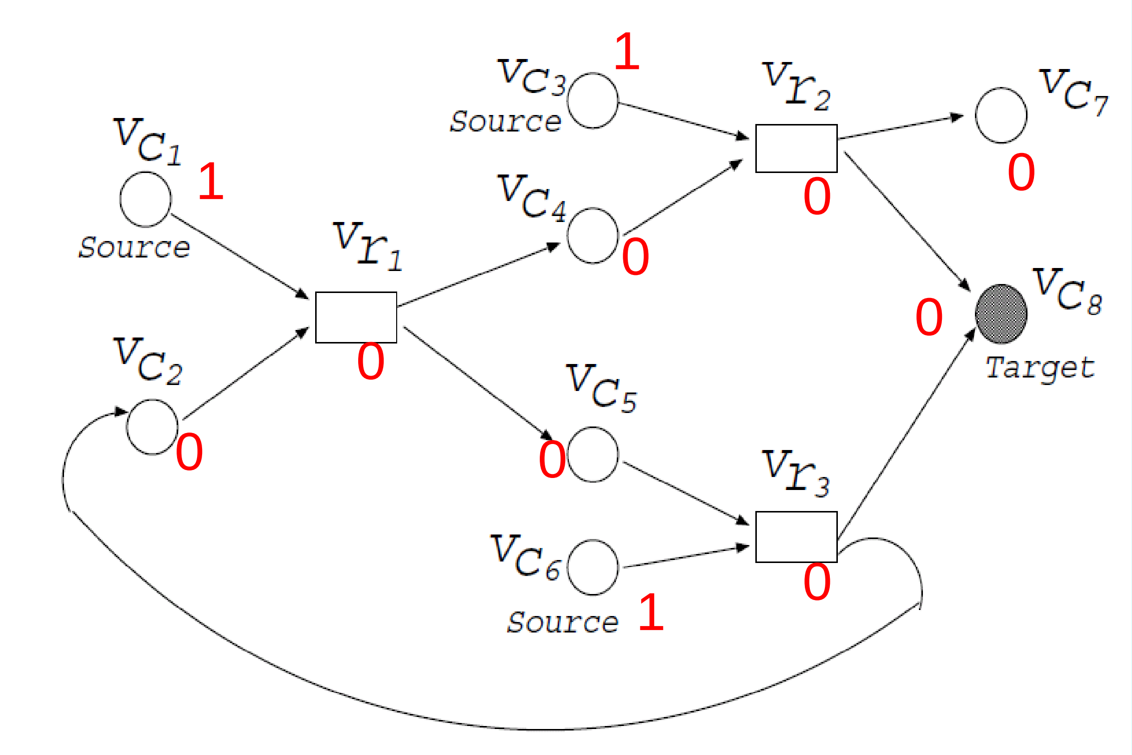
$$(1\text{-}C8) + R2 + R3 \geqq 1$$
$$C8 + (1\text{-}R2) \geqq 1$$
$$C8 + (1\text{-}R3) \geqq 1.$$

Example for solving Boolean reaction by ILP



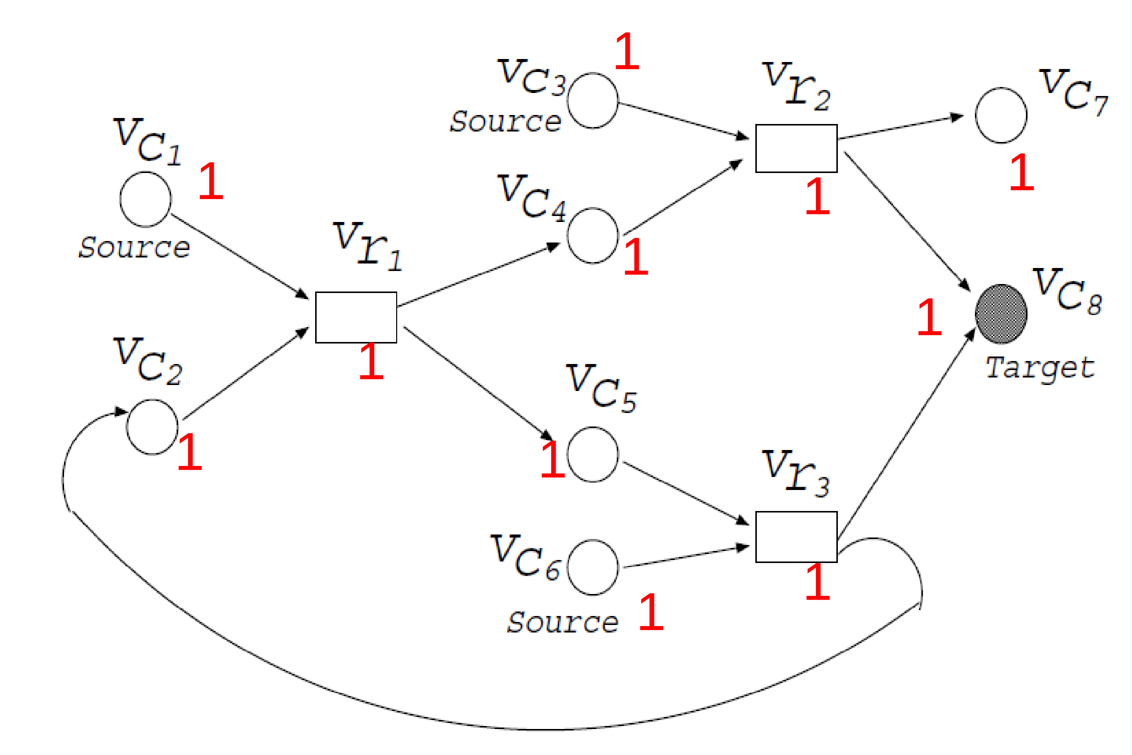
- To minimize the number of deleted reactions, the objective function is “Maximize E1 + E2 + E3.”
- The necessary constrains are
  - C8 = 0
  - The linear constraints for C2,C4,C5,C7,R1,R2,R3.
  - C1=C3=C8=1

Inappropriate solution due to directed cycle



– If 0 is assigned to every node included in the directed cycle, the target compound becomes non-producible even when no reaction is deleted.

Solutions depend on initial states



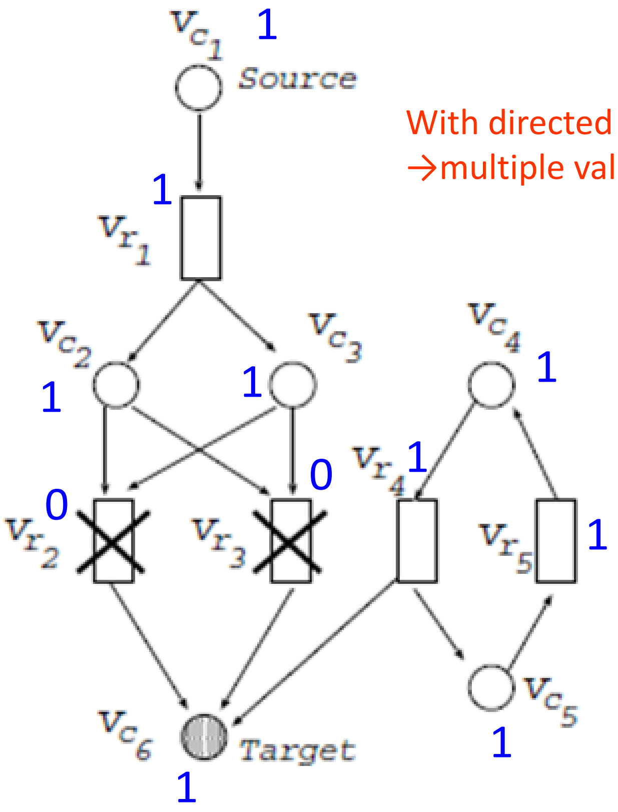
- We assume that 1 is assigned to every node in the initial state.
- Maximal valid assignment corresponds to the solution for this problem setting.

Source node
→ indegree = 0

Maximal valid assignment

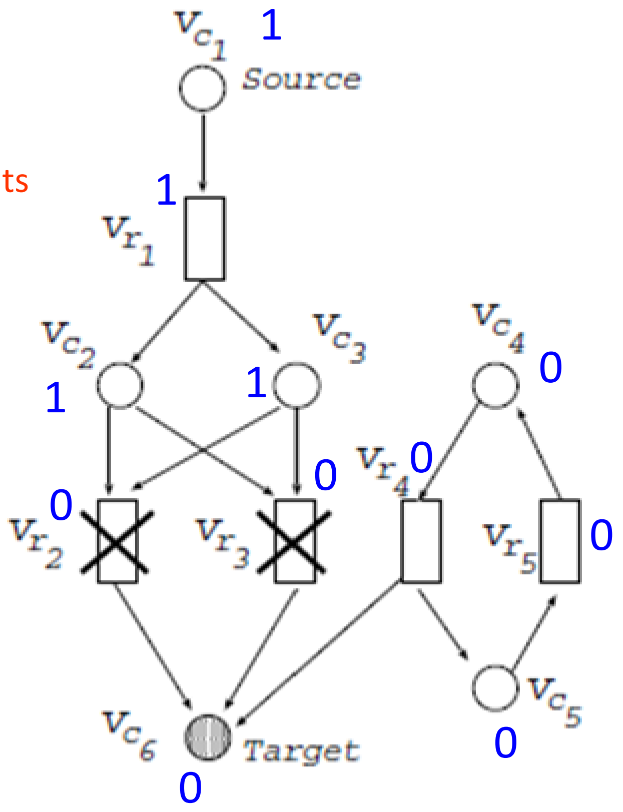
▭ reaction

○ compound



With directed cycle
→multiple valid assignments

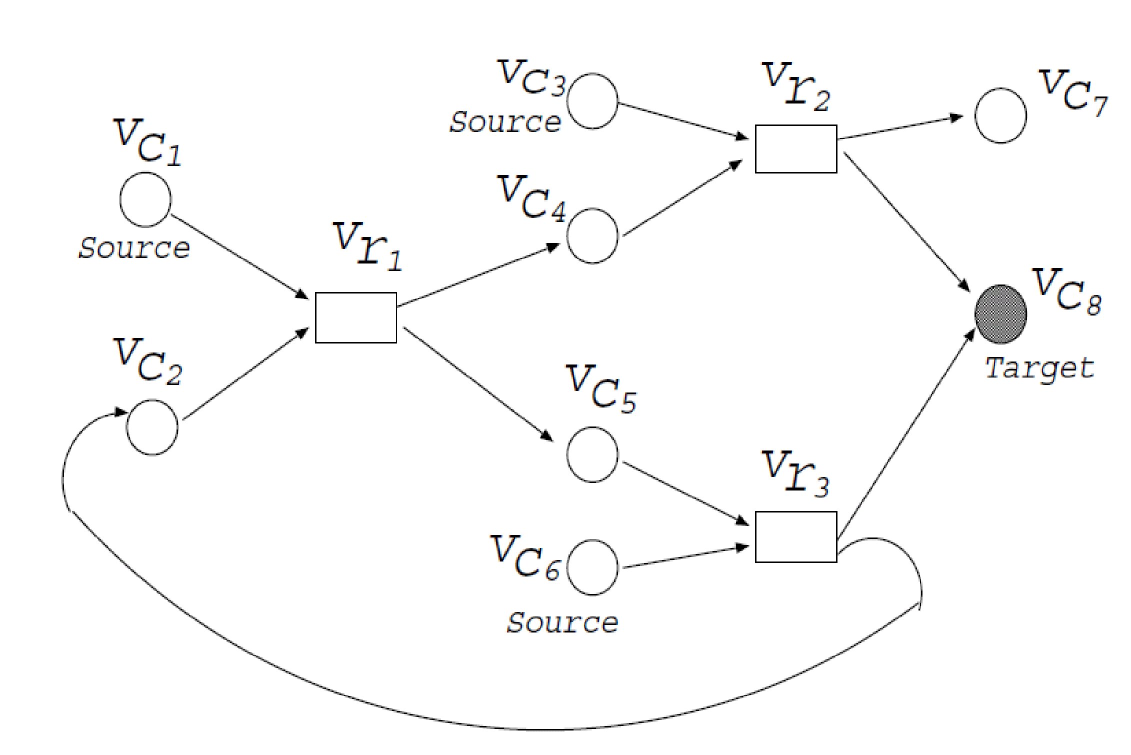
maximal valid assignment



Valid assignment

If the number of 1s is maximal in a valid assignment,
it is called a maximal valid assignment.

