

Constraint Programming and Biology: Biological networks

Agostino Dovier

Dept. Math and Computer Science, Univ. of Udine, Italy

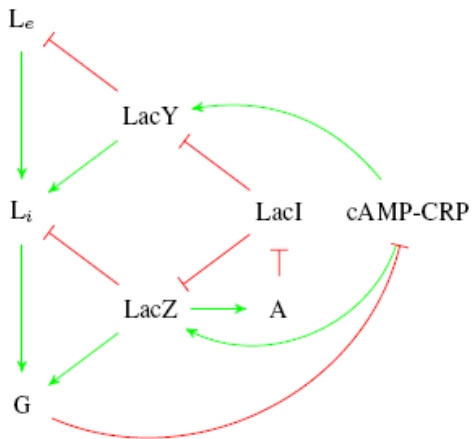
ACP Summer School in Constraint Programming
Wrocław, September 2012

Biological Networks

- We are in the area of Systems Biology
- There are a lot of approaches and tools, a lot of possibility of abstraction
- You can see some of them in WCB papers. In particular, have a look at Fages (et al) Prolog tool Biochamm
- Most approaches represent systems using networks (various kind of graphs/Petri Nets) that can handle continuous or discrete variables.
- We'll see now a simple but interesting example: discrete influence graph.

Influence Graphs

Operon Lactose in *E. coli* (example from Gebser, Schaub, Thiele, Veber, 2011)



Influence Graphs

- An **influence graph** is a directed graph $G = \langle N, E, \sigma \rangle$ s.t. $\sigma : E \rightarrow \{+, -\}$ is a labeling of the edges.
- σ can be partial. We consider it as total in this presentation.
- $i \longrightarrow j$ where $\sigma(i, j) = +$ means that i influences positively j (e.g. a positive (negative) variation of the level of i causes a positive (negative) variation of the level of j).
- $i \longrightarrow j$ where $\sigma(i, j) = -$ means that i influences negatively j (e.g. a positive (negative) variation of the level of i causes a negative (positive) variation of the level of j). It is often denoted as $i \longrightarrow\!\!-\! j$.

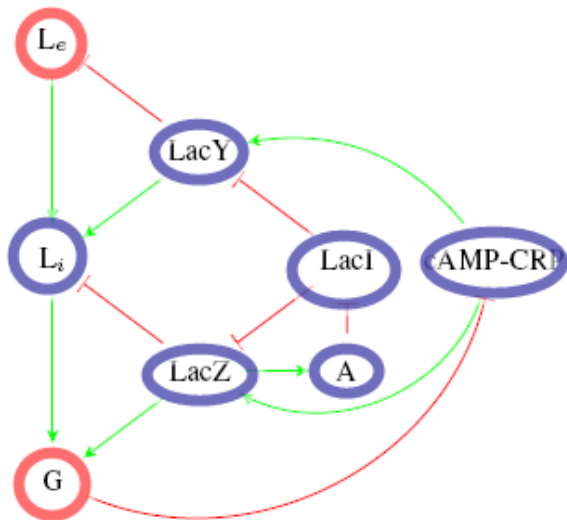
Influence Graphs

- Among the nodes there are **input** nodes, where we can increase or decrease the level of some substances
- From experimental results one builds a set of observations, namely, some partial assignments $\mu : N \rightarrow \{-, +\}$ for the “level” of the nodes.
- One of the first problems is understanding if these partial observations are “consistent”
- $G = (N, E, \sigma)$ and μ are *consistent* whether there is a *total* extension μ' of μ (defined for all nodes in N) such that **for each non-input node** $n \in N$ there is an edge $(m, n) \in E$ such that

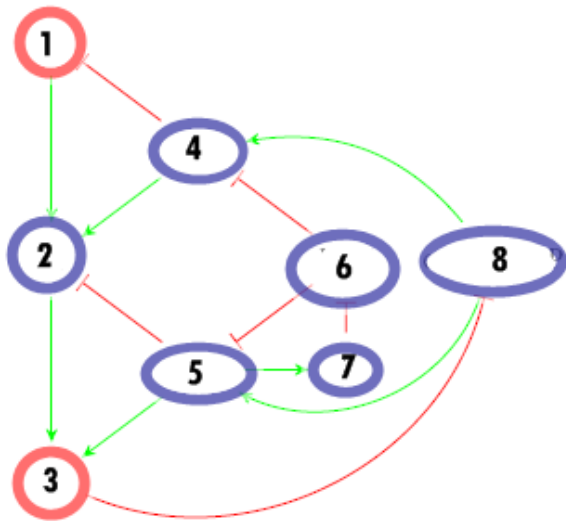
$$\sigma(m, n)\mu'(m) = \mu'(n)$$

(i.e. $++ = -- = +$, $+ - = - + = -$, using the rule of sign)

