

# Algorithms for subnetwork mining in heterogeneous networks

11th Symposium on Experimental Algorithms (SEA'12)

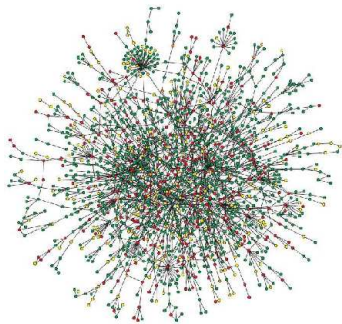
Guillaume Fertin, Hafedh Mohamed-Babou, Irena Rusu

Université de Nantes, LINA - UMR CNRS 6241, France.

Friday 8th June 2012

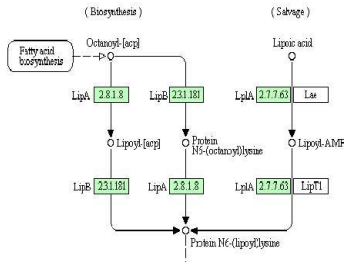
- Types of biological networks
- Biological networks comparison
- New approach : SKEW SUBGRAPH MINING (SKEWGRAM)
- Polynomial-time algorithms and harness results
- Conclusion & perspectives

# Types of biological networks



**PPI network**

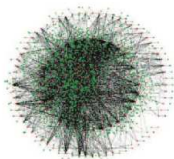
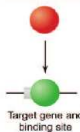
*H. Jeong et al. [2001]*



**Metabolic network**

*KEGG, pathway eco00785*

Transcription factor



**Gene regulatory network**

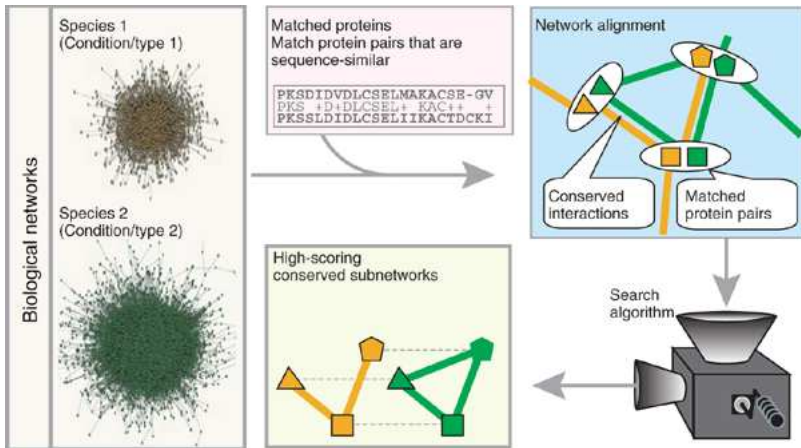
*M.M. Babu et al. [2004]*

# Homogeneous and Heterogeneous networks

- **Homogeneous** networks  $A$  and  $B$  :
  - same type, same graph representation, different species (**interspecies**).
- **Heterogeneous** networks  $A$  and  $B$  :
  - different type, different graph representation (e.g. a directed graph **vs.** an undirected graph), same species (**intraspecies**).

# Homogeneous networks comparison

Automatic methods : **networks alignment** [Kelley et al. 2003, 2006].



# Heterogeneous networks comparison

How to compare two heterogeneous networks  $A$  and  $B$ ?

Manual or case-by-case methods :

- Metabolic network **vs.** linear genome [Zheng et al. 2002], [Rison et al. 2002], [Lee et al. 2004], [Pal 2004.], [Boyer et al. 2005, 2009].
- Metabolic network **vs.** protein-protein interaction network [Pawel et al. 2008], [Huthmacher et al. 2008].

# Heterogeneous networks comparison

How to compare two heterogeneous networks  $A$  and  $B$ ?

Manual or case-by-case methods :

- Metabolic network **vs.** linear genome [Zheng et al. 2002], [Rison et al. 2002], [Lee et al. 2004], [Pal 2004.], [Boyer et al. 2005, 2009].
- Metabolic network **vs.** protein-protein interaction network [Pawel et al. 2008], [Huthmacher et al. 2008].

.. We need a unified modeling, a unified algorithmic analysis, and unified solutions to this type of problems.

- $D = (V, A)$  : directed graph ;
- $G = (V, E)$  : an undirected graph on the same vertex set of  $G$ .

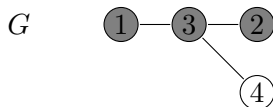
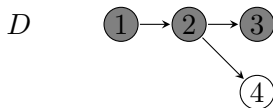
**Definition.** A path  $P$  in  $D$  is called  $(D, G)$ -**consistent** if  $G[V(P)]$  is connected.

## The problem ONE-TO-ONE SKEWGRAM

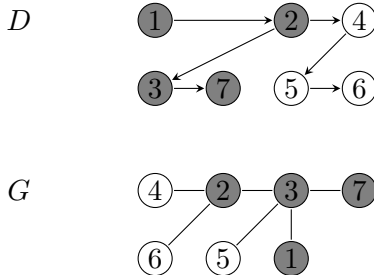
**Input.** A directed graph  $D = (V, A)$ , an undirected graph  $G = (V, A)$ .

**Question.** Find the longest  $(D, G)$ -consistent path.

# Longest $(D, G)$ -consistent path : example 1



# Longest $(D, G)$ -consistent path : example 2



# Complexity of ONE-TO-ONE SKEWGRAM

When  $D$  is an arbitrarily directed graph, ONE-TO-ONE SKEWGRAM contains the LONGEST PATH PROBLEM as a sub-problem. Thus ONE-TO-ONE SKEWGRAM is NP-complete.