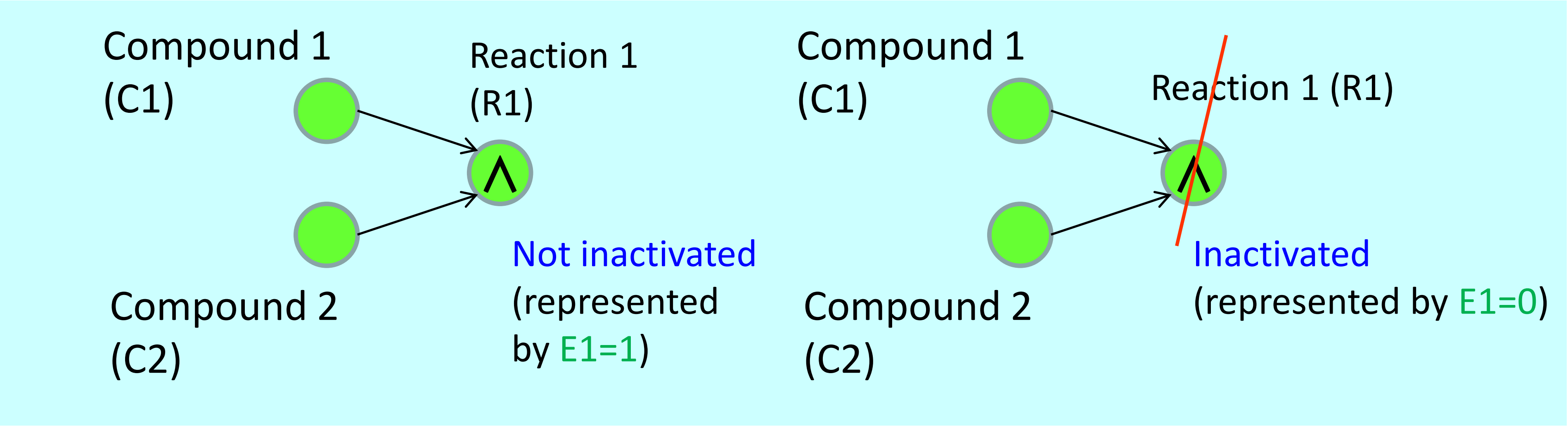
Notion of time



- For example, the constraints for C8 can be represented as

$$\{1-C8(t+1)\} + R2(t) + R3(t) \geqq 1$$
$$C8(t+1) + \{1-R2(t)\} \geqq 1$$
$$C8(t+1) + \{1-R3(t)\} \geqq 1$$

Notion of time



Not inactivated (represented by E1=1)

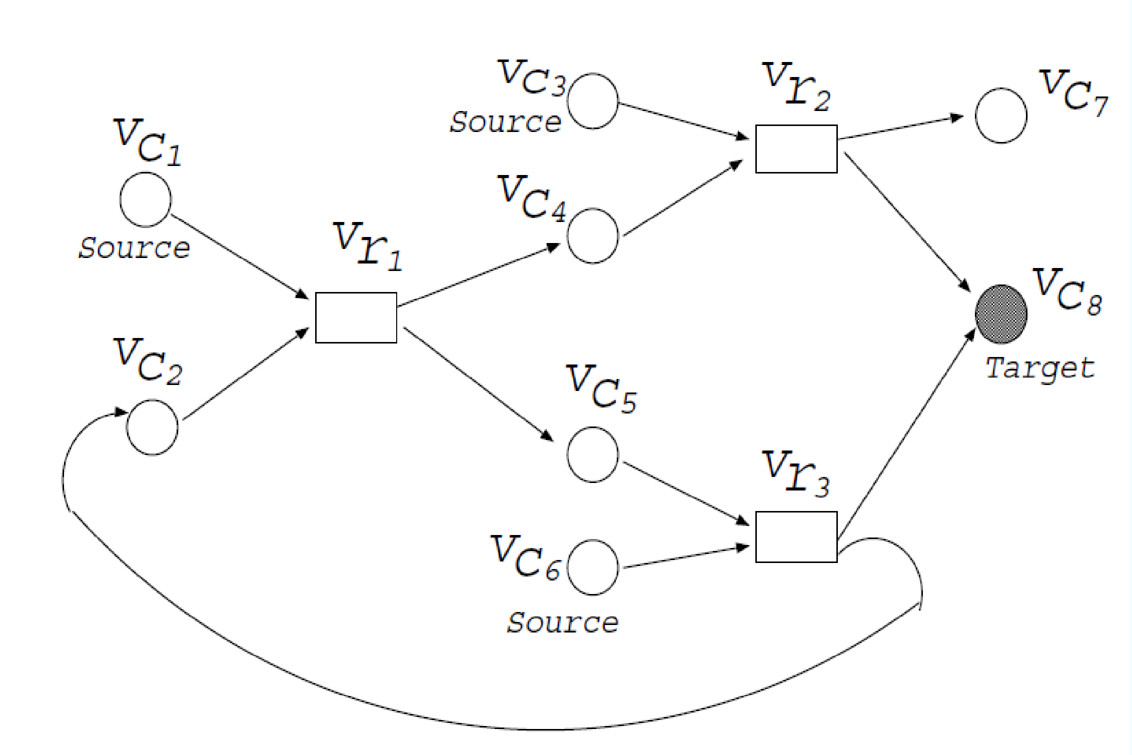
Inactivated (represented by E1=0)

Then, $R1 = C1 \wedge C2 \wedge E1$

becomes $R1(t) = C1(t-1) \wedge C2(t-1) \wedge E1(0)$.

This is further transformed into the following inequalities:

$$R1(t) + \{1-C1(t-1)\} + \{1-C2(t-1)\} + \{1-E1(0)\} \geqq 1$$
$$\{1-R1(t)\} + C1(t-1) \geqq 1$$
$$\{1-R1(t)\} + C2(t-1) \geqq 1$$
$$\{1-R1(t)\} + E1(t-1) \geqq 1$$

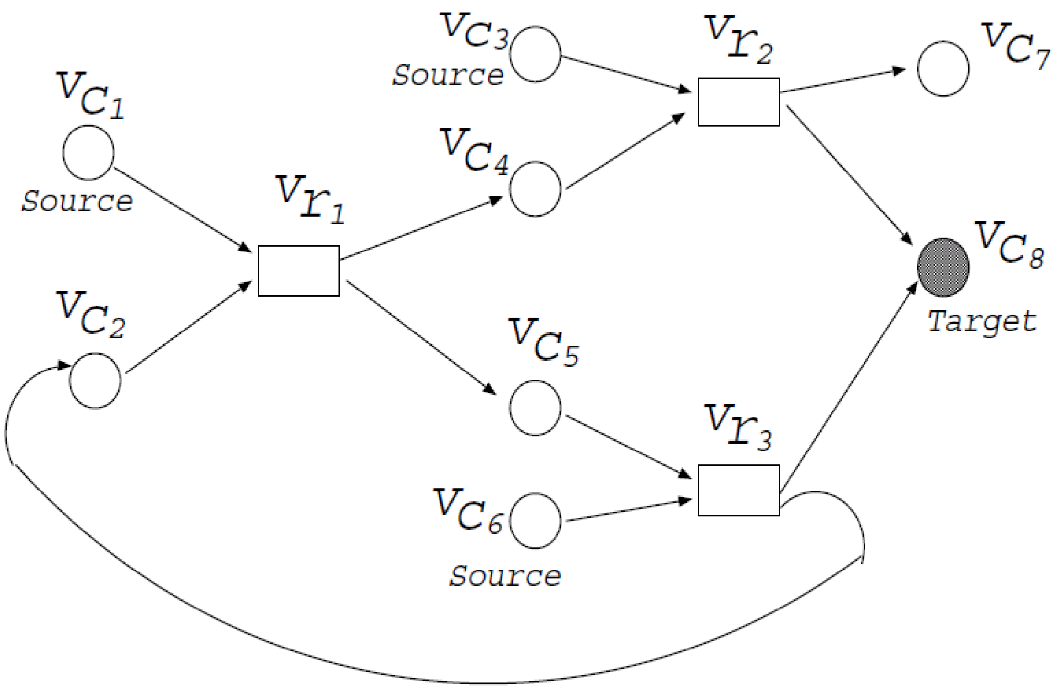


- The objective function:

  Maximize E1(0) + E2(0) + E3(0).

- Constraints:
  - C8(m+n) = 0; m:#compounds, n:#reactions
  - linear inequalities for C2,C4,C5,C7,R1,R2,R3
  - C1(0)=C3(0)=C8(0)=1

Computational time of ILP



- If the notion of time is used, #variables in ILP is $O\left((m+n)^2\right)$.

- Computational time for solving ILP is said to be proportional to an exponential function of #variables.

- Therefore it is not applicable to large networks.

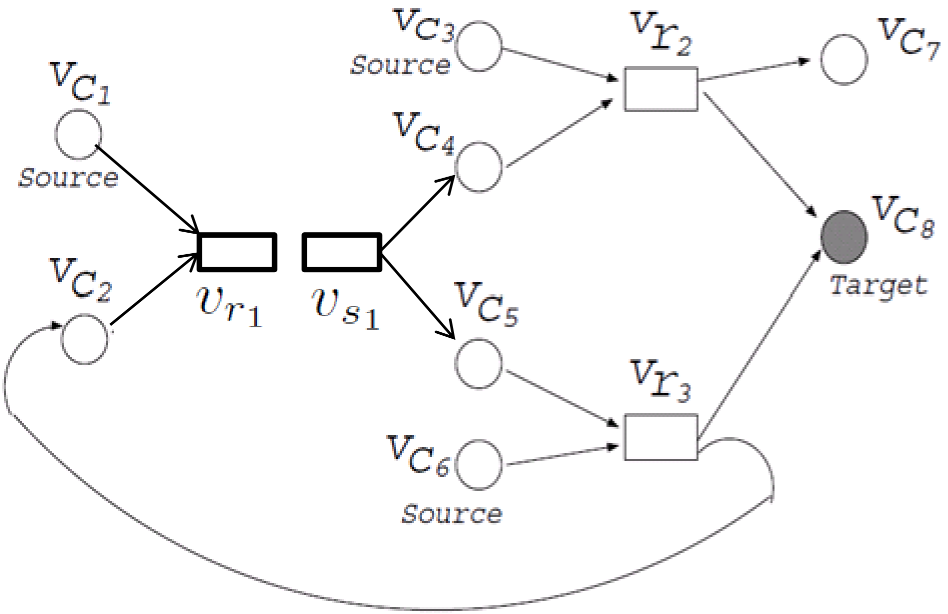
Speedup using feedback vertex set (FVS)

- If the notion of time is not used, #variables in ILP is $O(m+n)$.

- The notion of time is necessary to uniquely determine
the solution, because directed cycles may result in multiple solutions of ILP.

- Feedback vertex set
  - Removal of FVS makes the original network acyclic.

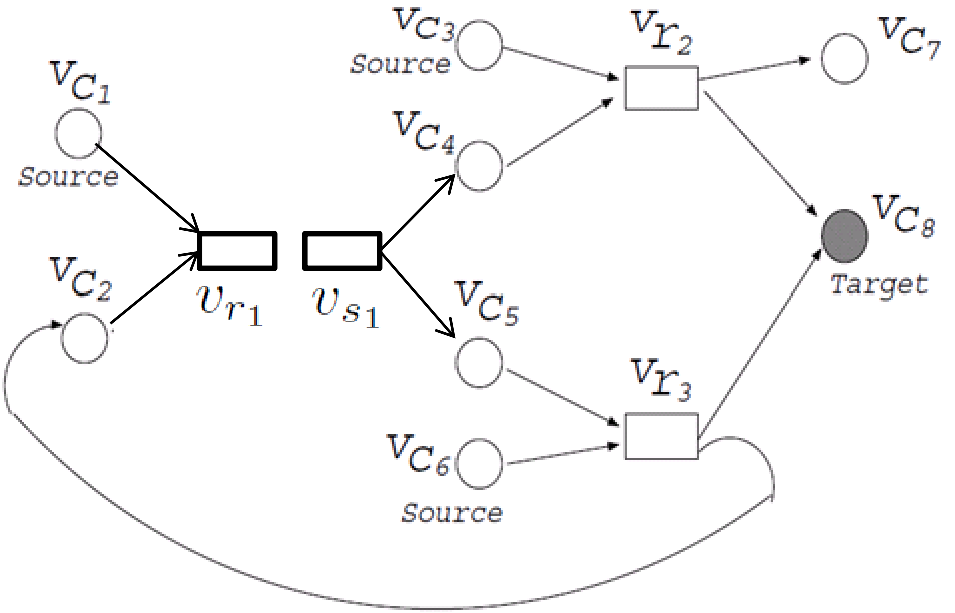
Speedup using feedback vertex set (FVS)



- Each vertex in FVS is divided into two vertecies.
    - One node has only inedges. (Type1)
    - The other node has only outedges. (Type2)

- Time advances only when the value of Type1 node is copied to Type2 node.

Speedup using feedback vertex set (FVS)



The FVS-based method decreases #variables in ILP from $O((m+n)^2)$ to $O\big(f(m+n)\big)$.

