On the Algorithmic Effectiveness of Digraph Decompositions and Complexity Measures

Michael Lampis, Georgia Kaouri and Valia Mitsou

Graph decompositions

- Treewidth (by Robertson and Seymour) is the most well-known and widely studied graph decomposition.
- Treewidth describes how much a graph looks like a tree.
- A large number of graph problems can be solved efficiently (in FPT time) for low treewidth. (Courcelle's theorem)
- Many equivalent definitions (e.g. cops-and-robber games, minimum fill-in, elimination orderings).

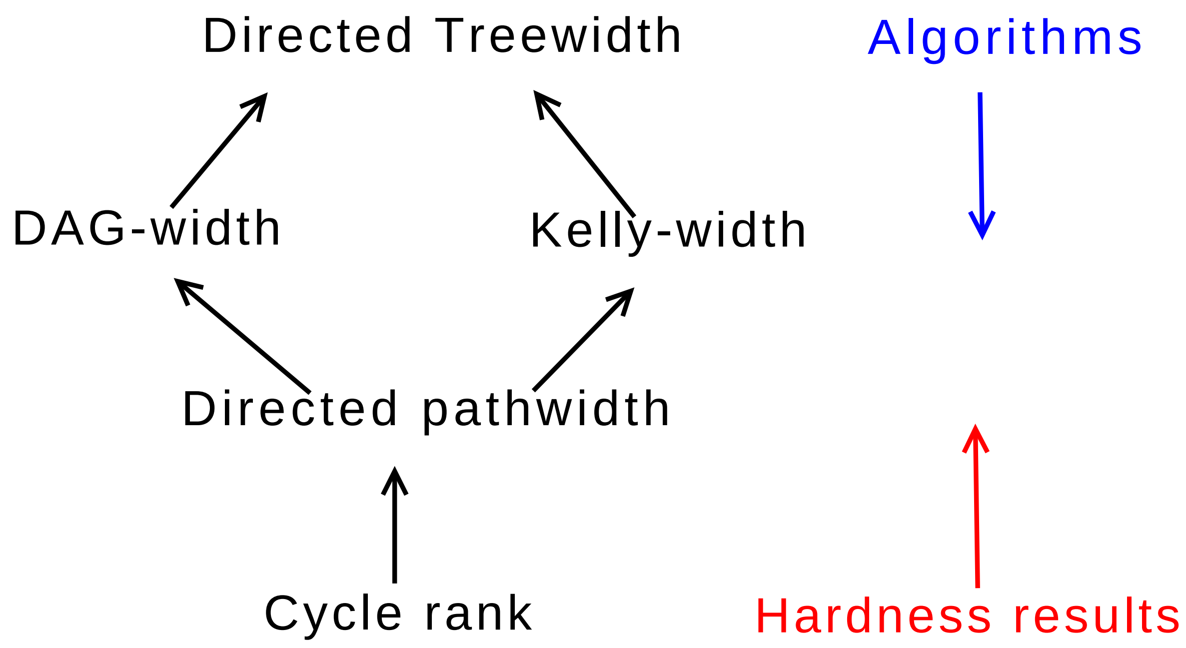
Digraph decompositions

- Treewidth is generally considered the right measure for undirected graphs.
- Treewidth can usually be employed for digraph problems as well: take the tree decomposition of the underlying undirected graph.
- This solution is not perfect. E.g. ignoring the direction of edges on a DAG may lead to a clique (large treewidth). But the problem may be trivial on DAGs (e.g. Hamiltonian Cycle).

Digraph decompositions

- What is the right treewidth analogue for digraphs?
- Directed treewidth [Johnson et al., 2001]
- DAG-width [Obdrzálek, 2006]
- Kelly-width [Hunter and Kreutzer, 2007]

Relations between measures



Known results

- An $O(n^k)$ algorithm for Hamiltonian Cycle where k is the directed treewidth. [Johnson et al., 2001]
- An $O(n^k)$ algorithm for parity games where k is the DAG-width [Obdrzálek, 2006]
- A $O(n^k)$ algorithms for both where k is the kelly-width [Hunter and Kreutzer, 2007]
- No FPT algorithms are known!

Our results

- MaxDiCut is NP-complete when restricted to DAGs
- Hamiltonian Cycle is $W[2]$-hard when the parameter is the cycle rank of the input graph.

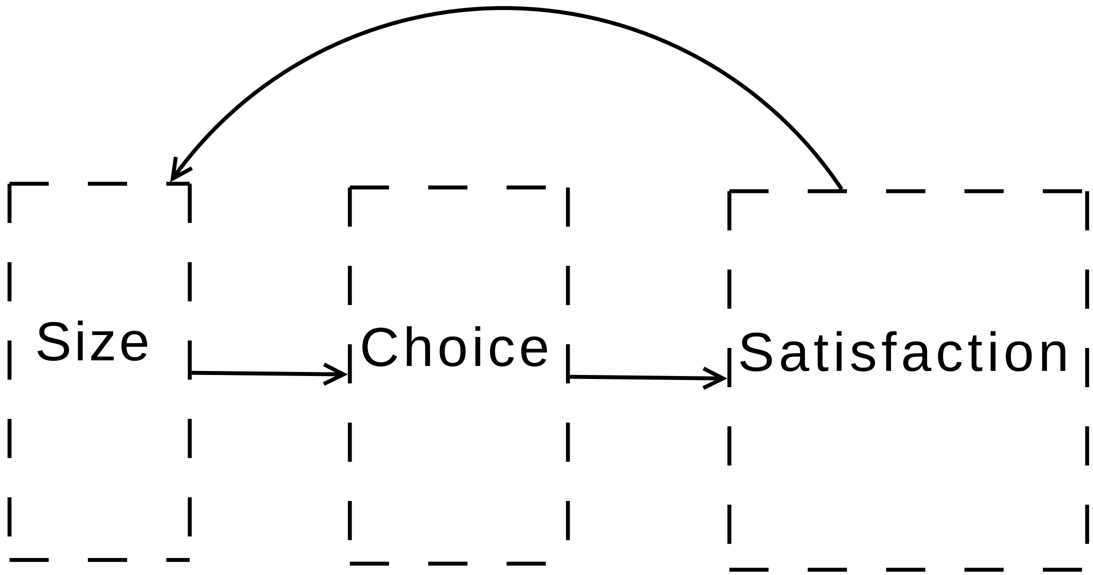
Implication:

- Both problems are intractable for all considered complexity measures.