When  $D$  is a DAG?

# Complexity of ONE-TO-ONE SKEWGRAM when $D$ is a DAG

$D^*$  is the underlying graph of  $D$ .

$G$	$D^*$	Tree	Outerplanar	General graph
Chordless path or cycle, (bi-)star		P		
Tree with diameter 4		P	NPC	NPC
General graph		P	NPC	NPC APX-hard

TABLE: Complexity of the ONE-TO-ONE SKEWGRAM problem.

# Complexity of the ONE-TO-ONE SKEWGRAM problem

**Theorem.** ONE-TO-ONE SKEWGRAM (in its decision version) is NP-complete, even when  $D^*$  is an outerplanar graph and  $G$  is a tree of diameter 4.

# Complexity of the ONE-TO-ONE SKEWGRAM problem

**Theorem.** ONE-TO-ONE SKEWGRAM (in its decision version) is NP-complete, even when  $D^*$  is an outerplanar graph and  $G$  is a tree of diameter 4.

- ONE-TO-ONE SKEWGRAM is in NP.

# Complexity of the ONE-TO-ONE SKEWGRAM problem

**Theorem.** ONE-TO-ONE SKEWGRAM (in its decision version) is NP-complete, even when  $D^*$  is an outerplanar graph and  $G$  is a tree of diameter 4.

- ONE-TO-ONE SKEWGRAM is in NP.

A reduction from MAX 2SAT :

- $\mathcal{V}_n = \{x_1, \dots, x_n\}$  is a set of variables;
- $\mathcal{C} = \{C_1, \dots, C_p\}$  is a set of  $p$  clauses over  $\mathcal{V}_n$ ;
- Each clause is a disjunction of two literals.

**Goal :** find an assignment that satisfies a **maximum number** of clauses.

# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

- level 0 : a vertex  $s$ ;

$$\begin{aligned} p &= 2 \text{ and } n = 3 \\ \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$

$L_0$



$D$

# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

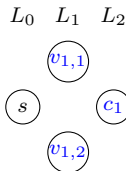
- level 0 : a vertex  $s$ ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond  
to the literals of clause  
 $C_i$ ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$   
corresponds to  $C_i$ ;

$p = 2$  and  $n = 3$

$\mathcal{C} = \{C_1, C_2\}$

$C_1 = (x_1 \vee x_3), C_2 = (\overline{x_1} \vee x_2)$

$\mathcal{X} = \{x_1, x_2, x_3\}$



$D$

# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

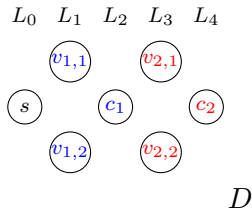
- level 0 : a vertex  $s$ ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond  
to the literals of clause  
 $C_i$ ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$   
corresponds to  $C_i$ ;

$p = 2$  and  $n = 3$

$\mathcal{C} = \{C_1, C_2\}$

$C_1 = (x_1 \vee x_3), C_2 = (\overline{x_1} \vee x_2)$

$\mathcal{X} = \{x_1, x_2, x_3\}$



# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

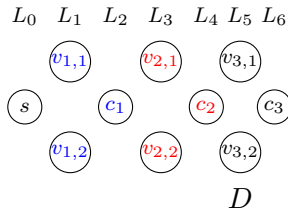
- level 0 : a vertex  $s$ ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond  
to the literals of clause  
 $C_i$ ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$   
corresponds to  $C_i$ ;
- level  $2p + 1$  :  $v_{p+1,1}$  and  
 $v_{p+1,2}$  correspond,  
respectively, to  $x_n$  and  
 $\overline{x_n}$ ;
- level  $2p + 2$  :  $c_{p+1}$ ;

$$p = 2 \text{ and } n = 3$$

$$\mathcal{C} = \{C_1, C_2\}$$

$$C_1 = (x_1 \vee x_3), C_2 = (\overline{x_1} \vee x_2)$$

$$\mathcal{X} = \{x_1, x_2, x_3\}$$



# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

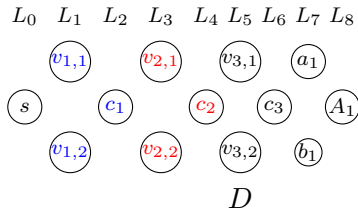
- level 0 : a vertex  $s$ ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond  
to the literals of clause  
 $C_i$ ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$   
corresponds to  $C_i$ ;
- level  $2p + 1$  :  $v_{p+1,1}$  and  
 $v_{p+1,2}$  correspond,  
respectively, to  $x_n$  and  
 $\overline{x_n}$ ;
- level  $2p + 2$  :  $c_{p+1}$ ;
- level  $2p + 2 + 2j - 1$ ,  
 $1 \leq j \leq n$  :  $a_j$  and  $b_j$   
correspond, respectively,  
to  $x_j$  and  $\overline{x_j}$ ;
- level  $2p + 2 + 2i$ ,  
 $1 \leq j \leq n - 1$  :  $A_j$ .