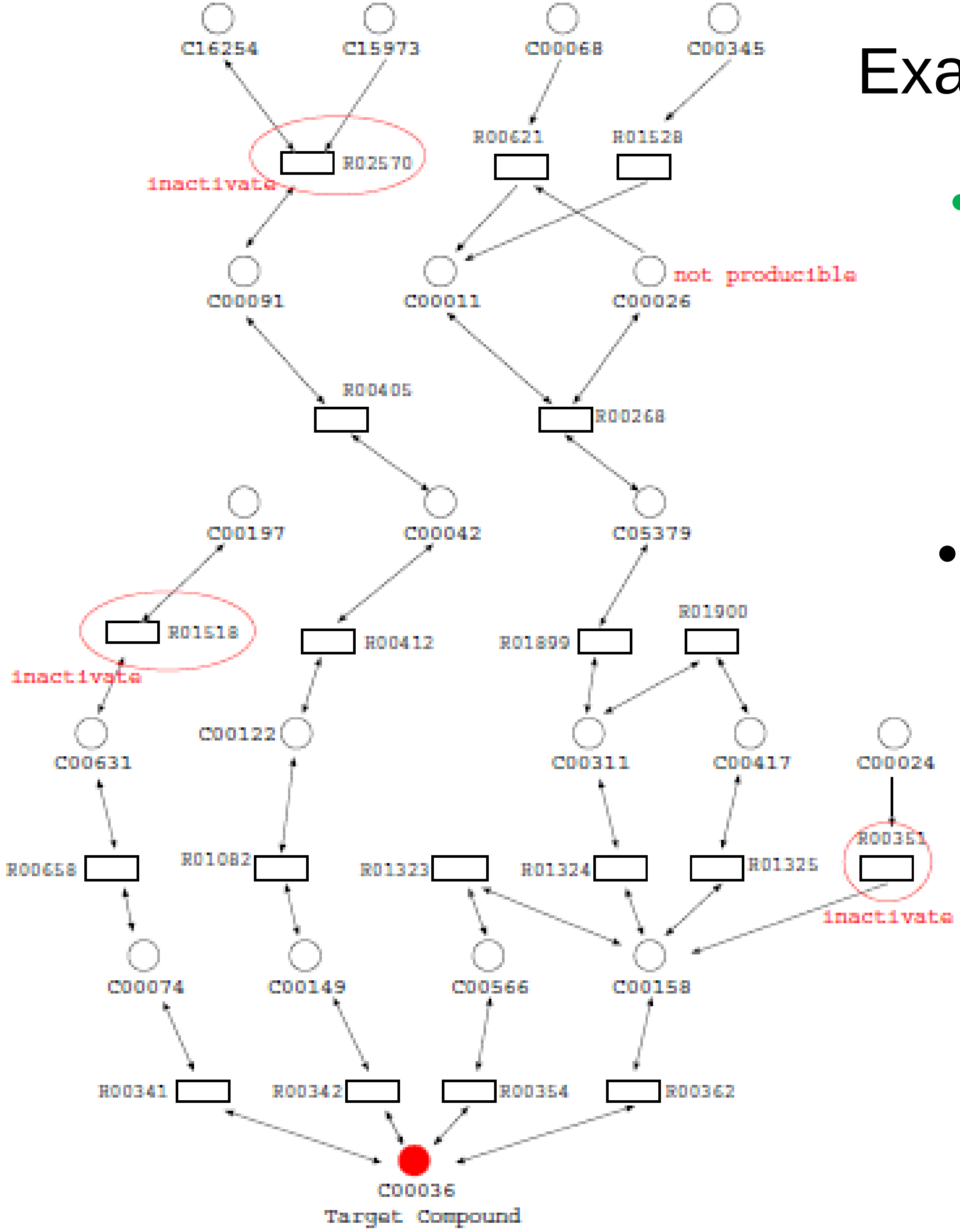
(f :the size of FVS)

Finding the minimum FVS is an NP-complete problem, but it is not necessary to find the minimum one.

Computational experiment

- We applied our method for *E. coli* metabolic network consisting of Glycolysis/gluconeogenesis (00010), Citrate cycle (00020) and Pentose phosphate pathway (00030) of KEGG database.

- Pyruvate (C00022), Acetyl-CoA (C00024), Acetate(C00033), Oxaloacetate (C00036) and Phosphoenolpyruvate （C00074) were used as target compounds.



Example

- Target compound

  Oxaloacetate
  (C00036)

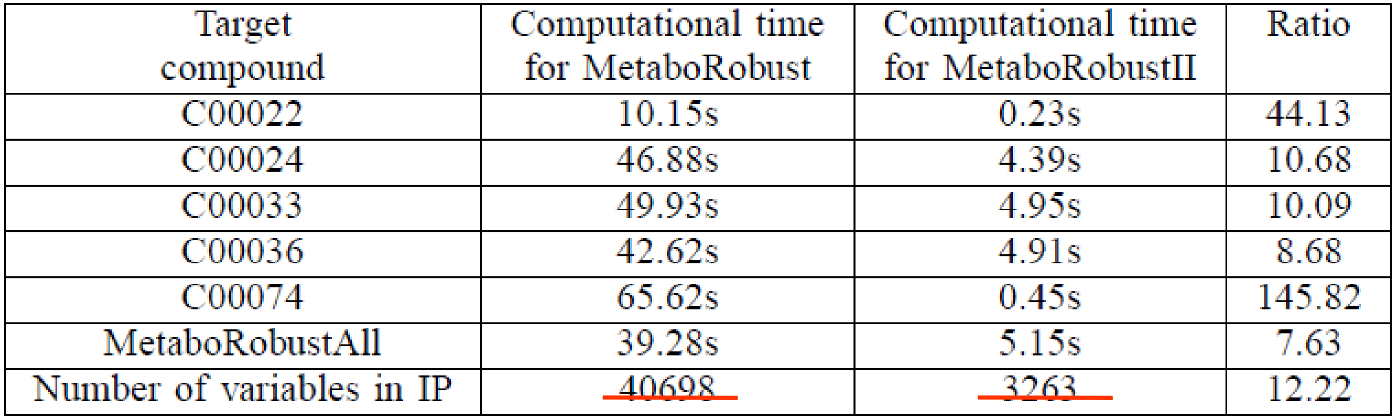
- Detected reactions

  R00351
  R01518
  R02570

Computer experiment

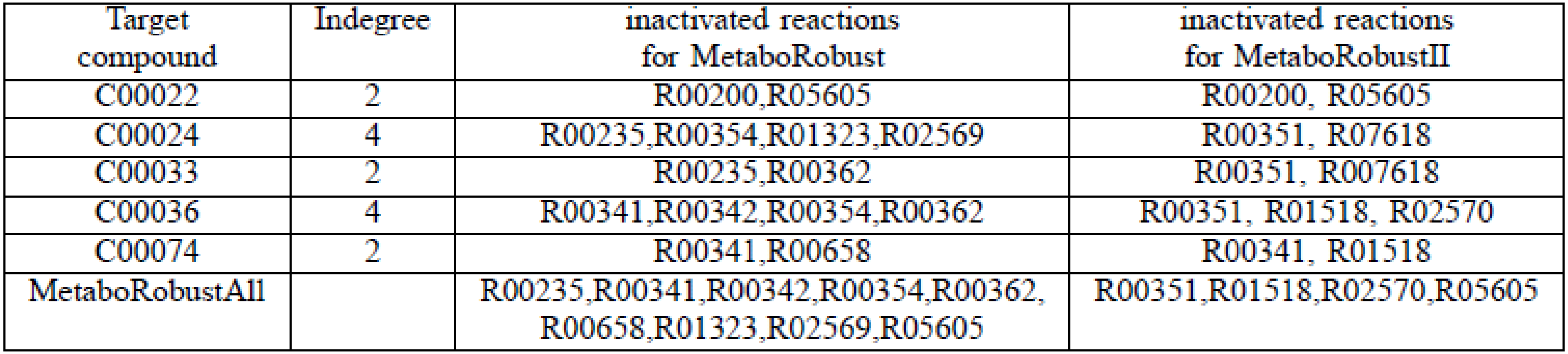
- R00351 is necessary for starting the TCA cycle.
- R01518 is included in the Embden-Meyerhof (EM) pathway generating phosphoenol pyruvate (C00074) from glycolysis.
- R02570 is related to generate Succinyl-CoA (C00091), Succinate (C00042), Fumarate(C00122) and Malate (C00149).

Computer experiment



Target compound	Computational time for MetaboRobust	Computational time for MetaboRobustII	Ratio
C00022	10.15s	0.23s	44.13
C00024	46.88s	4.39s	10.68
C00033	49.93s	4.95s	10.09
C00036	42.62s	4.91s	8.68
C00074	65.62s	0.45s	145.82
MetaboRobustAll	39.28s	5.15s	7.63
Number of variables in IP	40698	3263	12.22

Although the definitions of these two problems are slightly different from each other, we compare them to estimate the efficiency of utilizing feedback vertex sets.



Target compound	Indegree	inactivated reactions for MetaboRobust	inactivated reactions for MetaboRobustII
C00022	2	R00200,R05605	R00200, R05605
C00024	4	R00235,R00354,R01323,R02569	R00351, R07618
C00033	2	R00235,R00362	R00351, R007618
C00036	4	R00341,R00342,R00354,R00362	R00351, R01518, R02570
C00074	2	R00341,R00658	R00341, R01518
MetaboRobustAll		R00235,R00341,R00342,R00354,R00362, R00658,R01323,R02569,R05605	R00351,R01518,R02570,R05605

A naïve method

With FVS and special treatment for reversible reactions.

Finding Minimum Reaction Cuts of Metabolic Networks Under a Boolean Model Using Integer Programming and Feedback Vertex Sets

Takeyuki Tamura, Kyoto University, Japan

Kazuhiro Takemoto, University of Tokyo, Japan

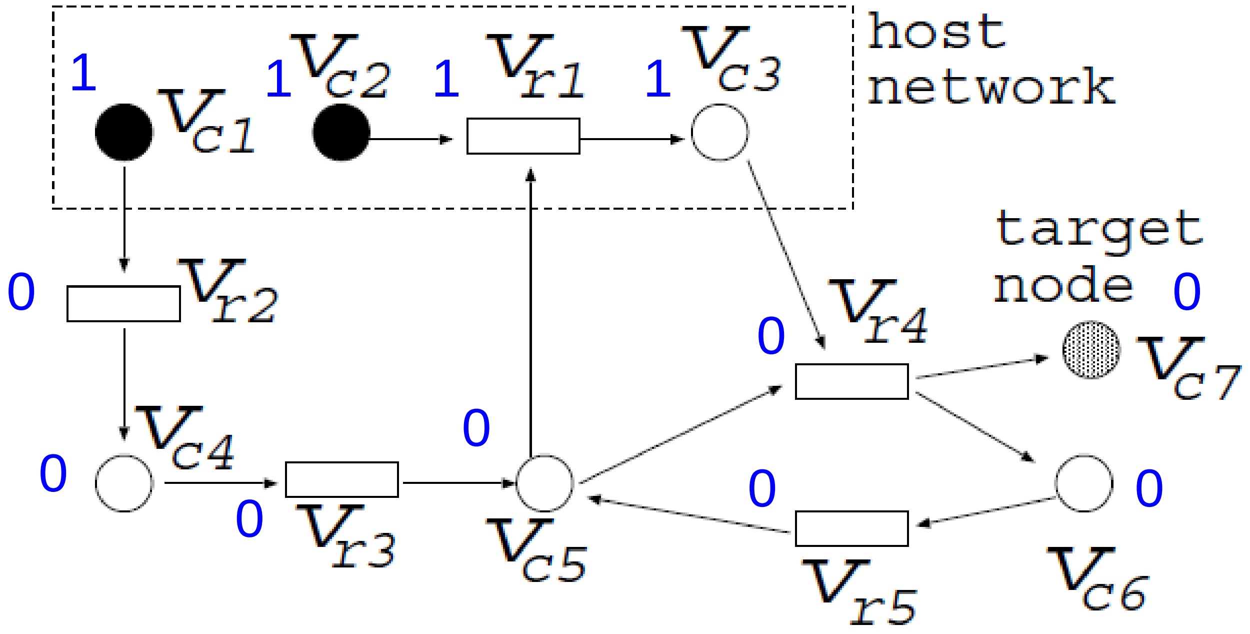
Tatsuya Akutsu, Kyoto University, Japan

ABSTRACT

In this paper, the authors consider the problem of, given a metabolic network, a set of source compounds and a set of target compounds, finding a minimum size reaction cut, where a Boolean model is used as a model of metabolic networks. The problem has potential applications to measurement of structural robustness of metabolic networks and detection of drug targets. They develop an integer programming-based method for this optimization problem. In order to cope with cycles and reversible reactions, they further develop a novel integer programming (IP) formalization method using a feedback vertex set (FVS). When applied to an E. coli metabolic network consisting of Glycolysis/Glyconeogenesis, Citrate cycle and Pentose phosphate pathway obtained from KEGG database, the FVS-based method can find an optimal set of reactions to be inactivated much faster than a naive IP-based method and several times faster than a flux balance-based method. The authors also confirm that our proposed method works even for large networks and discuss the biological meaning of our results.