Hamiltonian cycle

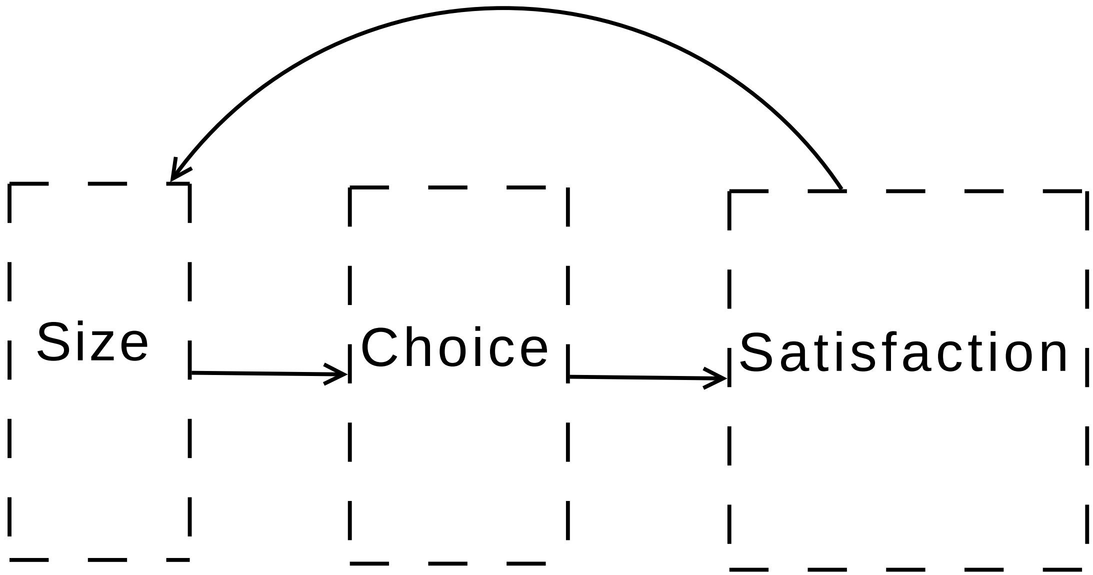
- Reduction from Dominating Set.
- We are given an undirected graph G and a number k. Does G have a dominating set of size k?
- Construct a digraph G'. G' will be Hamiltonian iff G has a dominating set of size k.
- G' will have small width (a function of k) under all definitions.

The reduction



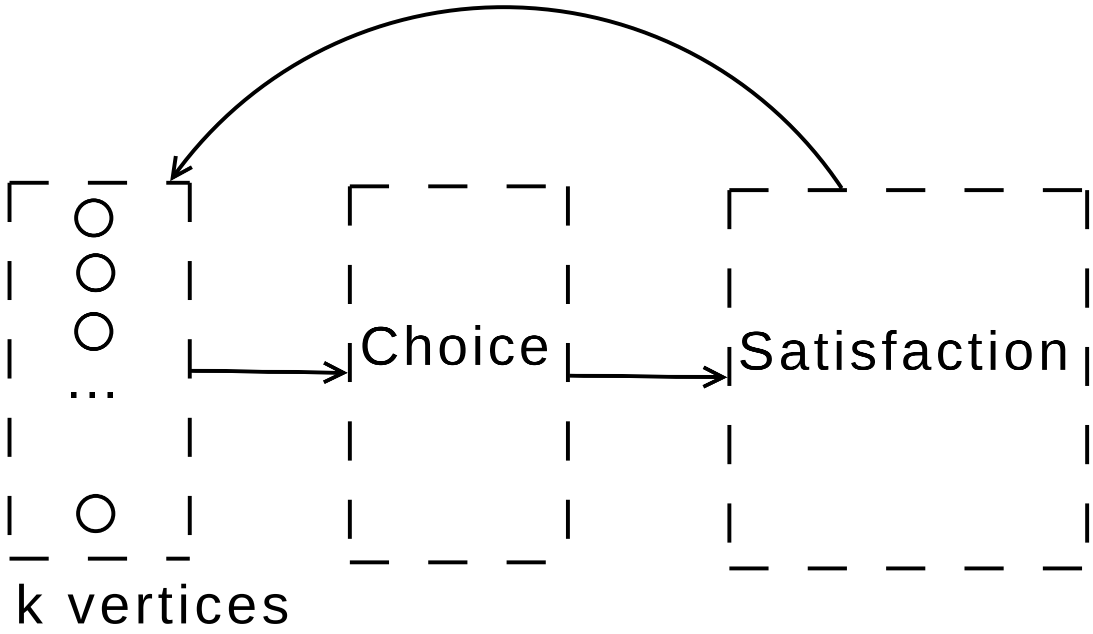
- Construction has three parts.

The reduction



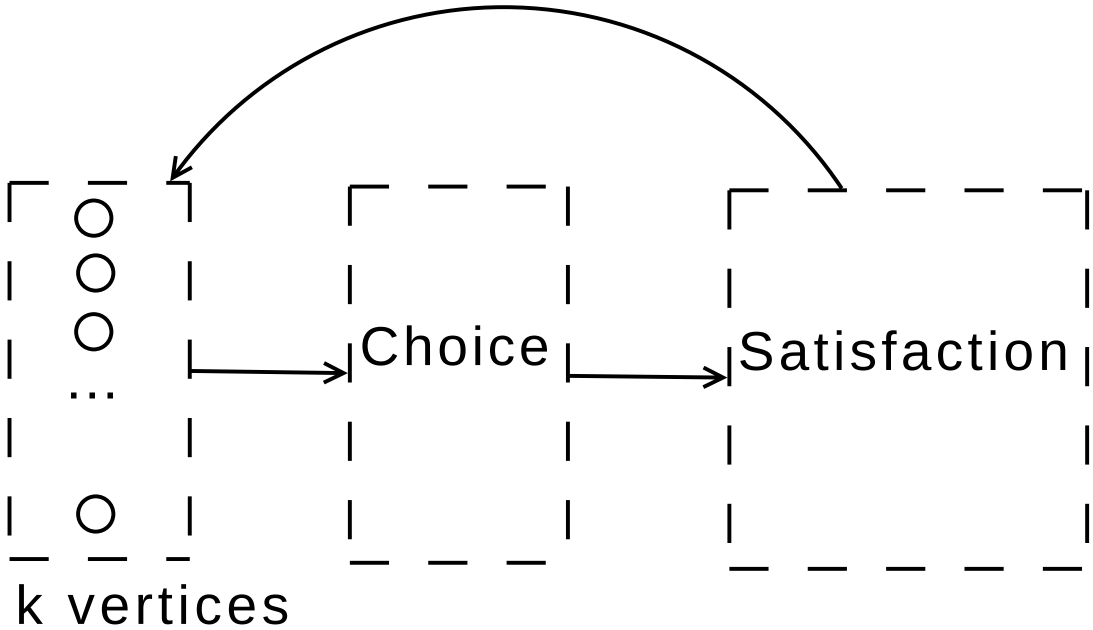
- Construction has three parts.
- The first part makes sure that G' can only be Hamiltonian iff I pick a dominating set of size k.