$$p = 2 \text{ and } n = 3$$

$$\mathcal{C} = \{C_1, C_2\}$$

$$C_1 = (x_1 \vee x_3), C_2 = (\overline{x_1} \vee x_2)$$

$$\mathcal{X} = \{x_1, x_2, x_3\}$$

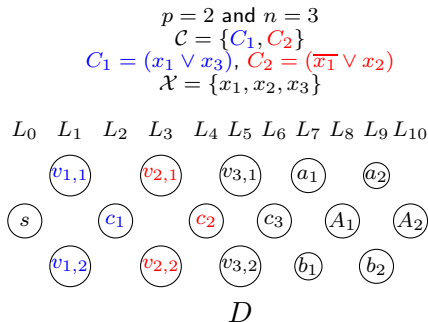


# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

- level 0 : a vertex  $s$ ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond  
to the literals of clause  
 $C_i$ ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$   
corresponds to  $C_i$ ;
- level  $2p + 1$  :  $v_{p+1,1}$  and  
 $v_{p+1,2}$  correspond,  
respectively, to  $x_n$  and  
 $\overline{x_n}$ ;
- level  $2p + 2$  :  $c_{p+1}$ ;
- level  $2p + 2 + 2j - 1$ ,  
 $1 \leq j \leq n$  :  $a_j$  and  $b_j$   
correspond, respectively,  
to  $x_j$  and  $\overline{x_j}$ ;
- level  $2p + 2 + 2i$ ,  
 $1 \leq j \leq n - 1$  :  $A_j$ .



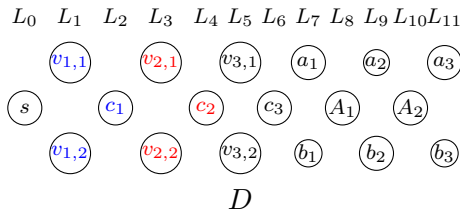
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

- level 0 : a vertex  $s$ ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond  
to the literals of clause  
 $C_i$ ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$   
corresponds to  $C_i$ ;
- level  $2p + 1$  :  $v_{p+1,1}$  and  
 $v_{p+1,2}$  correspond,  
respectively, to  $x_n$  and  
 $\overline{x_n}$ ;
- level  $2p + 2$  :  $c_{p+1}$ ;
- level  $2p + 2 + 2j - 1$ ,  
 $1 \leq j \leq n$  :  $a_j$  and  $b_j$   
correspond, respectively,  
to  $x_j$  and  $\overline{x_j}$ ;
- level  $2p + 2 + 2i$ ,  
 $1 \leq j \leq n - 1$  :  $A_j$ .

$$\begin{aligned}
 p &= 2 \text{ and } n = 3 \\
 \mathcal{C} &= \{C_1, C_2\} \\
 C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\
 \mathcal{X} &= \{x_1, x_2, x_3\}
 \end{aligned}$$



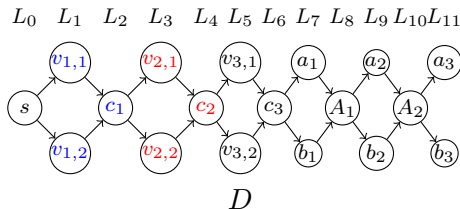
## Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

- level 0 : a vertex  $s$  ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  $v_{i,1}$  and  $v_{i,2}$  correspond to the literals of clause  $C_i$  ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$  corresponds to  $C_i$  ;
- level  $2p + 1$  :  $v_{p+1,1}$  and  $v_{p+1,2}$  correspond, respectively, to  $x_n$  and  $\overline{x_n}$  ;
- level  $2p + 2$  :  $c_{p+1}$  ;
- level  $2p + 2 + 2j - 1$ ,  $1 \leq j \leq n$  :  $a_j$  and  $b_j$  correspond, respectively, to  $x_j$  and  $\overline{x_j}$  ;
- level  $2p + 2 + 2i$ ,  $1 \leq j \leq n - 1$  :  $A_j$ .

$$\begin{aligned} p &= 2 \text{ and } n = 3 \\ \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



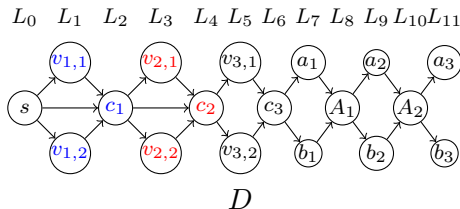
## Complexity of the ONE-TO-ONE SKEWGRAM problem

Let  $\mathcal{C} = \{C_1, \dots, C_p\}$  and  $\mathcal{V}_n = \{x_1, \dots, x_n\}$  be an instance of MAX 2SAT.

We build the directed graph  $D$  on  $2p + 2n + 2$  levels.

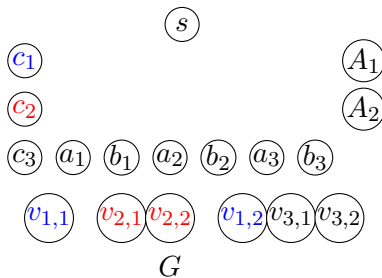
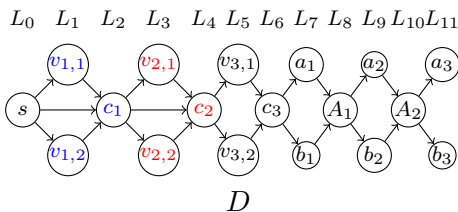
- level 0 : a vertex  $s$  ;
- level  $2i - 1$ ,  $1 \leq i \leq p$  :  
 $v_{i,1}$  and  $v_{i,2}$  correspond to the literals of clause  $C_i$  ;
- level  $2i$ ,  $1 \leq i \leq p$  :  $c_i$  corresponds to  $C_i$  ;
- level  $2p + 1$  :  $v_{p+1,1}$  and  $v_{p+1,2}$  correspond, respectively, to  $x_n$  and  $\overline{x_n}$  ;
- level  $2p + 2$  :  $c_{p+1}$  ;
- level  $2p + 2 + 2j - 1$ ,  
 $1 \leq j \leq n$  :  $a_j$  and  $b_j$  correspond, respectively, to  $x_j$  and  $\overline{x_j}$  ;
- level  $2p + 2 + 2i$ ,  
 $1 \leq j \leq n - 1$  :  $A_j$ .