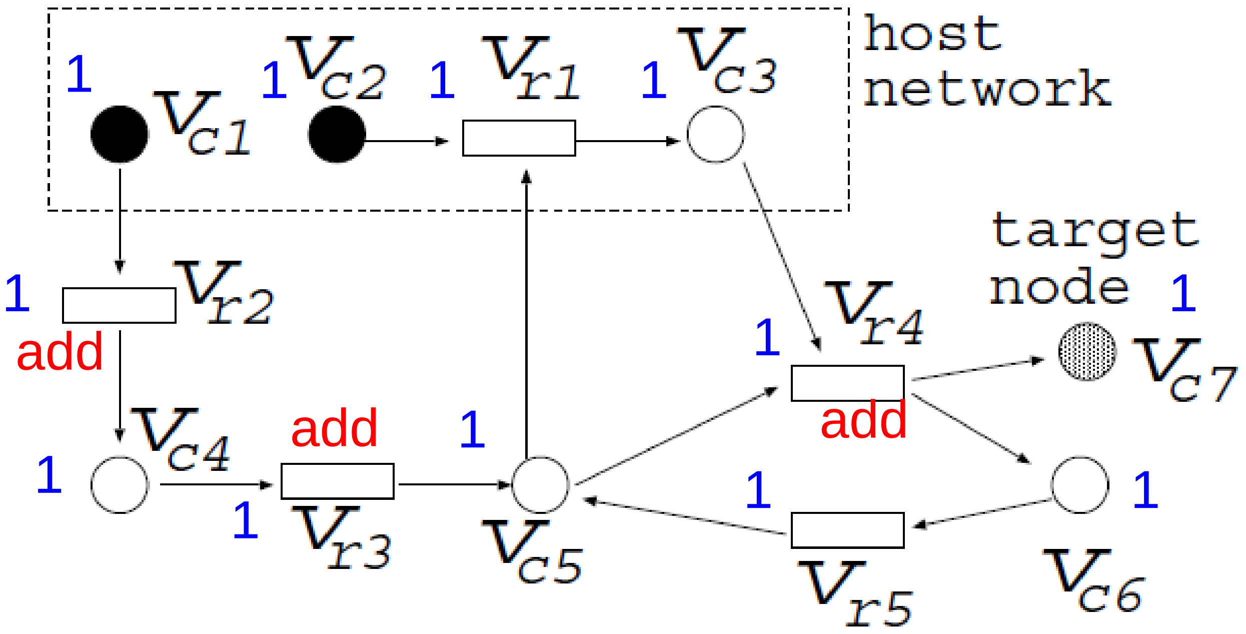
Keywords: *Metabolic Network, Reaction Cut, Flux Balance, Robustness, Integer Programming*

Minimum Reaction Insertion (MRI) Problem



Add minimum number of reactions so that the target compound becomes producible.

Minimal Valid assignment (MinVA)

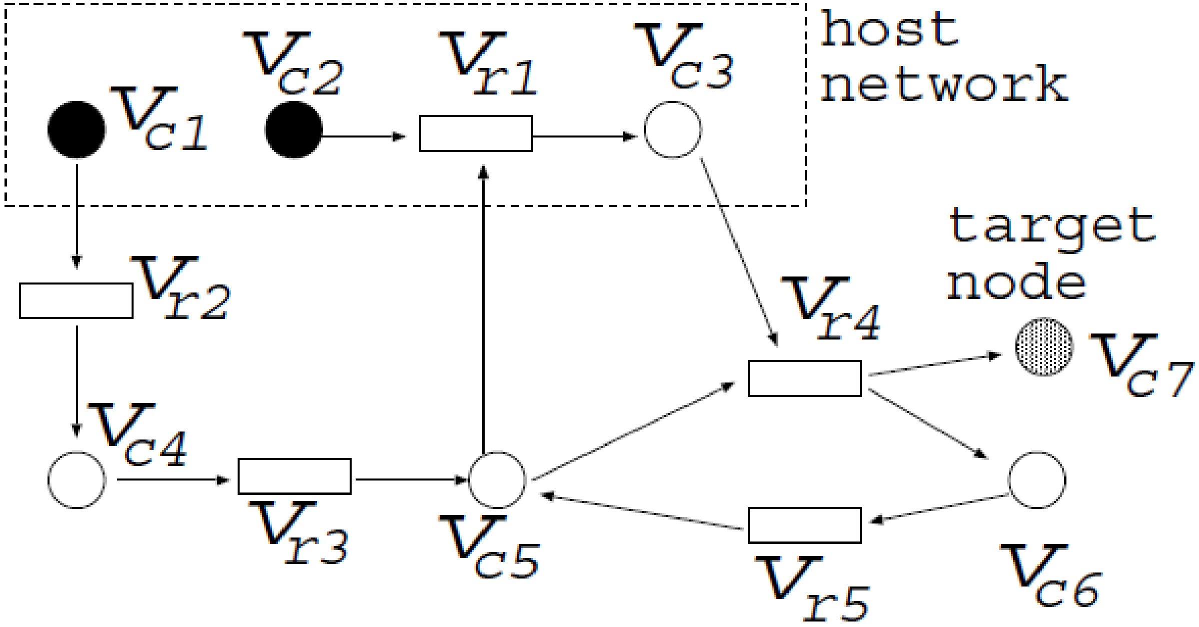


The 0-1 assignment calculated by the simple method corresponds to the Minimal Valid assignment (MinVA).

MinVA has the least number of 1s among valid assignments.

The objective function for BRM

Minimize ER2 + ER3 + ER4 + ER 5



Integer Programming-Based Method for Designing Synthetic Metabolic Networks by Minimum Reaction Insertion in a Boolean Model

Wei Lu[1], **Takeyuki Tamura**[1*], **Jiangning Song**[2,3], **Tatsuya Akutsu**[1*]

1 Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto, Japan, **2** Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Australia, **3** National Engineering Laboratory for Industrial Enzymes, Tianjin Institute of Industrial Biotechnology, Chinese Academy of Sciences, Tianjin, China

Abstract

In this paper, we consider the Minimum Reaction Insertion (MRI) problem for finding the minimum number of additional reactions from a reference metabolic network to a host metabolic network so that a target compound becomes producible in the revised host metabolic network in a Boolean model. Although a similar problem for larger networks is solvable in a flux balance analysis (FBA)-based model, the solution of the FBA-based model tends to include more reactions than that of the Boolean model. However, solving MRI using the Boolean model is computationally more expensive than using the FBA-based model since the Boolean model needs more integer variables. Therefore, in this study, to solve MRI for larger networks in the Boolean model, we have developed an efficient Integer Programming formalization method in which the number of integer variables is reduced by the notion of feedback vertex set and minimal valid assignment. As a result of computer experiments conducted using the data of metabolic networks of *E. coli* and reference networks downloaded from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database, we have found that the developed method can appropriately solve MRI in the Boolean model and is applicable to large scale-networks for which an exhaustive search does not work. We have also compared the developed method with the existing connectivity-based methods and FBA-based methods, and show the difference between the solutions of our method and the existing methods. A theoretical analysis of MRI is also conducted, and the NP-completeness of MRI is proved in the Boolean model. Our developed software is available at "http://sunflower.kuicr.kyoto-u.ac.jp/~rogi/minRect/minRect.html."

Citation: Lu W, Tamura T, Song J, Akutsu T (2014) Integer Programming-Based Method for Designing Synthetic Metabolic Networks by Minimum Reaction Insertion in a Boolean Model. PLoS ONE 9(3): e92637. doi:10.1371/journal.pone.0092637

Editor: Ying Xu, University of Georgia, United States of America

Received August 30, 2013; **Accepted** February 25, 2014; **Published** March 20, 2014

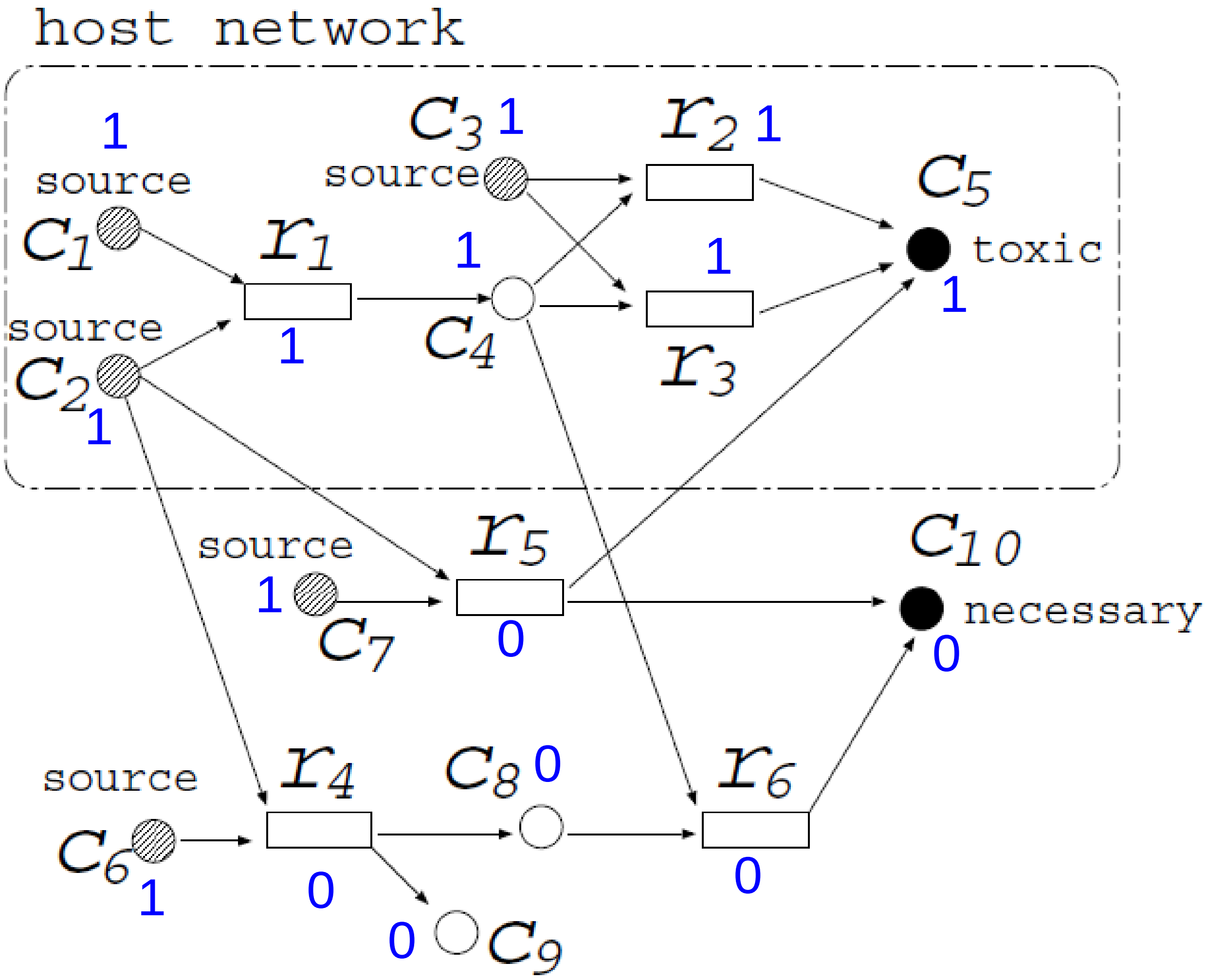
Funding: TT was partially supported by JSPS, Japan (Grant-in-Aid for Young Scientists (B)25730005 and Grant-in-Aid for Scientific Research (A) 25250028). JS is an Australian National Health and Medical Research Council (NHMRC) Peter Doherty Fellow and a recipient of the Hundred Talents Program of the Chinese Academy of Sciences (CAS). TA was partly supported by the Chinese Academy of Sciences Visiting Professorship for Senior International Scientists, China, and Grant-in-Aid #22240009 from JSPS, Japan. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: tamura@kuicr.kyoto-u.ac.jp (TT); takutsu@kuicr.kyoto-u.ac.jp (TA)