The reduction



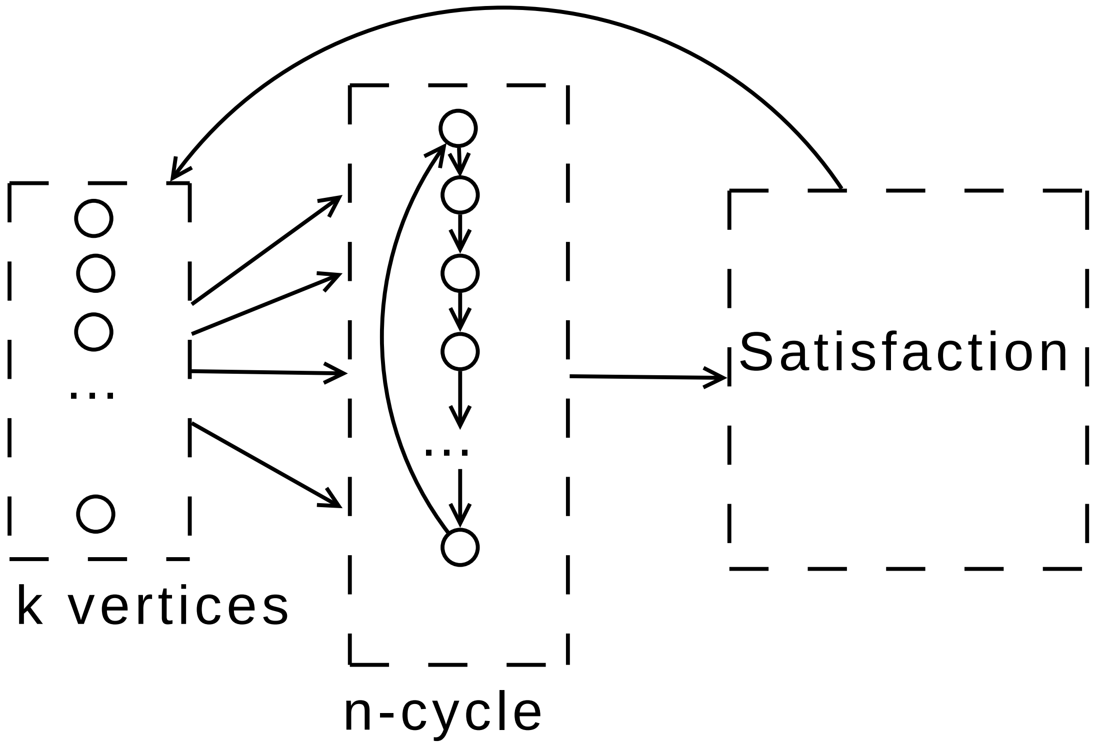
- This is accomplished by using exactly k vertices.

The reduction



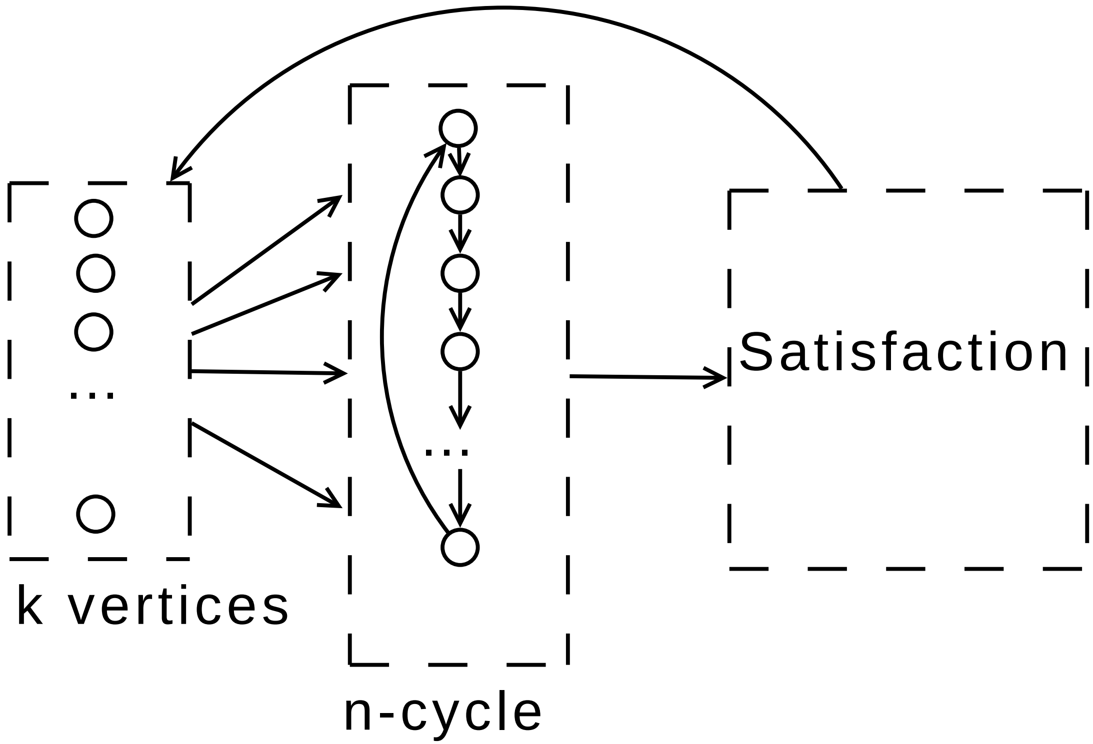
- The second part represents a choice of dominating set.