Operon Lactose in *E. coli*

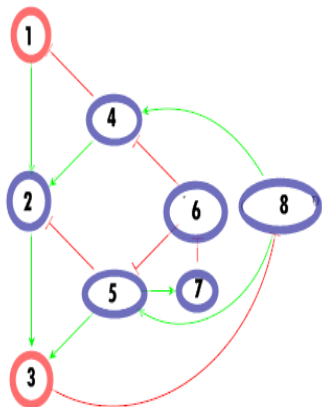


Operon Lactose in *E. coli*



Operon Lactose in E. coli

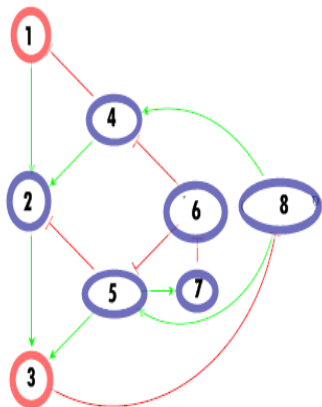
Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)

Operon Lactose in E. coli

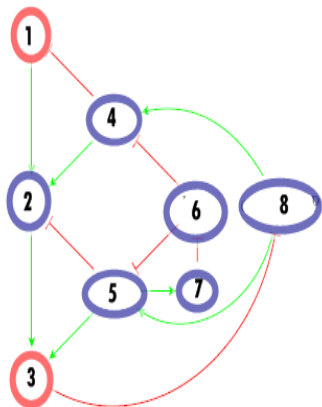
Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)
+	+	+	+	+	-	+	-	YES

Operon Lactose in E. coli

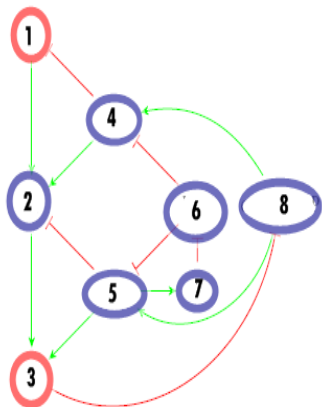
Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)
+	+	+	+	+	-	+	-	YES
+	-	-	?	?	?	?	+	SAT

Operon Lactose in E. coli

Some examples



1	2	3	4	5	6	7	8	
+	+	+	+	+	+	+	+	NO (8)
+	+	+	+	+	-	+	-	YES
+	-	-	?	?	?	?	+	SAT
+	-	-	-	-	+	-	+	YES
+	-	-	+	+	-	+	+	YES

Problem definition

Checking Consistency

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N , establish whether G and μ are consistent.

Problem definition

Checking Consistency

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N , establish whether G and μ are consistent.

If μ is total, it is just a polynomial check.

Problem definition

Checking Consistency

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N , establish whether G and μ are consistent.

If μ is total, it is just a polynomial check.

If μ is partial ...

NP-completeness

Veber et al. 2004

$$(X_1 \vee \neg X_2 \vee X_3) \wedge (\neg X_1 \vee X_2)$$

X_1

X_2

X_3

NP-completeness

Veber et al. 2004

$$(X_1 \vee \neg X_2 \vee X_3) \wedge (\neg X_1 \vee X_2)$$

X_1

X_2

X_3

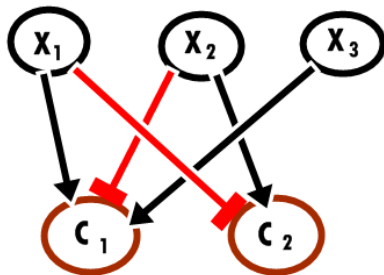
C_1

C_2

NP-completeness

Veber et al. 2004

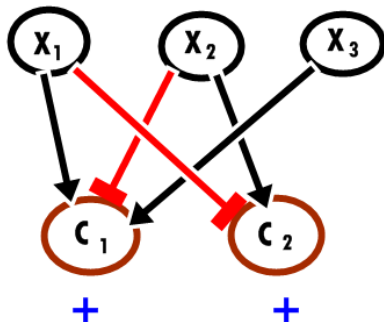
$$(X_1 \vee \neg X_2 \vee X_3) \wedge (\neg X_1 \vee X_2)$$