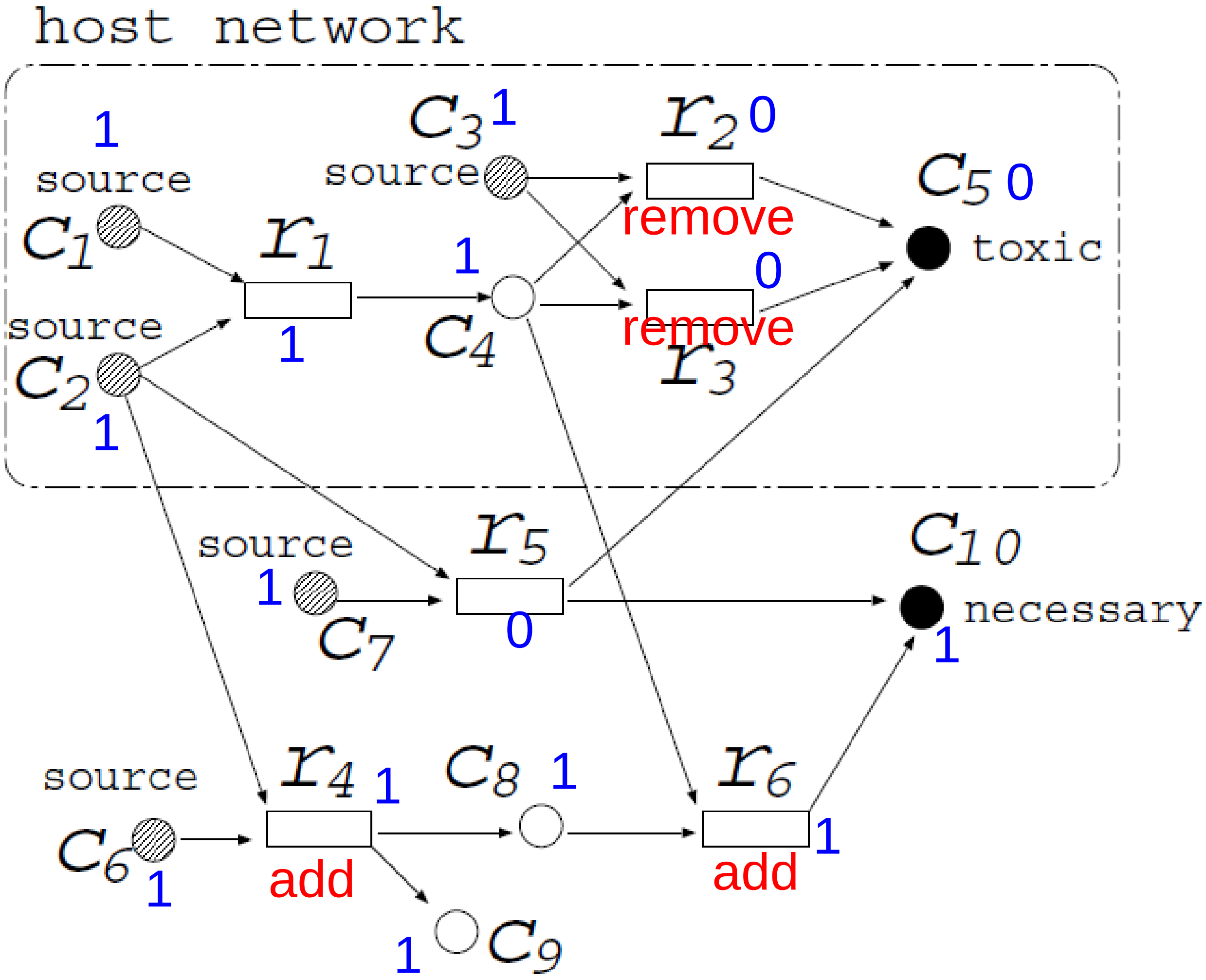
These authors contributed equally to this work.

Boolean Reaction Modification (BRM) Problem



Minimize the total number of added and removed reactions
so that the toxic(unnecessary) compound becomes non-producible
and the necessary compound becomes producible.

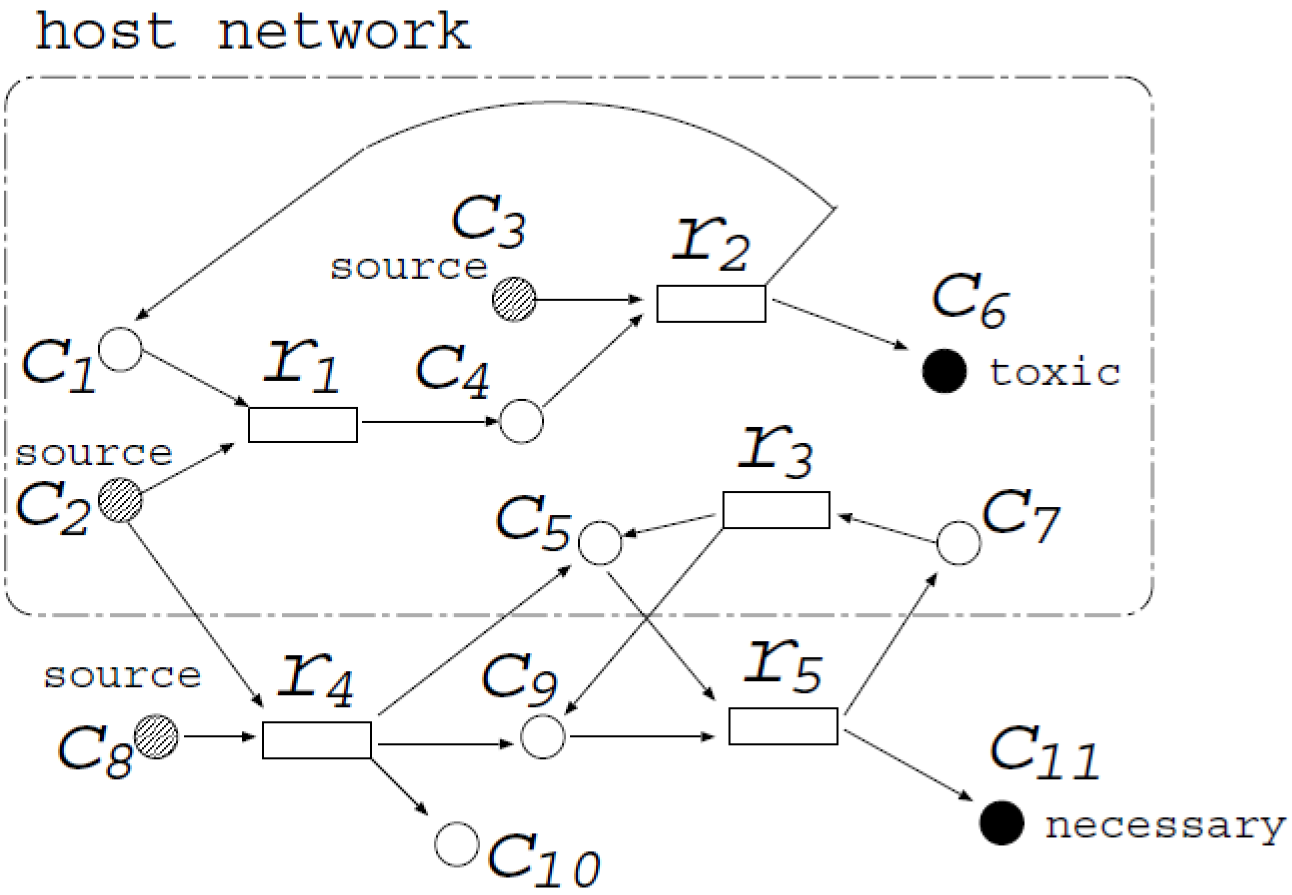
Boolean Reaction Modification (BRM) Problem



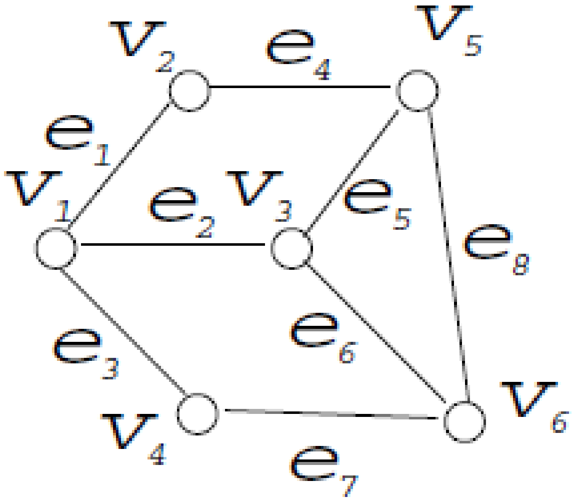
Remove {r2, r3}, and add {r4, r6}.

The objective function for BRM

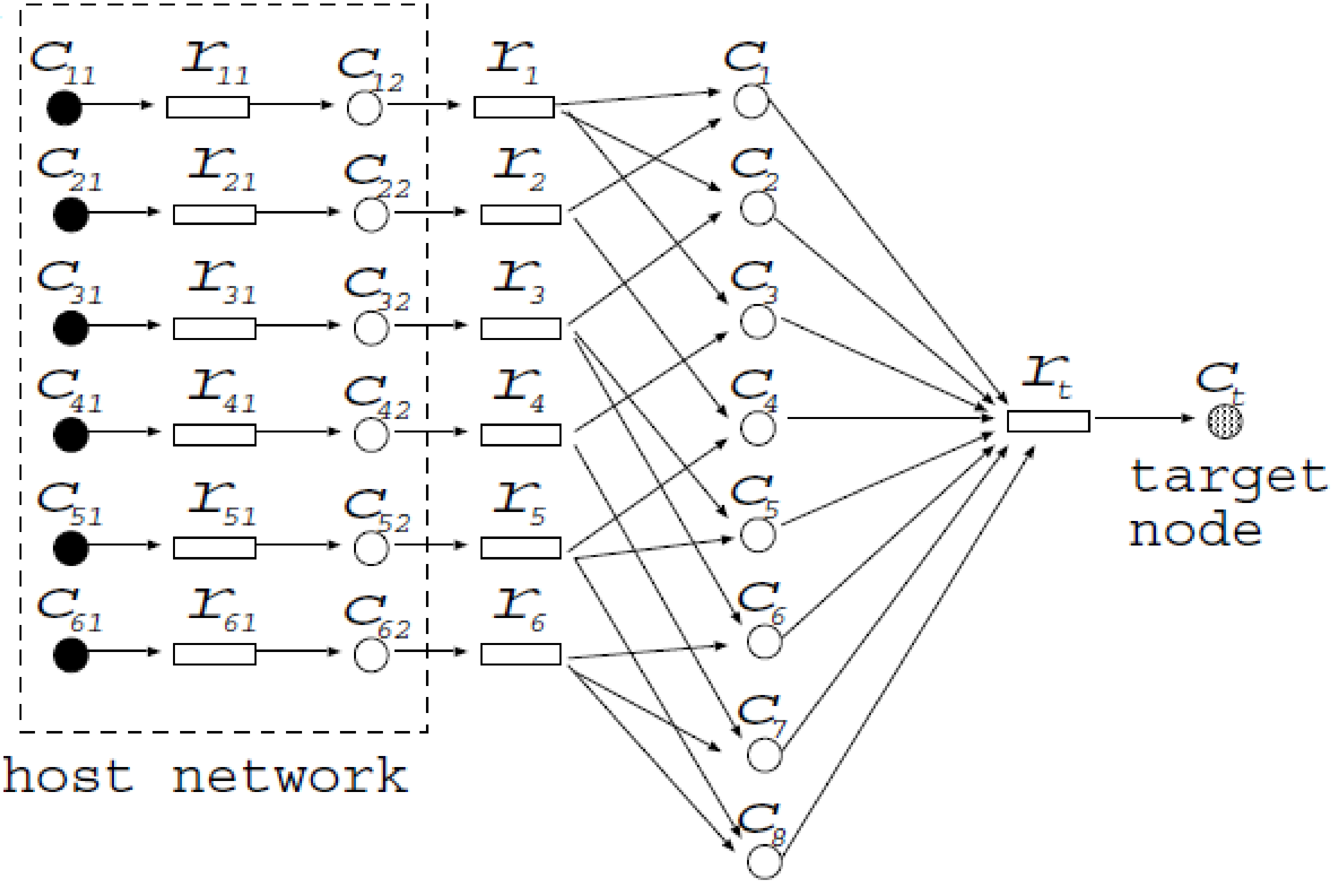
Maximize ER1 + ER2 + ER3 – ER4 – ER 5



NP-completeness of MRI problem



Polynomial time reduction from Minimum Vertex Cover problem


c_{11} r_{11} c_{12} r_1
c_{21} r_{21} c_{22} r_2
c_{31} r_{31} c_{32} r_3
c_{41} r_{41} c_{42} r_4
c_{51} r_{51} c_{52} r_5
c_{61} r_{61} c_{62} r_6
c_1
c_2
c_3
c_4
c_5
c_6
c_7
c_8
r_t
c_t
target node
host network

Computing Minimum Reaction Modifications in a Boolean Metabolic Network

Takeyuki Tamura, Wei Lu, Jiangning Song, and Tatsuya Akutsu

Abstract—In metabolic network modification, we newly add enzymes or/and knock-out genes to maximize the biomass production with minimum side-effect. Although this problem has been studied for various problem settings via mathematical models including flux balance analysis, elementary mode, and Boolean models, some important problem settings still remain to be studied. In this paper, we consider the Boolean Reaction Modification (BRM) problem, where a host metabolic network and a reference metabolic network are given in the Boolean model. The host network initially produces some toxic compounds and cannot produce some necessary compounds, but the reference network can produce the necessary compounds, and we should minimize the total number of removed reactions from the host network and added reactions from the reference network so that the toxic compounds are not producible, but the necessary compounds are producible in the resulting host network. We developed integer linear programming (ILP)-based methods for BRM, and compared them with OptStrain and SimOptStrain. The results show that our method performed better for reducing the total number of added and removed reactions, while OptStrain and SimOptStrain performed better for optimizing the production of the target compound. Our developed software is freely available at “http://sunflower.kuicr.kyoto-u.ac.jp/~rogi/solBRM/solBRM.html”.