The reduction



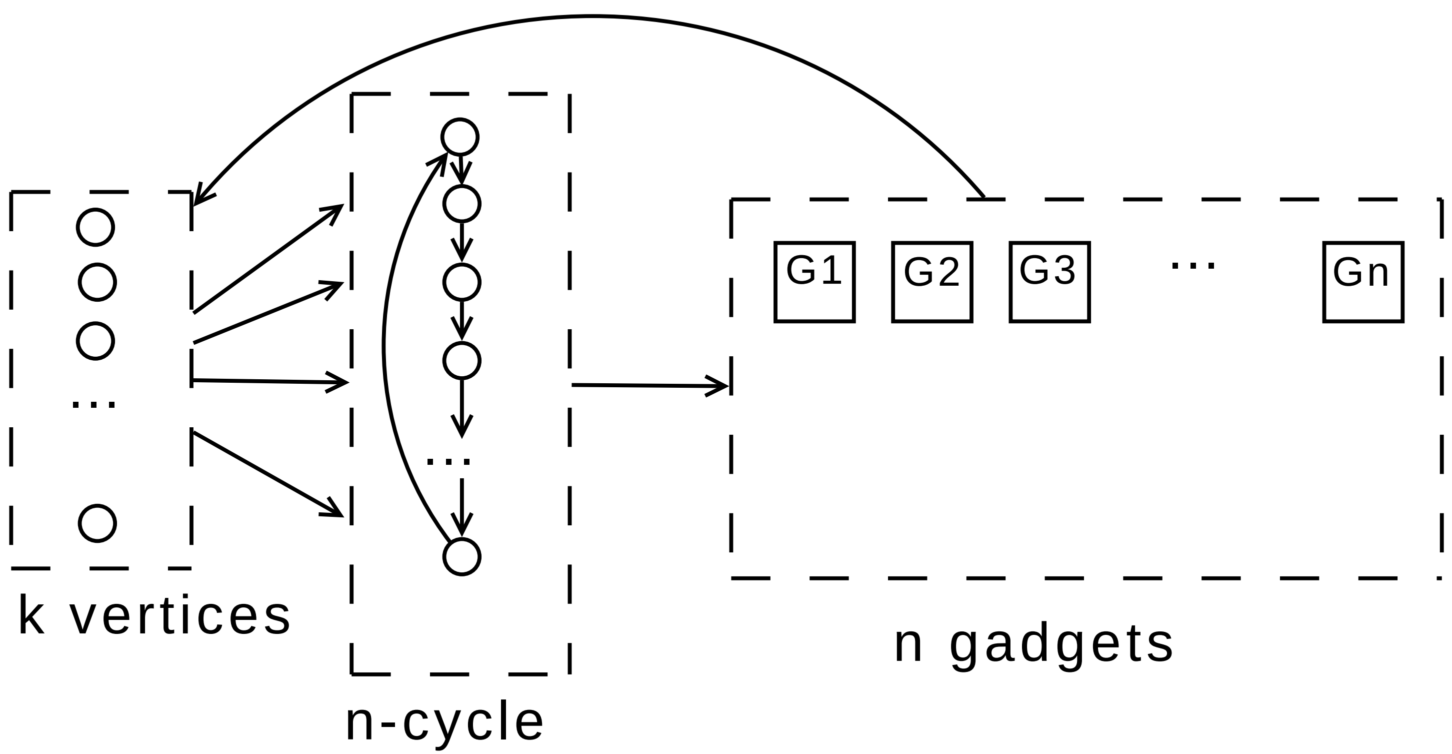
- This is accomplished by using an n-cycle.
- The exit points from the cycle correspond to vertices in the dominating set.

The reduction



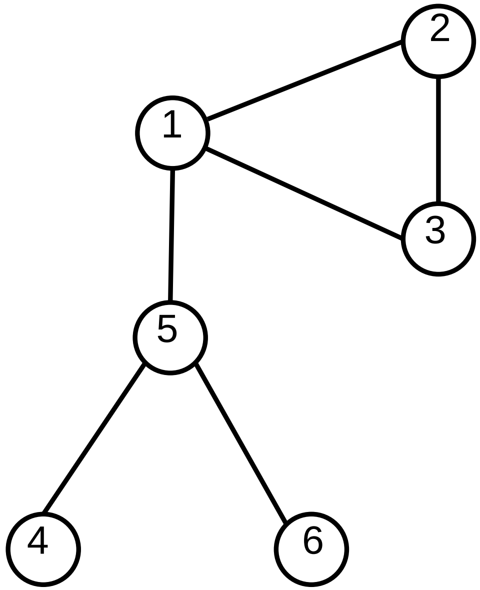
- Finally, the third part makes sure that the choice is indeed a dominating set.

The reduction



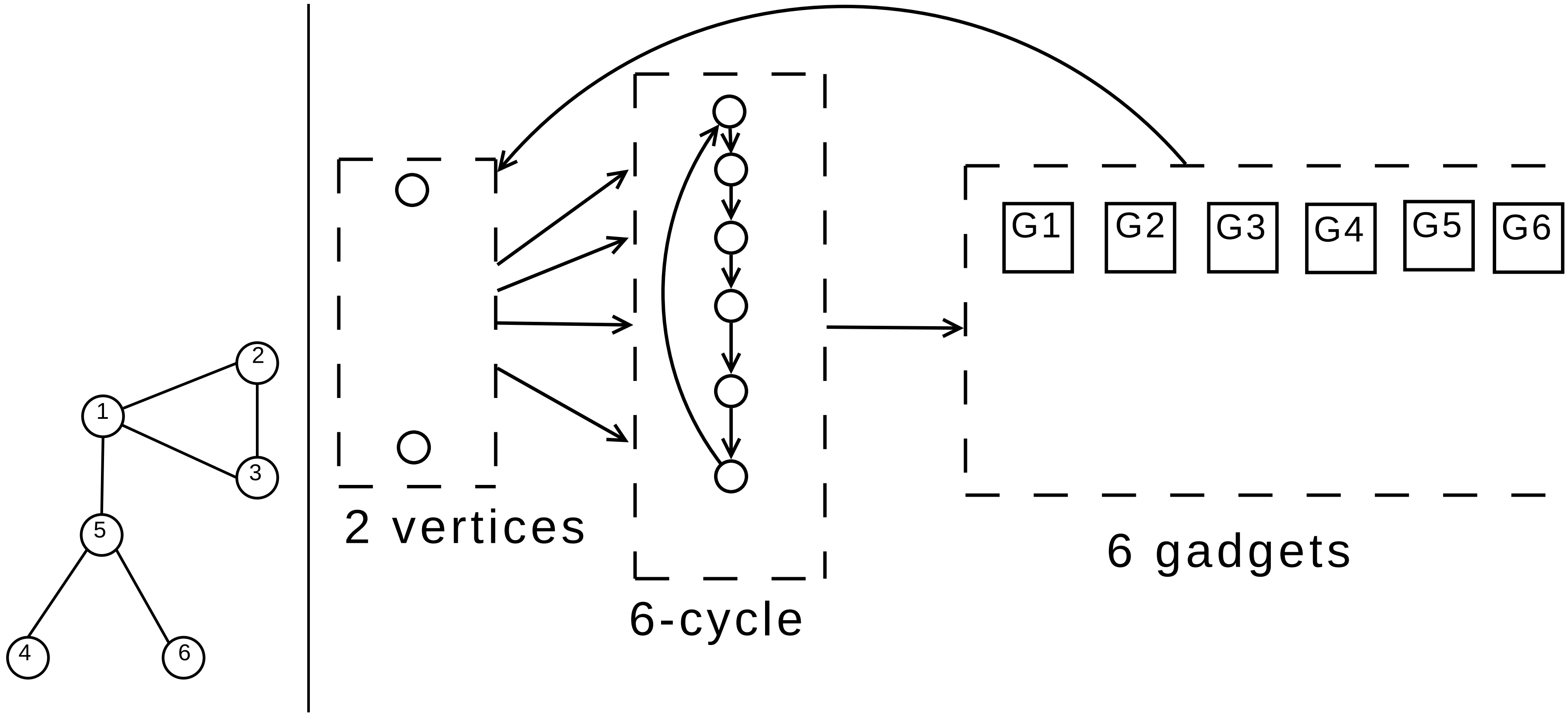
- This is accomplished by placing a gadget to check domination for each vertex of G.