$$\begin{aligned} p &= 2 \text{ and } n = 3 \\ \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



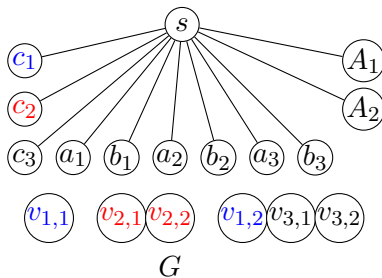
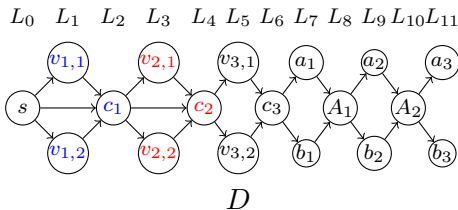
# Complexity of the ONE-TO-ONE SKEWGRAM problem

$$\begin{aligned}\mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\}\end{aligned}$$



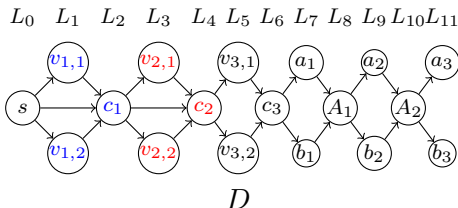
# Complexity of the ONE-TO-ONE SKEWGRAM problem

$$\begin{aligned}\mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\}\end{aligned}$$

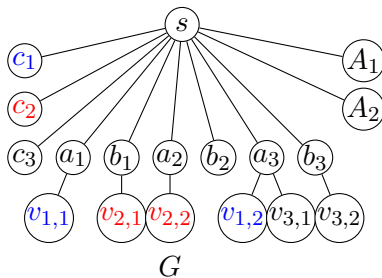


# Complexity of the ONE-TO-ONE SKEWGRAM problem

$$\begin{aligned}\mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\}\end{aligned}$$

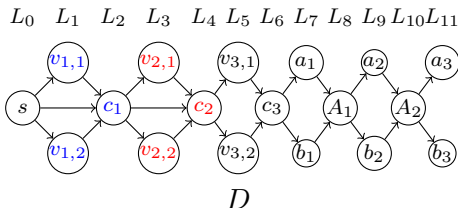


We put an edge  
between  $a_i$  (resp.  $b_i$ ) and  
any vertex  $v_{l,m}$  corresponding  
to  $x_i$  (resp.  $\overline{x_i}$ ).

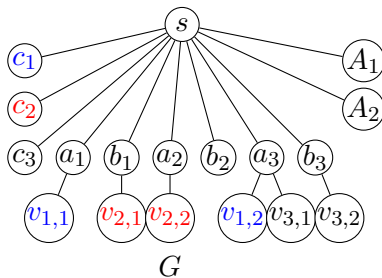


# Complexity of the ONE-TO-ONE SKEWGRAM problem

$$\begin{aligned}\mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\}\end{aligned}$$

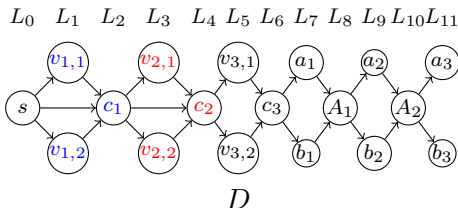


**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $p + k + 2 + 2n$  vertices.

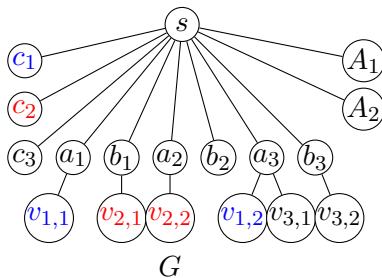


# Complexity of the ONE-TO-ONE SKEWGRAM problem

$$\begin{aligned}\mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\}\end{aligned}$$



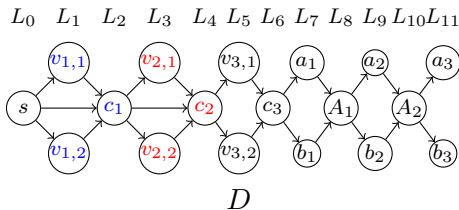
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



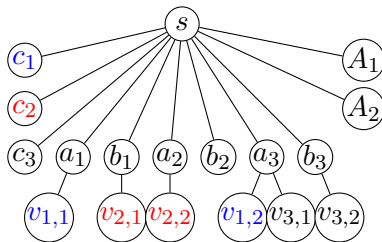
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned} \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



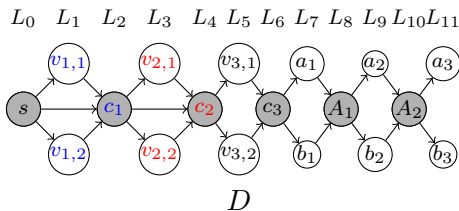
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



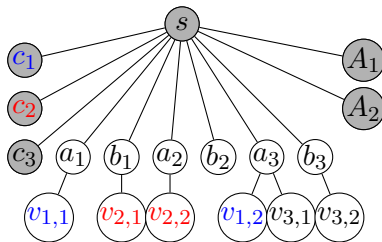
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned}\mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\}\end{aligned}$$



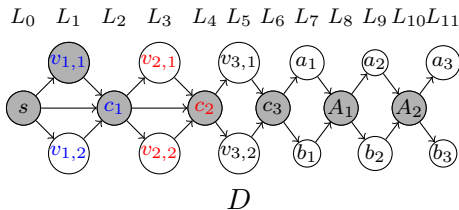
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