Index Terms—metabolic network, algorithm, integer linear programming, Boolean model, flux balance analysis, feedback vertex set

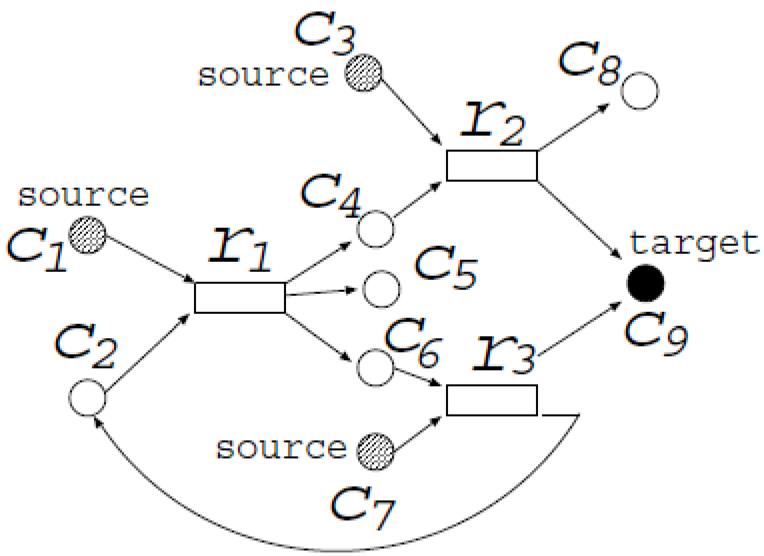
1 Introduction

Mathematical modeling of metabolic networks often consists of two phases, construction and completion [1]. The construction phase infers a metabolic reaction list from annotated genome sequence, while the completion phase converts the reaction list, associated nutrient, secretion, and biomass metabolite sets into a mathematical model. For small networks, ordinary differential equation (ODE) models are often used as they have a detailed explanatory power. However, it is difficult to obtain detailed kinetic parameters for the ODE model for larger networks. Therefore, flux balance analysis (FBA) models are often used for genome-scale metabolic networks.

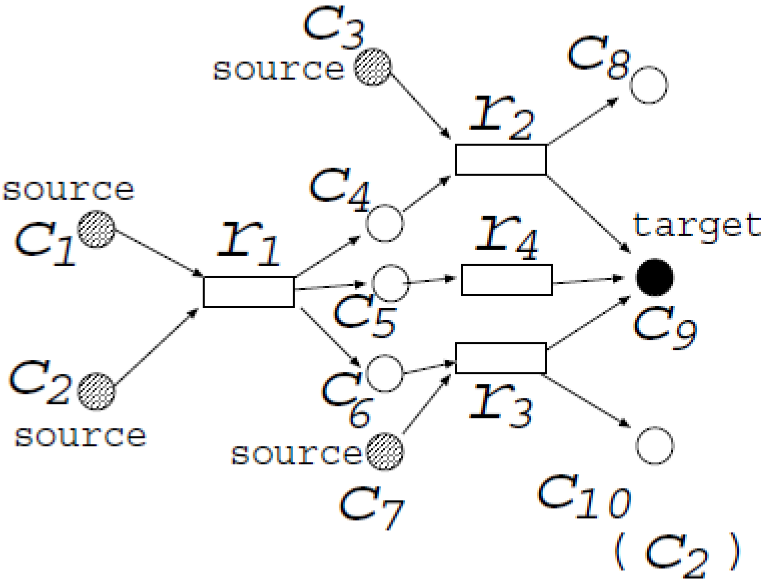
Many gap-filling and orphan-filling techniques have been developed for the completion phase of the FBA model [2]. Gap occurs when the reaction that consumes or produces a particular metabolite is completely unknown. Orphans occur when a particular reaction is known to occur, but it is

An algorithm called Biochemical Network Integrated Computational Explorer (BNICE) identifies all biochemical reactions that could link two metabolites [3]. The algorithms GapFind and GapFill are mixed integer programming (MILP)-based algorithms that can identify each gap and minimize the total number of gaps in the FBA model, respectively [4]. GrowMatch uses experimentally determined gene essentiality data to identify incorrect model predictions [5]. SMILEY is another constraint-based completion method using the relation between different nutrients and metabolite producibility [6]. Optimal Metabolic Network Identification (OMNI) is another MILP-based algorithm using data of ^{13}C labeling experiments [7]. OMNI compares experimentally measured fluxes to those predicted by FBA, and then attempts to minimize the total difference between the measured and predicted fluxes by adding or removing reactions [8].

Minimum Boolean Cut for Multiple metabolic networks



N_1



N_2

Finding minimum reaction cut to make the target compound producible in N1 and non-producible in N2.

NP-completeness of BRM

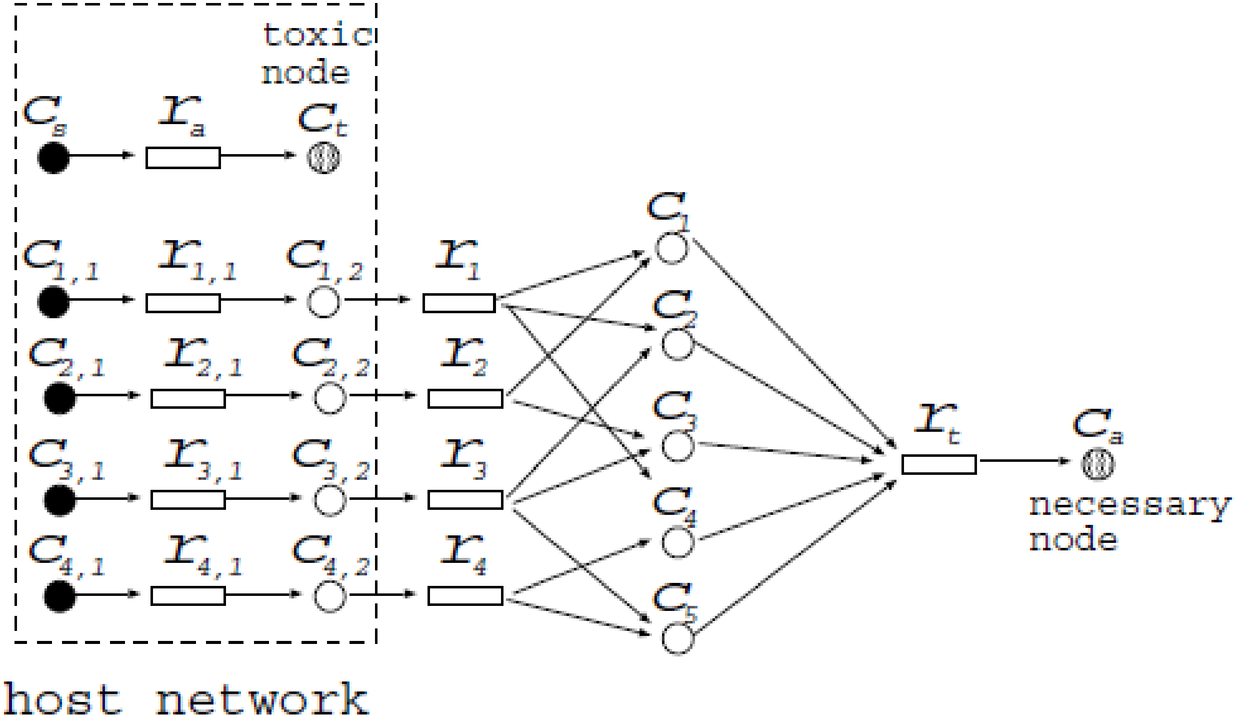
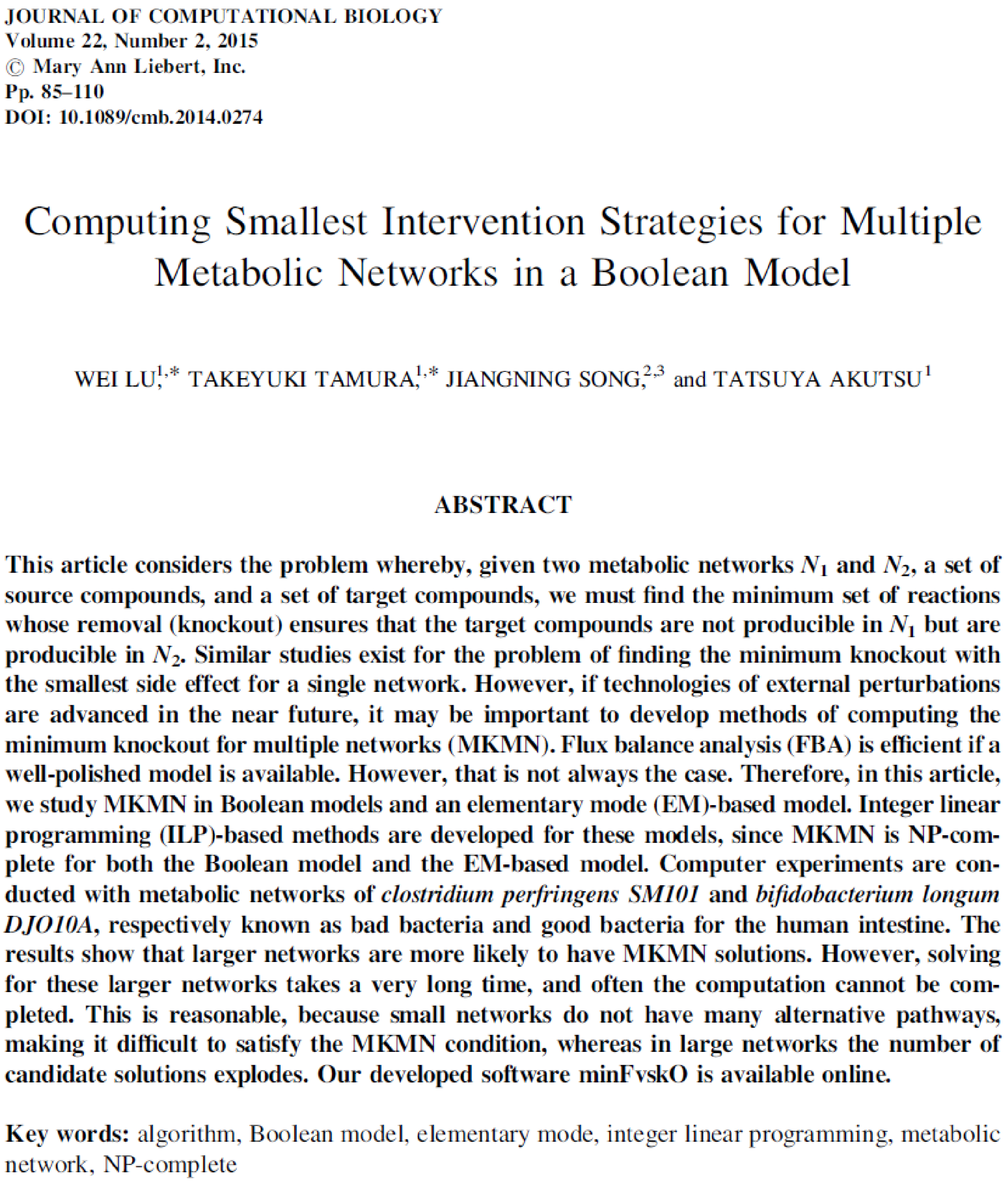


Illustration of the polynomial time reduction form Hitting Set Problem (HSP) with {1,2},{1,3},{2,3},{1,4},{3,4}.

Since BRM is NP-complete, we develop an Integer Linear Programming-based method.



JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 22, Number 2, 2015
© Mary Ann Liebert, Inc.
Pp. 85–110
DOI: 10.1089/cmb.2014.0274

Computing Smallest Intervention Strategies for Multiple Metabolic Networks in a Boolean Model

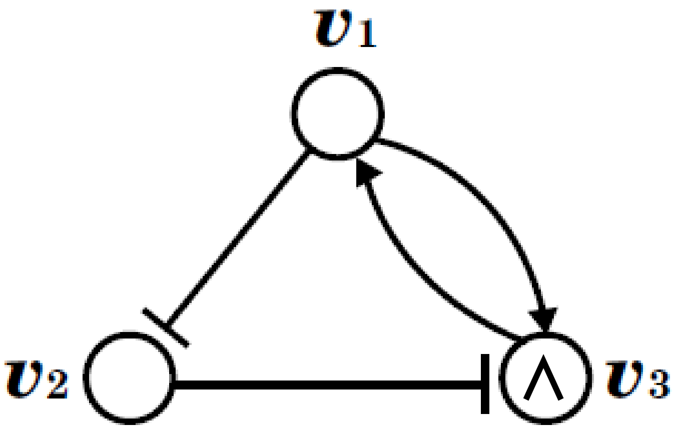
WEI LU,[1,*] TAKEYUKI TAMURA,[1,*] JIANGNING SONG,[2,3] and TATSUYA AKUTSU[1]