Example

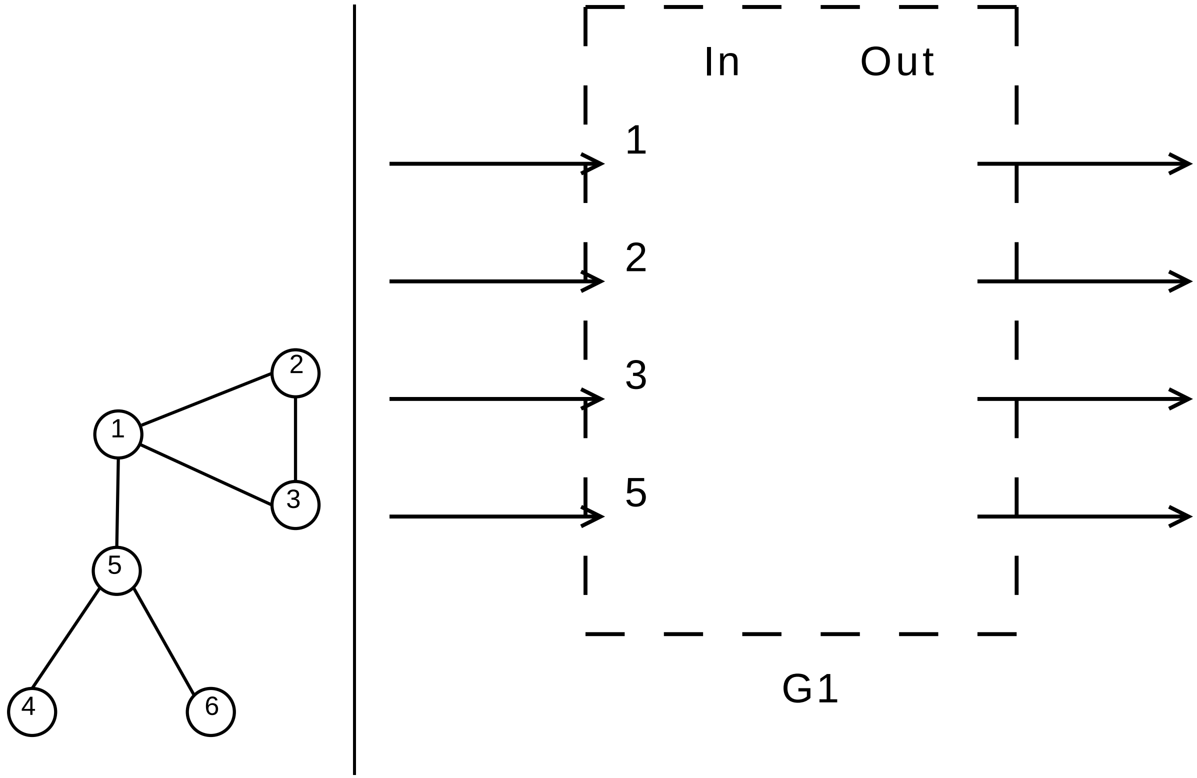


Suppose that we want to see if this graph has a dominating set of size 2.