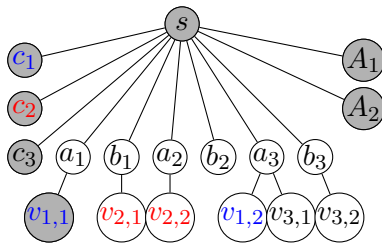
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned} \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



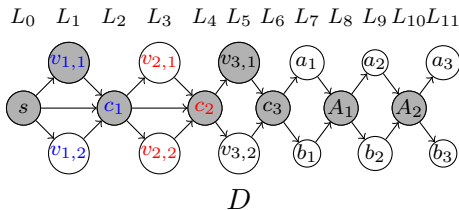
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



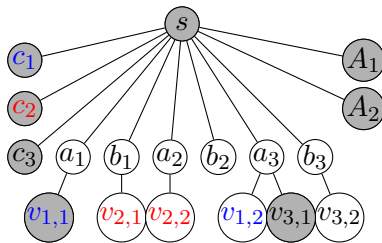
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned} \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



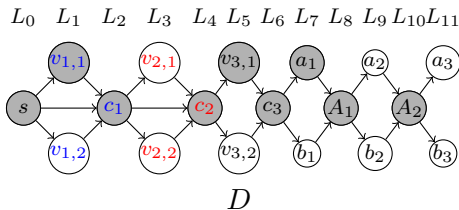
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



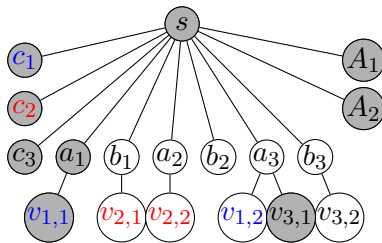
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned} \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



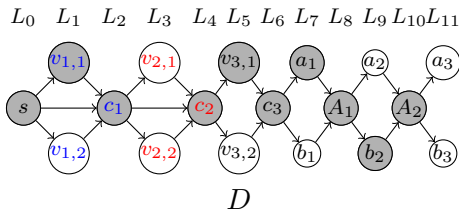
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



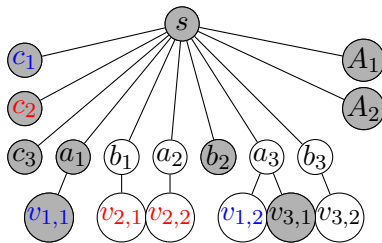
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned} \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



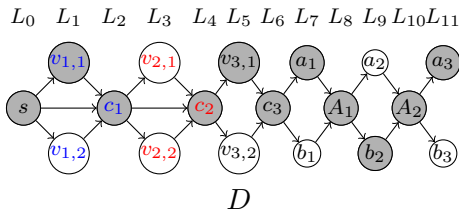
**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



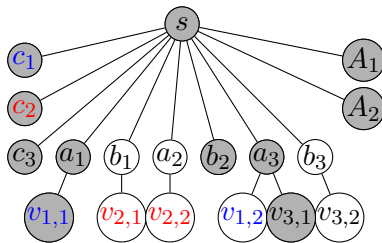
# Complexity of the ONE-TO-ONE SKEWGRAM problem

Assignment :  $\{x_1 = \text{true}, x_2 = \text{false}, x_3 = \text{true}\}$

$$\begin{aligned} \mathcal{C} &= \{C_1, C_2\} \\ C_1 &= (x_1 \vee x_3), \quad C_2 = (\overline{x_1} \vee x_2) \\ \mathcal{X} &= \{x_1, x_2, x_3\} \end{aligned}$$



**Property :** There is an assignment that satisfies at least  $k$  clauses iff there is a  $(D, G)$ -consistent path of at least  $10 + k$  vertices.



# Two algorithms for ONE-TO-ONE SKEWGRAM

- Heuristic called ALGOH.
- Exact exponential-time algorithm called ALGOBB using the branch and bound method.

# Two algorithms for ONE-TO-ONE SKEWGRAM

- Heuristic called ALGOH.
- Exact exponential-time algorithm called ALGOBB using the branch and bound method.

**Input :**  $D = (V, A)$ ,  $G = (V, E)$ , an arc  $xy \in A$ .

**Output :** A longest  $(D, G)$ -consistent path going through  $xy$ .

To solve ONE-TO-ONE SKEWGRAM, an execution is needed for every arc  $xy$  of  $D$ .

# Heuristic ALGOH : Part 1

Rule 1 : **Connectivity in  $D$  (reachability)**;

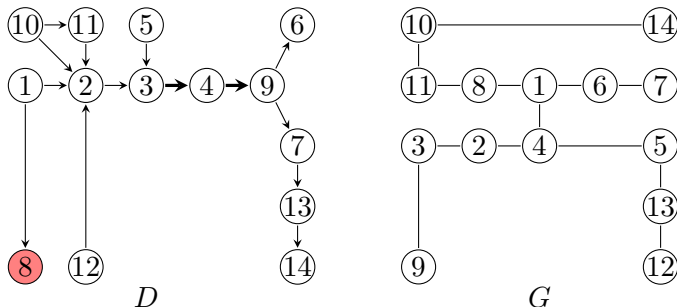


FIGURE: **Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .**

# Heuristic ALGOH : Part 1

Rule 1 : **Connectivity in  $D$  (reachability)** ;