ABSTRACT

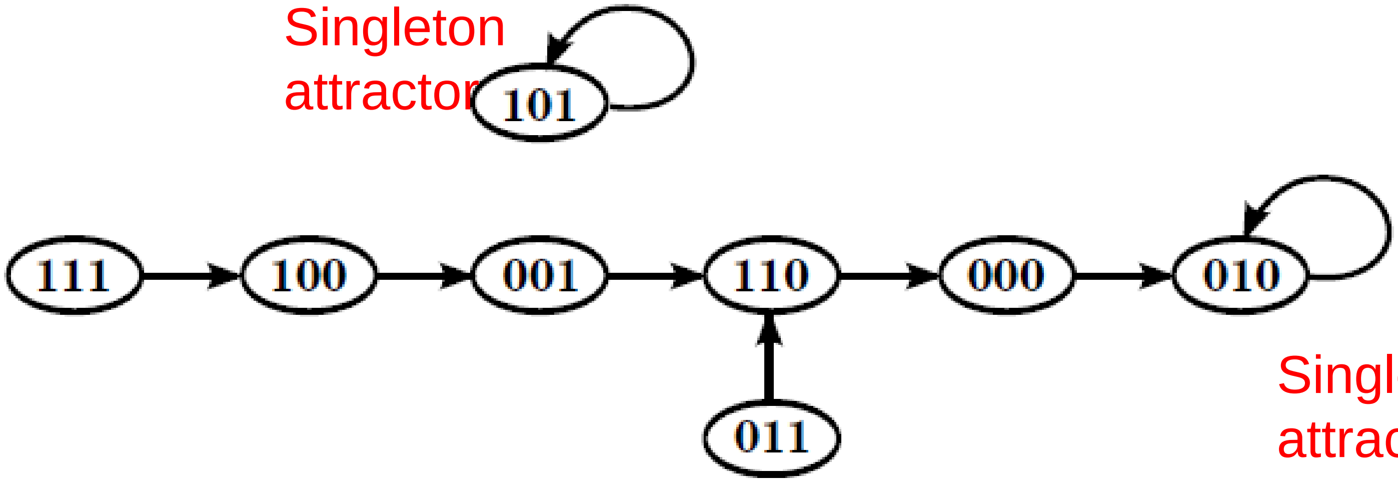
This article considers the problem whereby, given two metabolic networks N_1 and N_2, a set of source compounds, and a set of target compounds, we must find the minimum set of reactions whose removal (knockout) ensures that the target compounds are not producible in N_1 but are producible in N_2. Similar studies exist for the problem of finding the minimum knockout with the smallest side effect for a single network. However, if technologies of external perturbations are advanced in the near future, it may be important to develop methods of computing the minimum knockout for multiple networks (MKMN). Flux balance analysis (FBA) is efficient if a well-polished model is available. However, that is not always the case. Therefore, in this article, we study MKMN in Boolean models and an elementary mode (EM)-based model. Integer linear programming (ILP)-based methods are developed for these models, since MKMN is NP-complete for both the Boolean model and the EM-based model. Computer experiments are conducted with metabolic networks of *clostridium perfringens SM101* and *bifidobacterium longum DJO10A*, respectively known as bad bacteria and good bacteria for the human intestine. The results show that larger networks are more likely to have MKMN solutions. However, solving for these larger networks takes a very long time, and often the computation cannot be completed. This is reasonable, because small networks do not have many alternative pathways, making it difficult to satisfy the MKMN condition, whereas in large networks the number of candidate solutions explodes. Our developed software minFvskO is available online.

Key words: algorithm, Boolean model, elementary mode, integer linear programming, metabolic network, NP-complete

Boolean network (BN) and Singleton Attractor



$$
\begin{aligned}
v_1(t+1) &= v_3(t),\\
v_2(t+1) &= \overline{v_1(t)},\\
v_3(t+1) &= v_1(t) \wedge \overline{v_2(t)}.
\end{aligned}
$$



Singleton attractor

Singleton attractor

- A singleton attractor corresponds to the state of the cell.
    - Ex. Normal cell or cancer cell

For example, a Boolean function $x_1 = x_2 \vee \overline{x}_3$ is

$(\overline{x}_2 \wedge \overline{x}_3) \vee (x_2 \wedge \overline{x}_3) \vee (x_2 \wedge x_3)$ in DNF

Representing "AND" by linear constraints

$$
\begin{aligned}
x_{1,00} &\geq (1 - x_2) + (1 - x_3) - 1 = -x_2 - x_3 + 1 \\
x_{1,00} &\leq \tfrac{1}{2}(1 - x_2 + 1 - x_3), \\
x_{1,01} &= 0, \\
x_{1,10} &\geq x_2 + (1 - x_3) - 1 = x_2 - x_3, \\
x_{1,10} &\leq \tfrac{1}{2}(x_2 + 1 - x_3), \\
x_{1,11} &\geq x_2 + x_3 - 1, \\
x_{1,11} &\leq \tfrac{1}{2}(x_2 + x_3),
\end{aligned}
$$

Representing "OR" by linear constraints

$$
\begin{aligned}
x_1 &\leq x_{1,00} + x_{1,01} + x_{1,10} + x_{1,11}, \\
x_1 &\geq \tfrac{1}{4}(x_{1,00} + x_{1,01} + x_{1,10} + x_{1,11})
\end{aligned}
$$

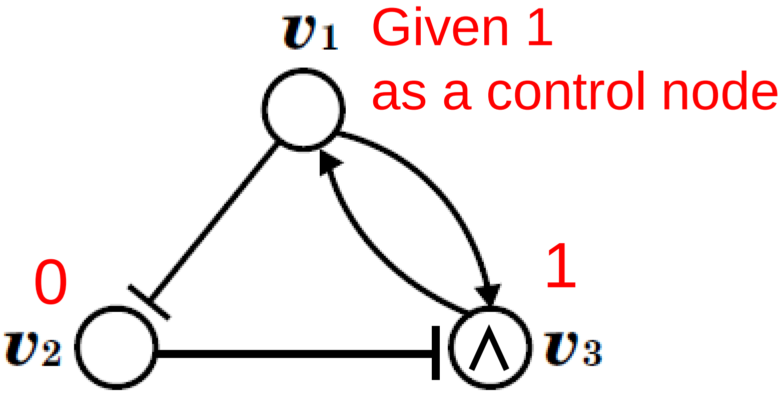
Solving Attractor Detection by ILP (Akutsu et al. 2012)

maximize x_2 (dummy function)

subject to

$$\begin{cases} x_{i,b_{i_1}\cdots b_{i_k}} \geq \left(\sum_{j=1,\cdots,k} \tau_{b_{i_j}}(x_{i_j})\right) - (k-1), \\ x_{i,b_{i_1}\cdots b_{i_k}} \leq \frac{1}{k}\sum_{j=1,\cdots,k} \tau_{b_{i_j}}(x_{i_j}), \\ \qquad \text{for all } i \in [1\cdots n] \text{ and } b_{i_1}\cdots b_{i_k} \in \{0,1\}^k, \text{such that} f_i(b_{i_1},\cdots,b_{i_k}) = 1, \\ x_{i,b_{i_1}\cdots b_{i_k}} = 0, \text{for all } i \in [1\cdots n] \text{ and } b_{i_1}\cdots b_{i_k} \in \{0,1\}^k, \text{such that } f_i(b_{i_1},\cdots,b_{i_k}) = 0, \\ x_i \leq \sum_{b_{i_1},\cdots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1}\cdots b_{i_k}}, \text{for all } i \in [1\cdots n], \\ x_i \geq \frac{1}{2^k}\sum_{b_{i_1},\cdots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1}\cdots b_{i_k}}, \\ x_i \in \{0,1\}, \text{for all } i \in [1\cdots n] \\ x_{i,b_{i_1}\cdots b_{i_k}} \in \{0,1\}, \text{for all } i \in [1\cdots n] \text{ and } b_{i_1}\cdots b_{i_k} \in \{0,1\}^k. \end{cases}$$

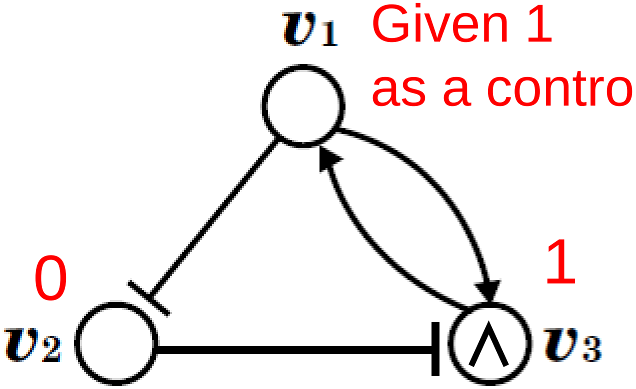
Attractor control for single BN



$$
\begin{aligned}
v_1(t+1) &= v_3(t),\\
v_2(t+1) &= \overline{v_1(t)},\\
v_3(t+1) &= v_1(t) \wedge \overline{v_2(t)}.
\end{aligned}
$$

- Suppose that we can control some nodes
  - Which nodes should be controlled?
  - What value should be assigned?
  - How the result should be evaluated?

Attractor control for single BN



Given 1
as a control node

$$v_1(t+1) = v_3(t),$$
$$v_2(t+1) = \overline{v_1(t)},$$
$$v_3(t+1) = v_1(t) \wedge \overline{v_2(t)}.$$

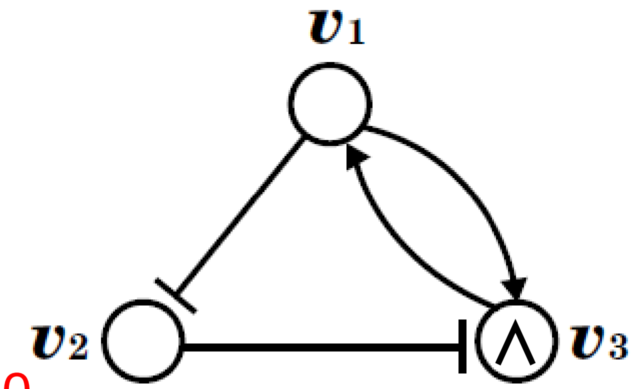
Score function $\sum_i \alpha_i \cdot (1 - w_i) \cdot x_i$

α … score for each node.
w … whether each node is chosen as a control node

Suppose that α1=1, α2=2, α3=3.

v1 = 1 : control node → score = 0
v2 = 0 : score = α2 ×0 = 2 ×0 = 0 Total score=3
v3 = 1 : score = α3 ×1 = 3 ×1 = 3

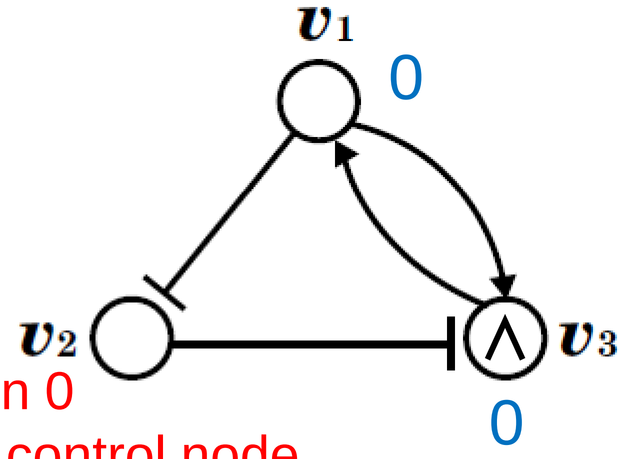
Attractor control for single BN



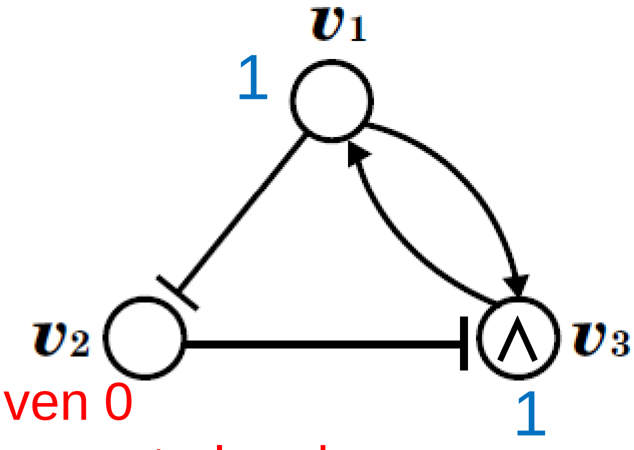
Given 0
as a control node

$$v_1(t+1) = v_3(t),$$
$$v_2(t+1) = \overline{v_1(t)},$$
$$v_3(t+1) = v_1(t) \wedge \overline{v_2(t)}.$$

There are two singleton attractors
when v2 is controlled and given 0.