Example

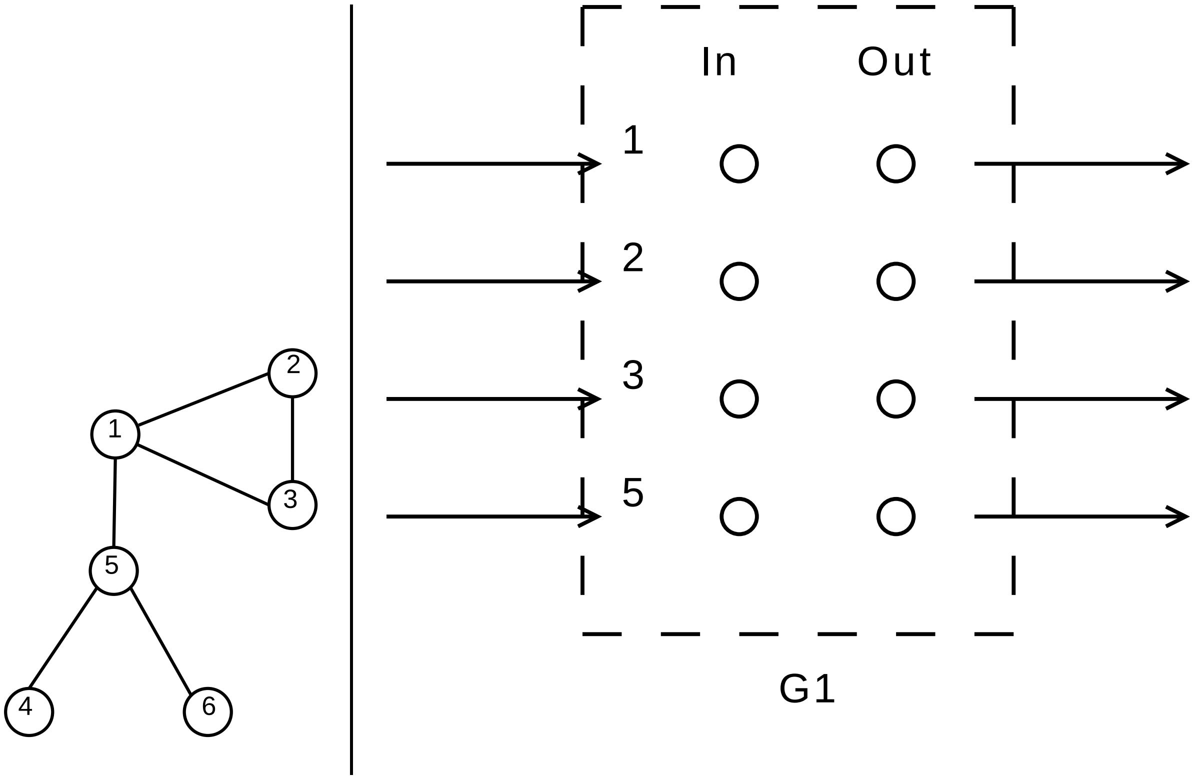


The satisfaction gadget



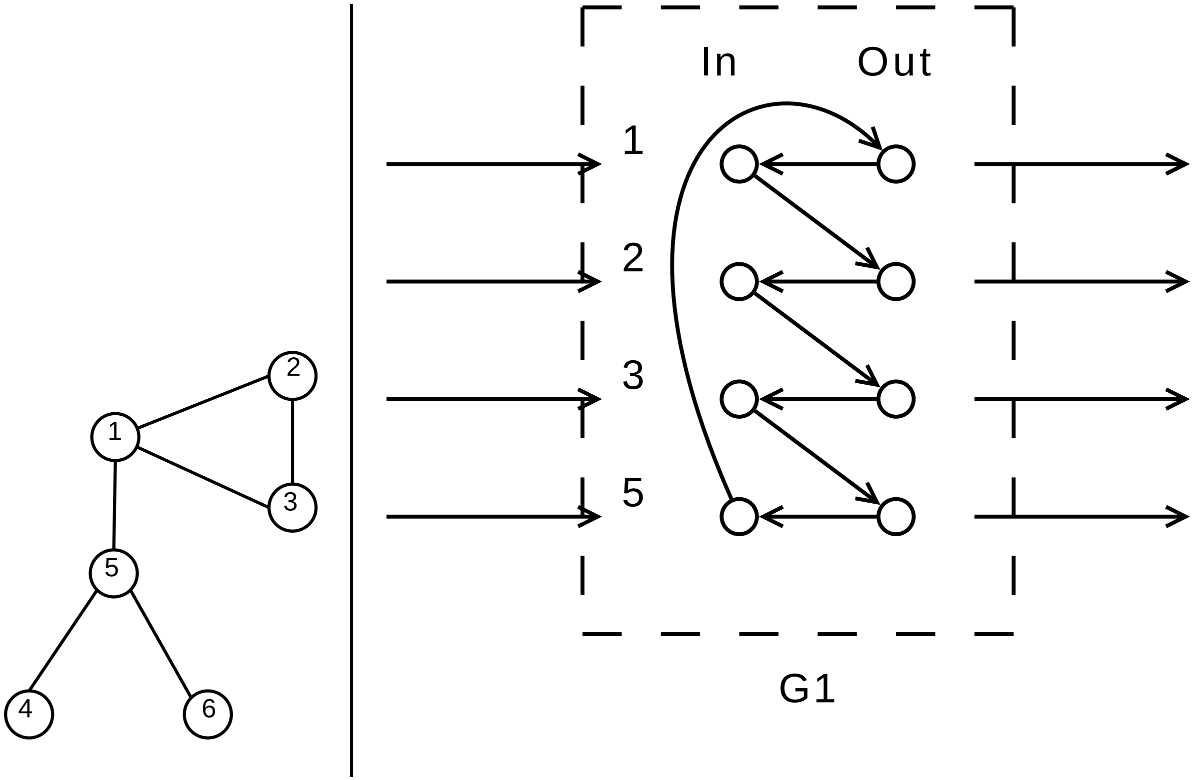
- Vertex 1 can be dominated in 4 ways: by picking 1,2,3 or 5.
- The gadget G_1 will have 4 inputs and 4 outputs.

The satisfaction gadget



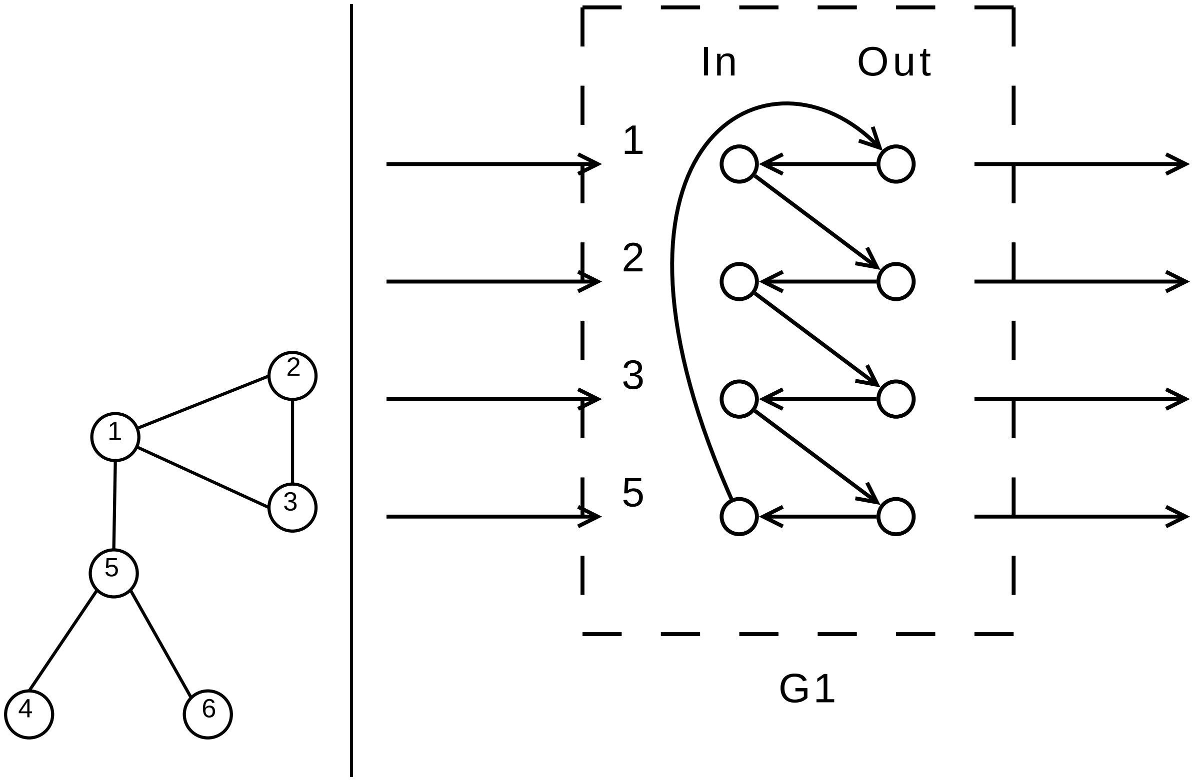
- For each input/output point use one vertex.