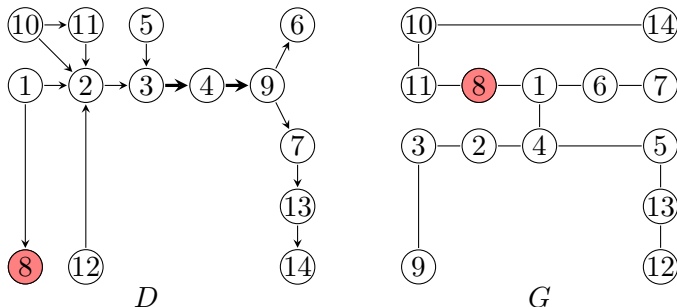


FIGURE: **Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .**

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : **Connectivity in  $G$  (bridges)**;

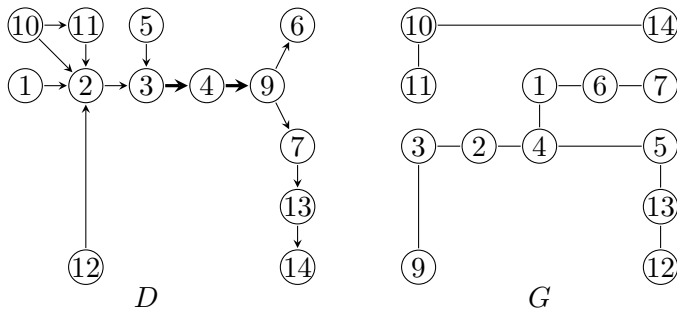


FIGURE: Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : **Connectivity in  $G$  (bridges)**;

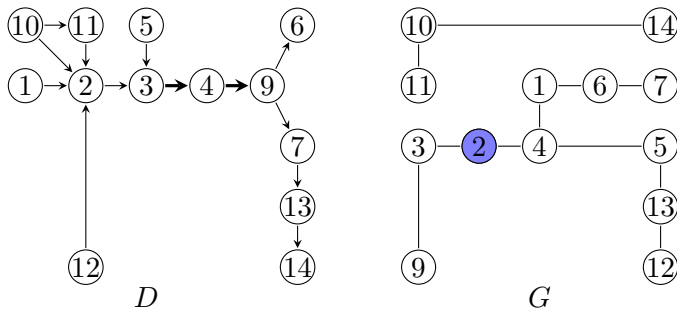


FIGURE: **Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .**

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : **Connectivity in  $G$  (bridges)**;

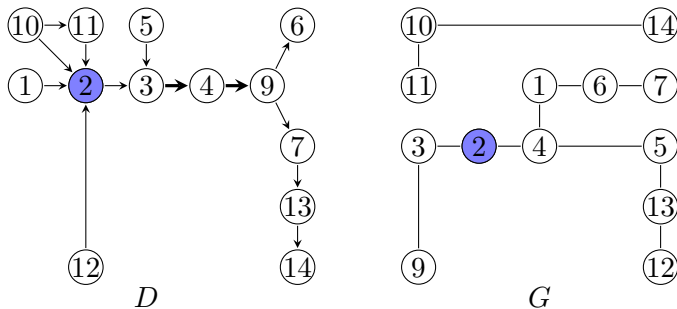


FIGURE: **Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .**

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : **Connectivity in  $G$  (bridges)**;

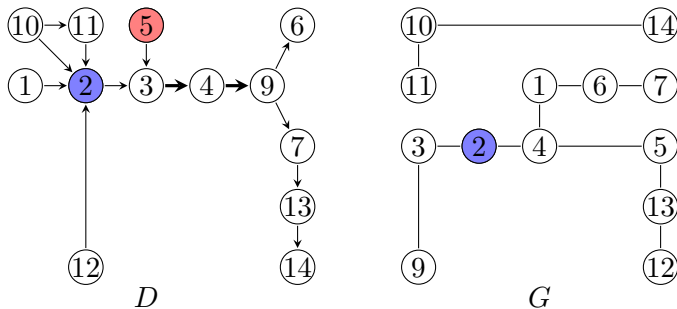


FIGURE: **Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .**

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : **Connectivity in  $G$  (bridges)**;

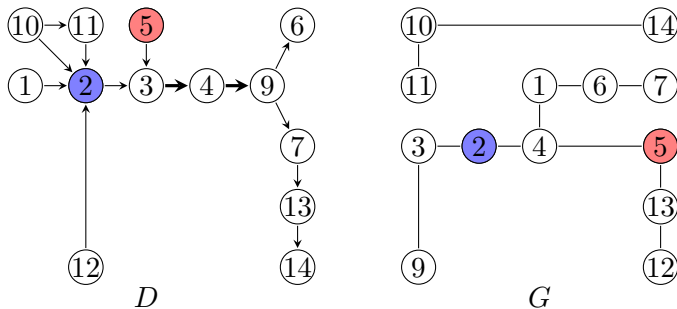


FIGURE: **Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .**

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : Connectivity in  $G$  (bridges);

Rule 3 : **Connectivity in  $D$  and  $G$  simultaneously (common connected components).**

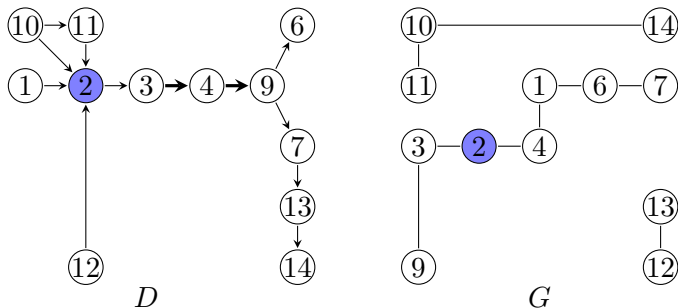


FIGURE: Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : Connectivity in  $G$  (bridges);