NP-completeness

Veber et al. 2004

$$(X_1 \vee \neg X_2 \vee X_3) \wedge (\neg X_1 \vee X_2)$$



Influence graphs

Modeling

- Let $G = (V, E)$, $V = \{V_1, \dots, V_n\}$
- Introduce X_1, \dots, X_n with domain $\{-1, 1\}$ (-1 for $-$, $+1$ for $+$)
- Assign the “known” values $X_i = \sigma(V_j)$.
- For $i = 1, \dots, n$, if V_i is not “input” then, let

$$(V_{i_1}, V_i, \sigma_{(i_1, i)}), \dots, (V_{i_k}, V_i, \sigma_{(i_k, i)})$$

be its entering edges. Then we set the constraint:

$$V_i \in \{X_{i_1} \sigma_{(i_1, i)}, \dots, X_{i_k} \sigma_{(i_k, i)}\}$$

- Running code: `influence_consistency.pl`.

Problem definition

Once inconsistency has been detected, the biologist would receive some guess on where the error can be. There are several chances. We show one.

Problem definition

Once inconsistency has been detected, the biologist would receive some guess on where the error can be. There are several chances. We show one.

Repairing

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N : find μ' such that G and μ' are consistent and μ' is obtained from μ by changing as few values as possible.

Problem definition

Once inconsistency has been detected, the biologist would receive some guess on where the error can be. There are several chances. We show one.

Repairing

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N : find μ' such that G and μ' are consistent and μ' is obtained from μ by changing as few values as possible.

This can be used for reasoning on the network.

Problem definition

Once inconsistency has been detected, the biologist would receive some guess on where the error can be. There are several chances. We show one.

Repairing

Given an influence graph $G = \langle N, E, \sigma \rangle$ and a partial assignment μ of the nodes N : find μ' such that G and μ' are consistent and μ' is obtained from μ by changing as few values as possible.

This can be used for reasoning on the network.

Similarly, one may ask for the minimum number of edges to be labeled in a different way, or to be added, and so on.

Influence graphs

Repairing

- Let $G = (V, E)$, $V = \{V_1, \dots, V_n\}$
- Introduce X_1, \dots, X_n and D_1, \dots, D_n valued in $\{-1, 1\}$
- Intuitively, X_i is the value of the node i , D_i is 1 (-1) if node i is consistent (inconsistent).
- Assign the “known” values $X_i = \sigma(V_i)$.
- For input nodes and for nodes not assigned by σ : $D_i = 1$
- For $i = 1, \dots, n$, if V_i is not “input” then, let

$$(V_{i_1}, V_i, \sigma(i_1, i)), \dots, (V_{i_k}, V_i, \sigma(i_k, i))$$

be its entering edges. Then we set the constraints:

$$V_i D_i \in \{X_{i_1} \sigma(i_1, i), \dots, X_{i_k} \sigma(i_k, i)\}$$

- Maximize $D_1 + \dots + D_n$
- Running code: `influence_repairing.pl`

Some references

- Soulé C. 2003. Graphics requirements for multistationarity. *Complexus*, 1(3):123–133.
- Siegel A., et al. 2006. Qualitative analysis of the relation between DNA microarray data and behavioral models of regulation networks. *Biosystems* 84, 2, 153–174.
- Guziolowski C. et al. 2009. Bioquali cytoscape plugin: analysing the global consistency of regulatory networks. *BMC Genomics*, 10.
- Corblin F. et al. 2009. A declarative constraint-based method for analyzing discrete genetic regulatory networks. *Biosystems* , 98(2):91-104. [Also in WCB05]
- Gebser, Schaub, Thiele, Veber. 2011 *Detecting Inconsistencies in Large Biological Networks with Answer Set Programming*. *TPLP* (2–3):323–360, 2011. [Also in WCB08]
- Guerra and Lynce. *Reasoning over Biological Networks using Maximum Satisfiability*. *Proc. of CP2012*.