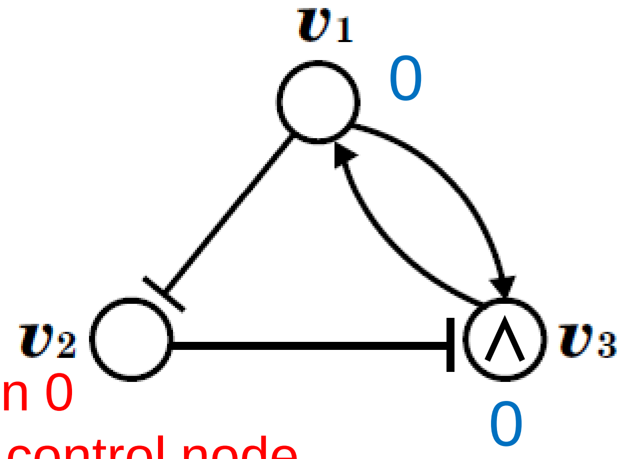


Given 0
as a control node



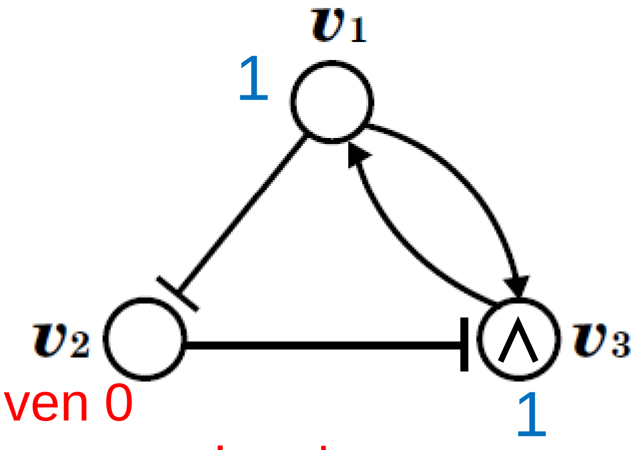
Given 0
as a control node

Choose m control nodes with the minimum score of singleton attractors greater than θ



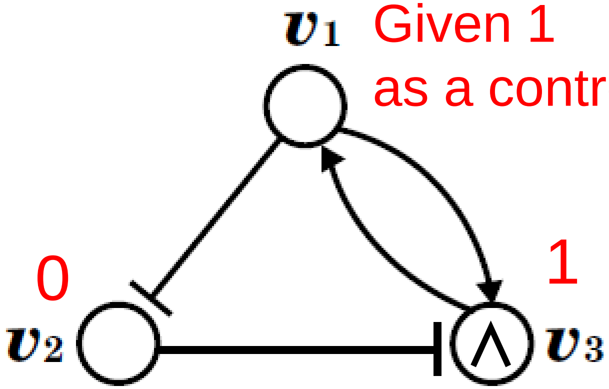
Given 0
as a control node

Score = 0 + 0 = 0



Given 0
as a control node

Score = 1 + 3 = 4

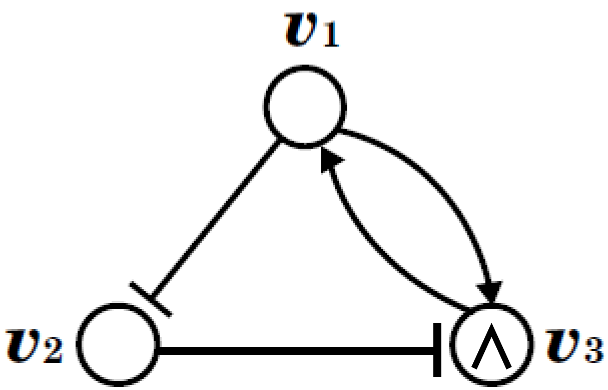


Given 1
as a control node

Score = 0 + 3 = 3

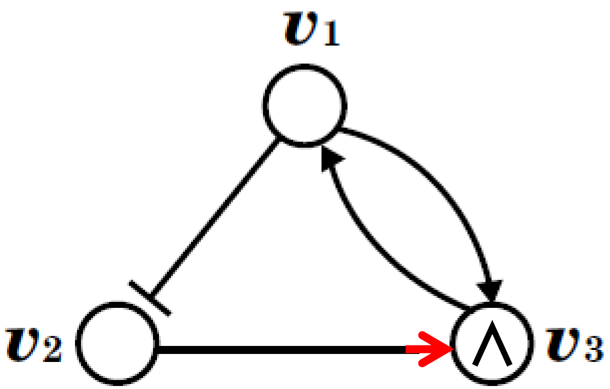
If m=1 and θ=2, then v1=1 can be a solution, but v2=0 is not.

Problem 3: Simultaneous Attractor Control (SAC)



N1 for cancer cells

$$h(\mathbf{v}) = \sum_i \alpha_i \cdot (1 - w_i) \cdot v_i$$



N2 for normal cells

$$g(\mathbf{v}) = \sum_i \beta_i \cdot (1 - w_i) \cdot v_i$$

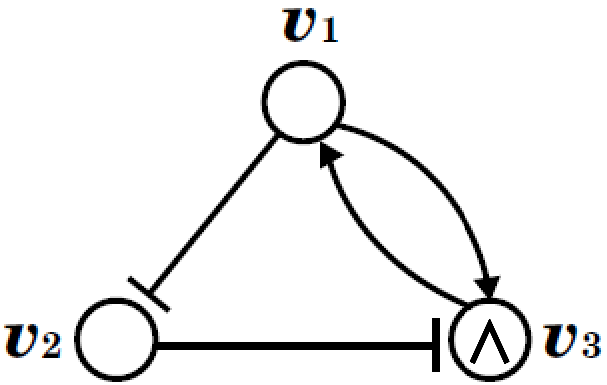
α, β … score for each node.

w … whether each node is chosen as a control node

Suppose that α1=1, α2=2, α3=3

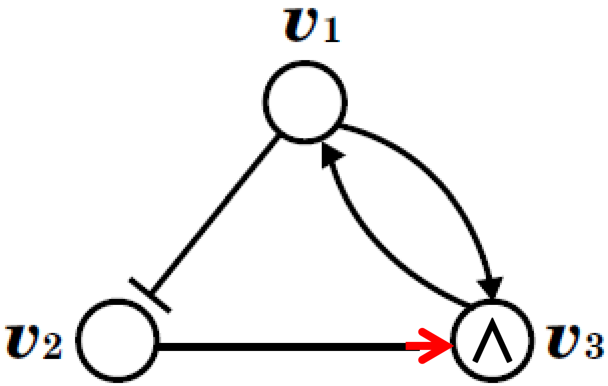
and β1=-3, β2=-1, β3=-2.

Problem 3: Simultaneous Attractor Control (SA



N1 for cancer cells

$$h(\mathbf{v}) = \sum_i \alpha_i \cdot (1 - w_i) \cdot v_i$$



N2 for normal cells

$$g(\mathbf{v}) = \sum_i \beta_i \cdot (1 - w_i) \cdot v_i$$

If v1=0 → N2 has no singleton attractor.

If v1=1 → score 4-3 1

If v2=0 → score 0+0 or 4+0 0

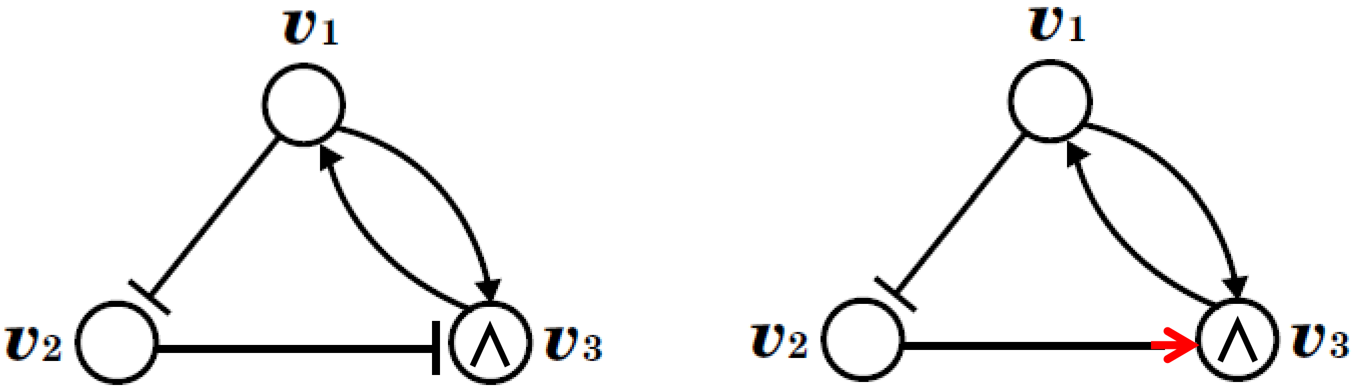
If v2=1 → score 2-1 or 2-6 -4

if v3=0 → score 2-1 1

If v3=1 → score 4-5 -1

Sum of the minimum scores of singleton attractors is used for evaluation.

ILP for Simultaneous Attractor Control (SAC)



- Variables for control nodes and the other nodes

$$x_i = \begin{cases} p_i & \text{if} \quad w_i = 0; \\ z_i & \text{if} \quad w_i = 1. \end{cases} \qquad s_i = \begin{cases} q_i & \text{if} \quad w_i = 0; \\ z_i & \text{if} \quad w_i = 1. \end{cases}$$

- Representing by linear constraints

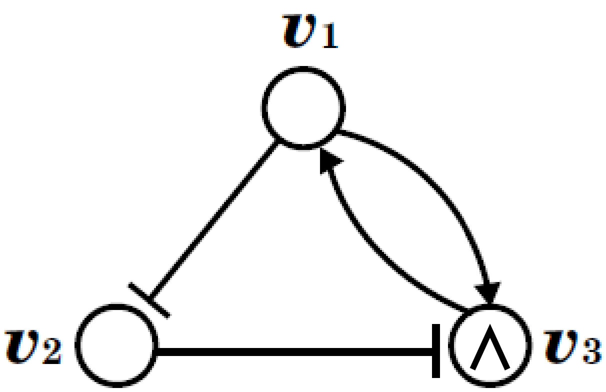
$$p_i - w_i \leq x_i \leq p_i + w_i$$

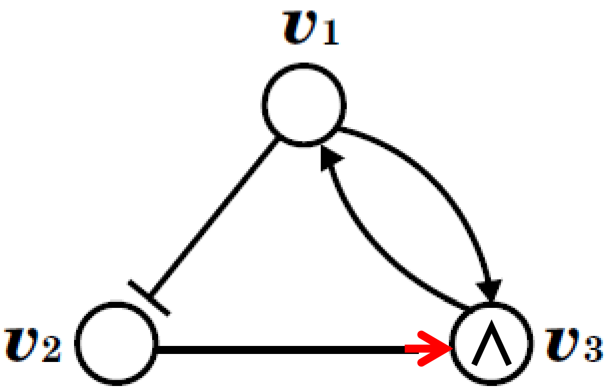
$$z_i - (1 - w_i) \leq x_i \leq z_i + (1 - w_i)$$

$$q_i - w_i \leq s_i \leq q_i + w_i$$

$$z_i - (1 - w_i) \leq s_i \leq z_i + (1 - w_i)$$

ILP for Simultaneous Attractor Control (SAC)





Score function for N1

$$h(\mathbf{v}) = \sum_i \alpha_i \cdot (1 - w_i) \cdot v_i$$

Score function for N2

$$g(\mathbf{v}) = \sum_i \beta_i \cdot (1 - w_i) \cdot v_i$$

- Representing by linear constraints

$$\sum_i \alpha_i \cdot u_i$$

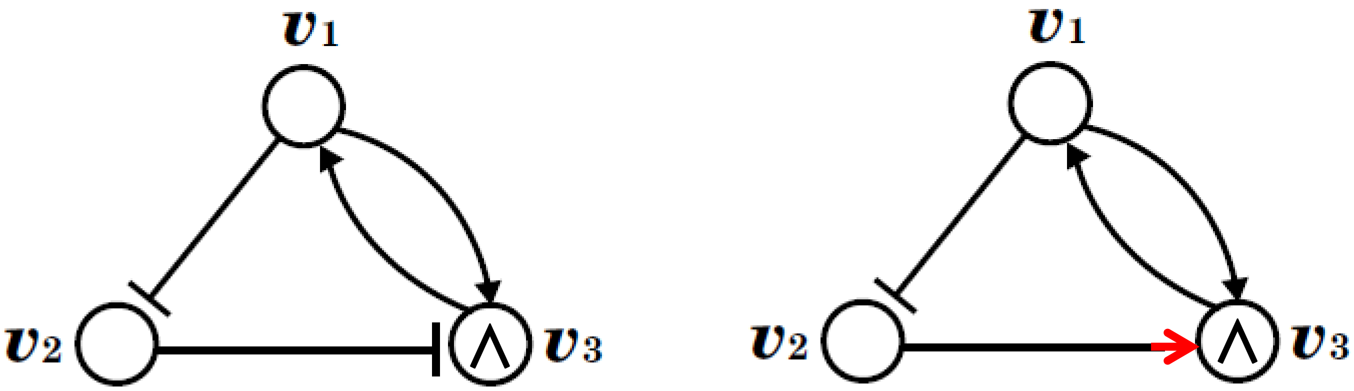
$$u_i \leq x_i \text{ and } u_i \leq 1 - w_i$$

$$\gamma_i = -\beta_i$$

$$\sum_i \gamma_i \cdot r_i$$

$$r_i \leq (1 - w_i) \text{ and } r_i \leq (1 - s_i)$$

An example of SAC for $m=1$ and $\theta=0.5$



- Firstly, find the maximum score. V2=0 (score=4) is found

If v1=0 → N2 has no singleton attractor
If v1=1 → score 4-3
If v2=0 → score 0+0 or 4+0
If v2=1 → score 2-1 or 2-6
if v3=0 → score 2-1
If v3=1 → score 4-5

- Then, under the condition of v2=0, ILP calculates the minimum score. → the minimum score (=0) < θ

Summary

- Boolean network as gene regulatory network
  - Exact exponential time algorithms for detecting singleton attractors of AND/OR BNs.
  - The algorithms are based on combinations of effective 0/1 assignment, SAT-based algorithm, and exhaustive search

- Boolean network as metabolic network
  - Integer linear programming(ILP)-based methods have been introduced for several optimization problems
  - These problems are NP-complete.
  - Feedback vertex sets were used to reduce # variables in ILP.
  - Considering maximal valid assignment and minimum valid assignment is effective.