Rule 3 : **Connectivity in  $D$  and  $G$  simultaneously (common connected components).**

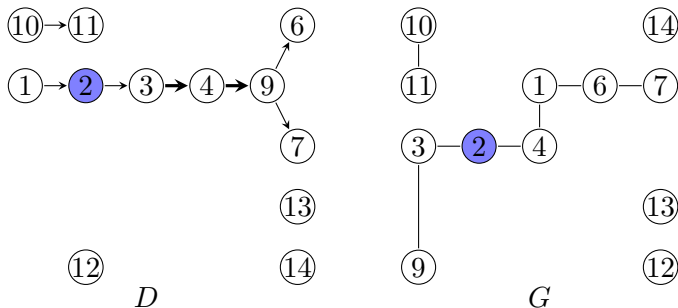


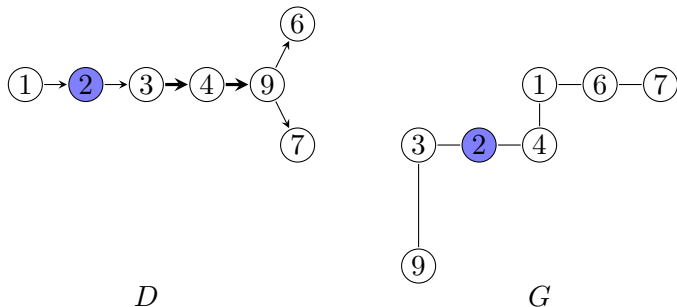
FIGURE: Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .

# Heuristic ALGOH : Part 1 (cover set of a path)

Rule 1 : Connectivity in  $D$  (reachability);

Rule 2 : Connectivity in  $G$  (bridges);

Rule 3 : **Connectivity in  $D$  and  $G$  simultaneously (common connected components).**

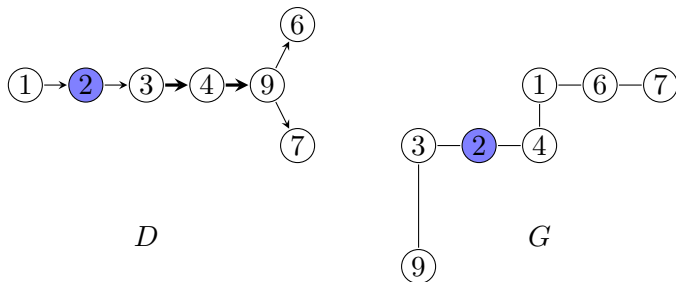


**FIGURE:** Computation of the cover set of the path  $3 \rightarrow 4 \rightarrow 9$ .

$$\text{COVERSET}(3 \rightarrow 4 \rightarrow 9) = \{1, 2, 3, 4, 6, 7, 9\}$$

# Heuristic ALGOH : Part 1 (cover set of a path)

**Rule 3 : Connectivity in  $D$  and  $G$  simultaneously (common connected components).**



$$\text{COVERSET}(3 \rightarrow 4 \rightarrow 9) = \{1, 2, 3, 4, 6, 7, 9\}$$

**Property** : Let  $p$  be a path in  $D$ . If  $D[\text{COVERSET}(p)]$  is Hamiltonian then its Hamiltonian path is the longest  $(D, G)$ -consistent path going through  $p$ .

**Scoring function** : for each directed path  $p$ ,  $value(p)$  is the number of vertices of the longest path in  $D[COVERSET(p)]$ .

- $ALGOH(D, G, x \rightarrow y)$ 
  - 1  $cp := x \rightarrow y$  /\* current path \*/
  - 2  $cs := \emptyset$  /\* current solution \*/
  - 3  $S := COVERSET(cp); G := G[S]; D := D[S];$
  - 4 We compute the list  $L$  of paths obtained by extending  $cp$  by one vertex ;
  - 5 We compute  $p_{max}$  s.t.  $value(p_{max}) = \max\{value(p) : p \in L\}$ .
  - 6 We compute  $ph_{max}$  s.t.  $value(ph_{max}) = \max\{value(p) : p \in L \text{ and } D[COVERSET(p)] \text{ is Hamiltonian}\}$ .
  - 7  $update(cs, ph_{max})$  ;
  - 8  $cp := p_{max}$  ; **return to step 3** ;

**End of the process** : when  $L = \emptyset$  or  $value(p_{max}) \leq |V(cs)|$ .

# Performances of ALGOH

- $|\text{ALGOH}|$  is the number of vertices of a solution found by the heuristic ALGOH.
- $|\text{ALGOBB}|$  is the number of vertices of a solution found by ALGOBB.
- Performance  $\rho = \frac{|\text{ALGOH}|}{|\text{ALGOBB}|}$ .

# Performances of ALGOH on Erdős-Rényi graphs

We vary two parameters :

- The number  $n$  of vertices of  $D$  and  $G$  (in the range 20, 30, 40, 50, 60) ;
- The probability  $p$  that an edge between any given two vertices exists (in the range 0.05, 0.1, 0.15, 0.2).