The satisfaction gadget



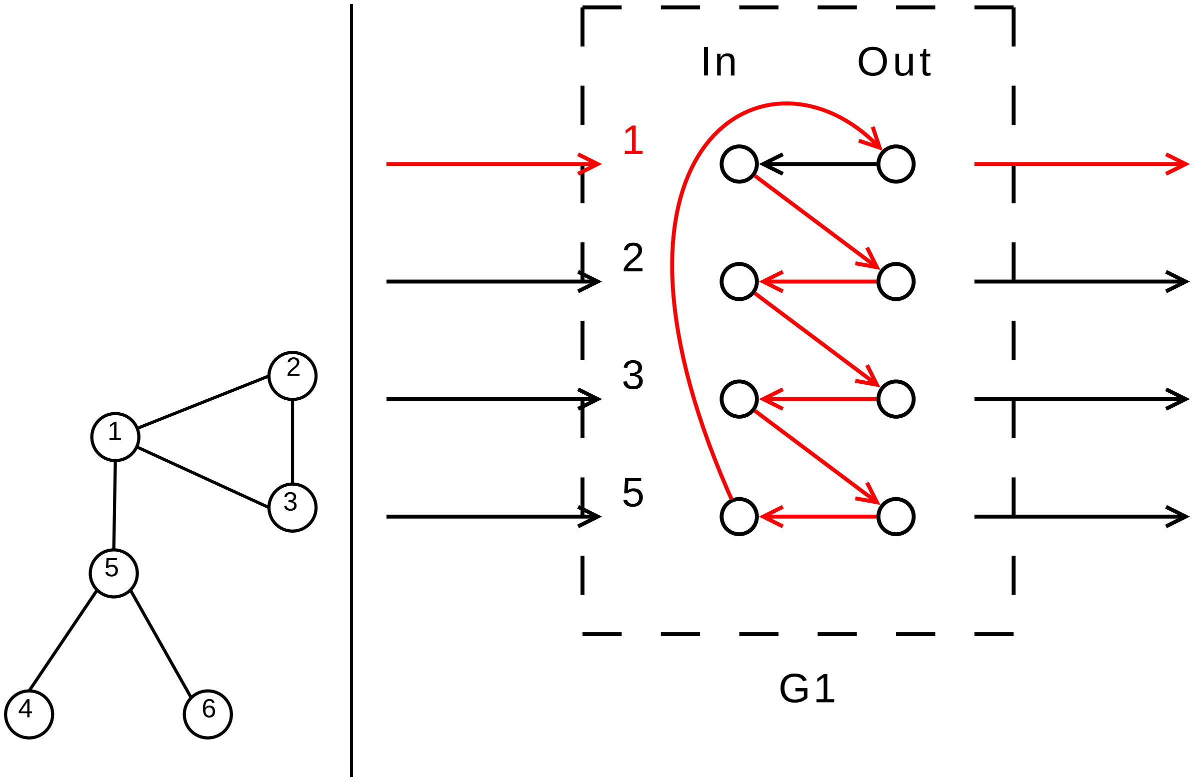
- Connect them in a directed cycle.

The satisfaction gadget



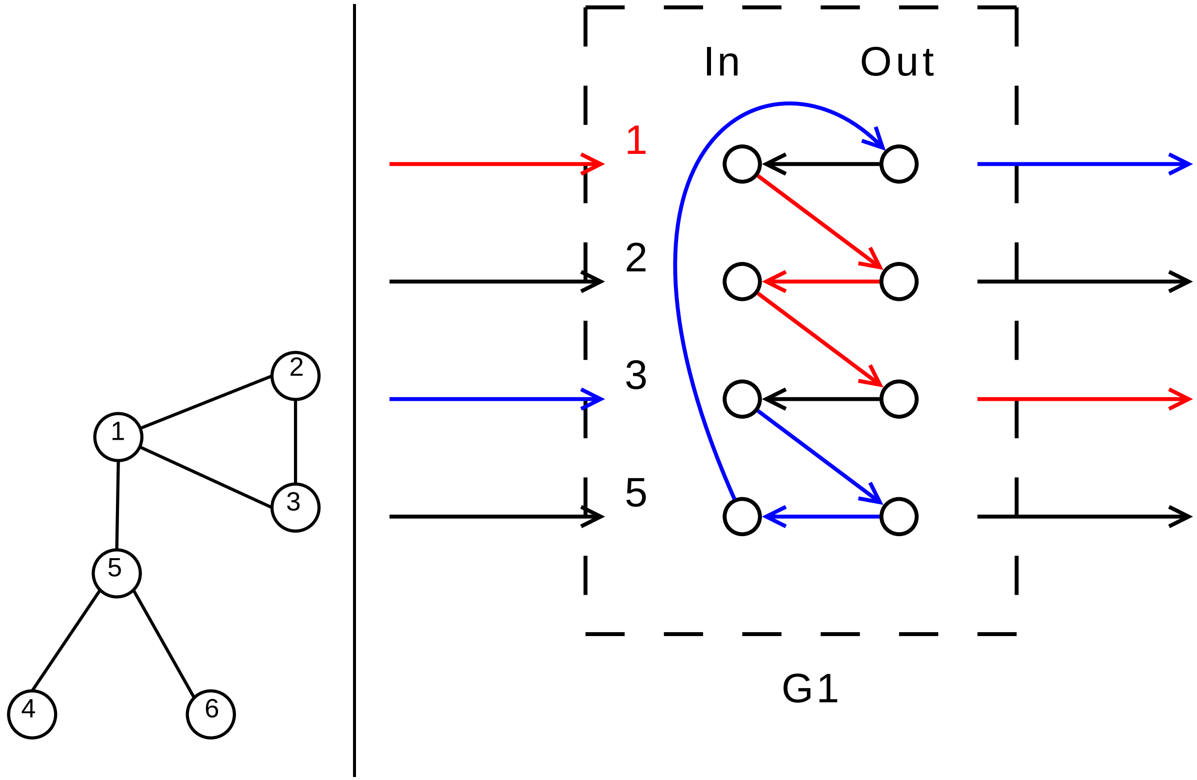
- This makes any Hamiltonian tour of the gadget exit from the same set of outputs it entered.