# Performances of ALGOH on Erdős-Rényi graphs

We vary two parameters :

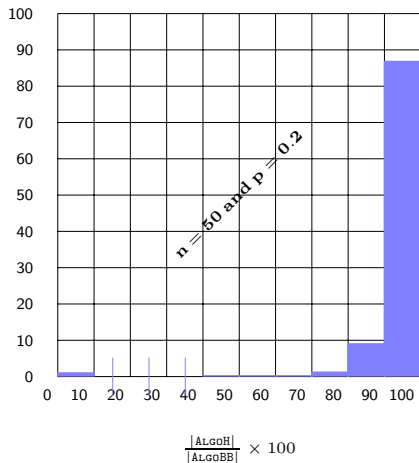
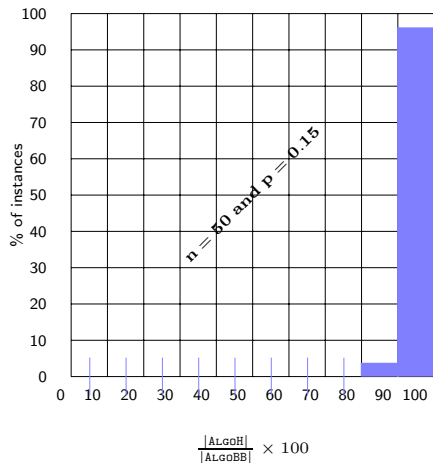
- The number  $n$  of vertices of  $D$  and  $G$  (in the range 20, 30, 40, 50, 60) ;
- The probability  $p$  that an edge between any given two vertices exists (in the range 0.05, 0.1, 0.15, 0.2).

For a fixed  $n$  and  $p$  we generated 500 instances :

- 100 couples  $(D, G)$  ;
- 5 arcs per couple ;

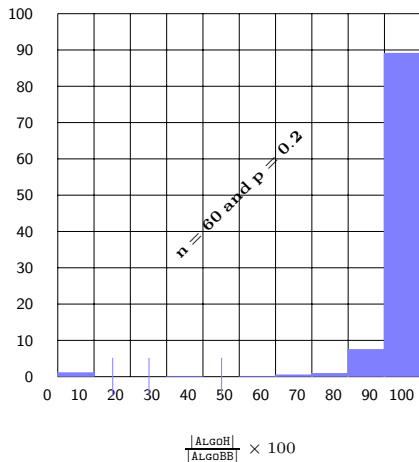
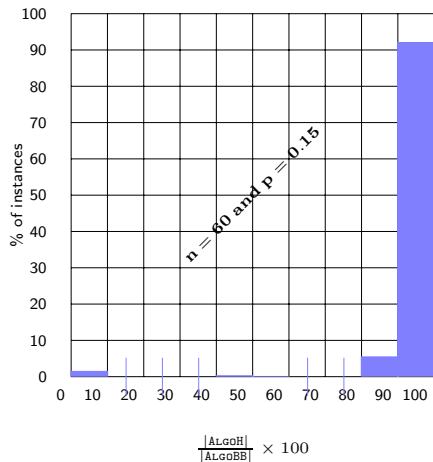
# Performances of ALGOH on Erdős-Rényi graphs

$n = 50$



# Performances of ALGOH on Erdős-Rényi graphs

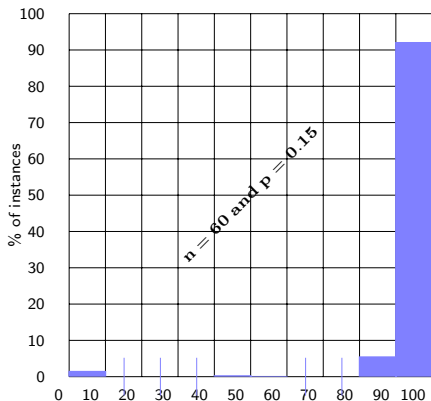
$n = 60$



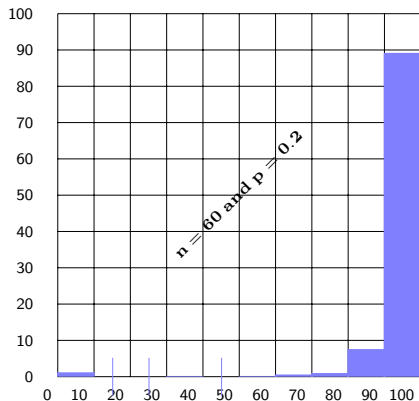
# Performances of ALGOH on Erdős-Rényi graphs

$$n = 60$$

Speed-up : ALGOH was 11 times faster than ALGOBB



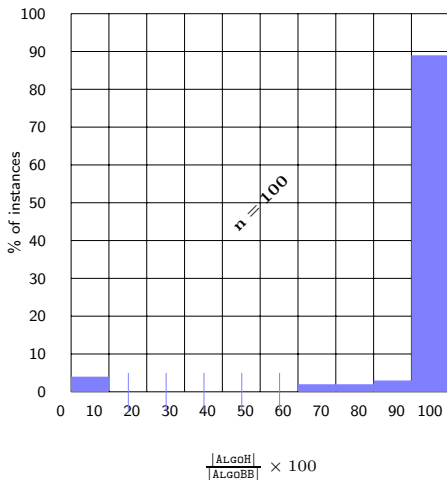
$$\frac{|\text{ALGOH}|}{|\text{ALGOBB}|} \times 100$$



$$\frac{|\text{ALGOH}|}{|\text{ALGOBB}|} \times 100$$

# Performances of ALGOH on scale-free graphs

- Generation of 100 couples  $(D, G)$  of 100 vertices using the public toolkit *NGCE* (<http://ngce.sourceforge.net/>) ;
- 10 arcs per couple.



- ONE-TO-ONE SKEWGRAM : New approach to compare heterogeneous biological networks
- ONE-TO-ONE SKEWGRAM is NP-complete.
- Two algorithms for ONE-TO-ONE SKEWGRAM.

- Investigate the complexity of ONE-TO-ONE SKEWGRAM in terms of :
  - Approximability (on specific graph classes)
  - Fixed parameter algorithms
- Consider inexact variants of the problem.