The satisfaction gadget



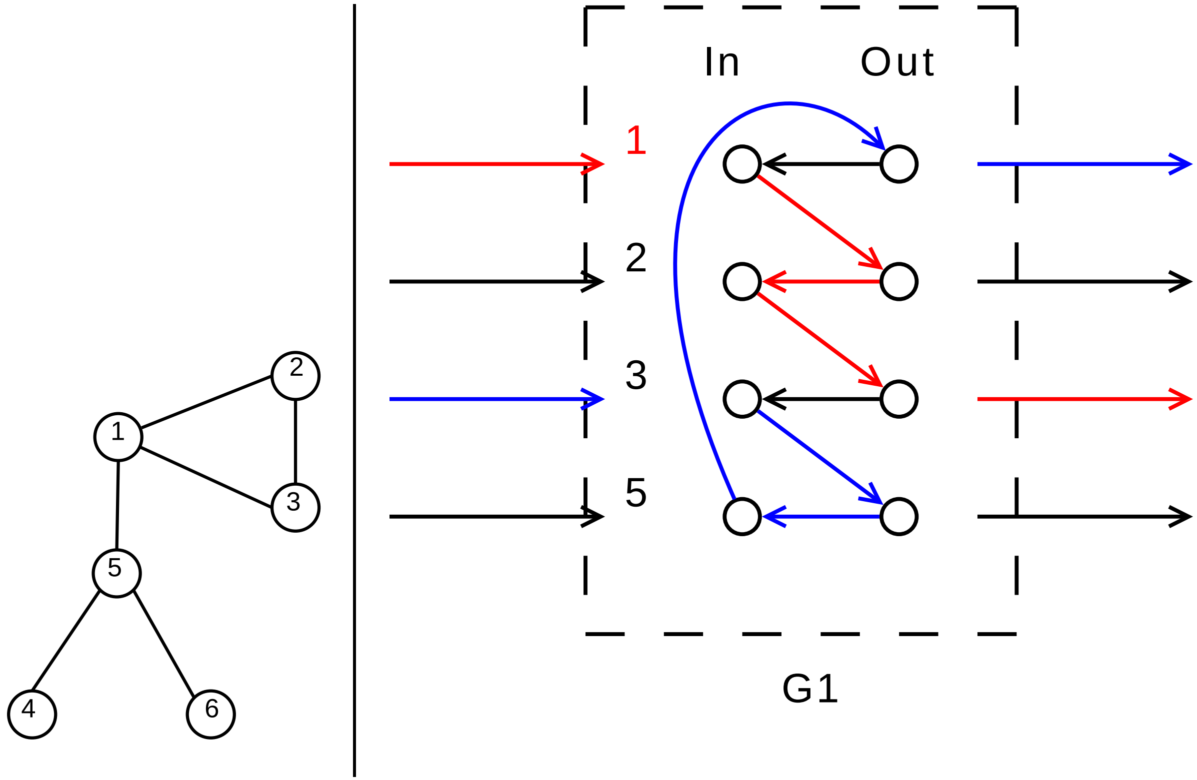
- Example: Entering through input point 1.

The satisfaction gadget



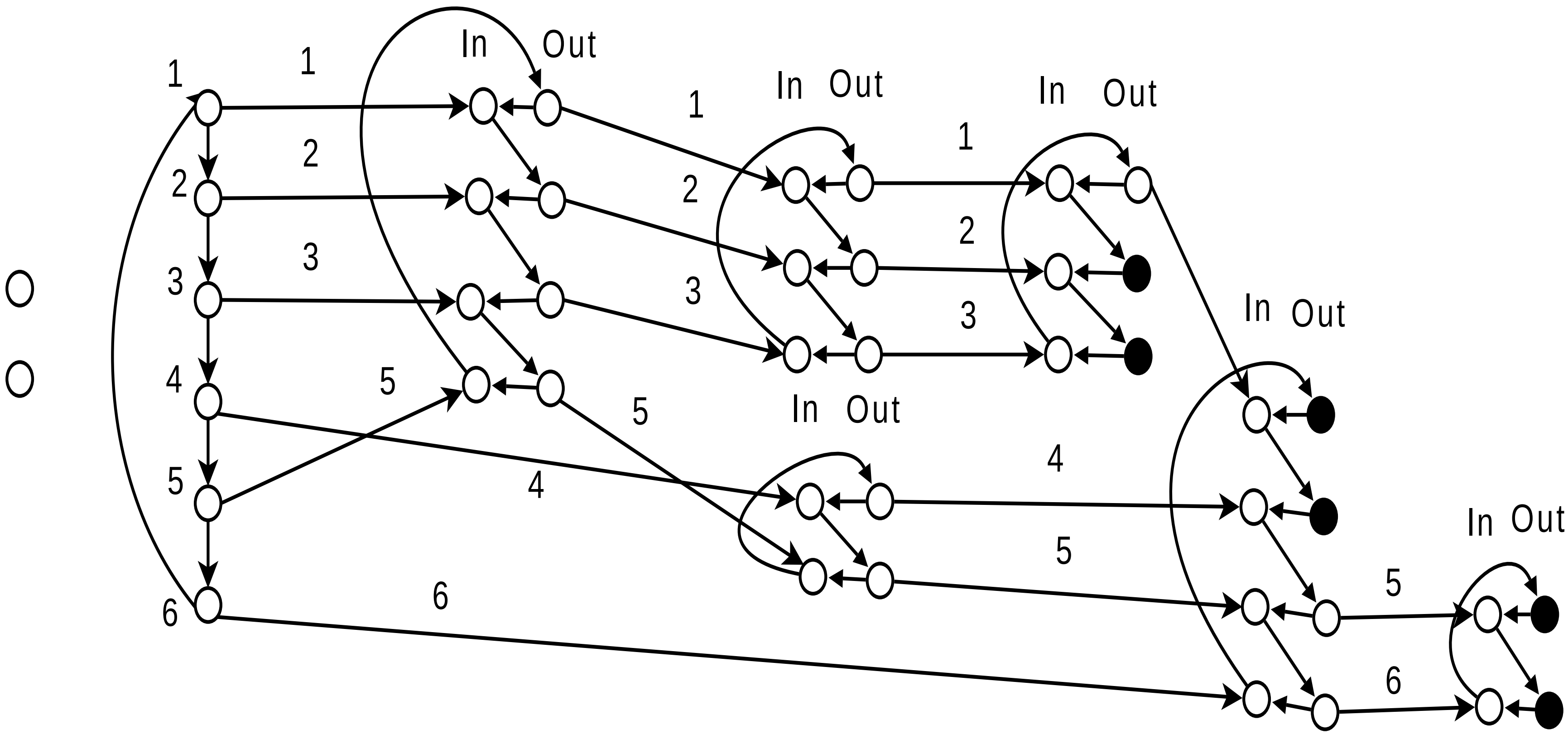
- Entering through input points 1 and 3.

The satisfaction gadget



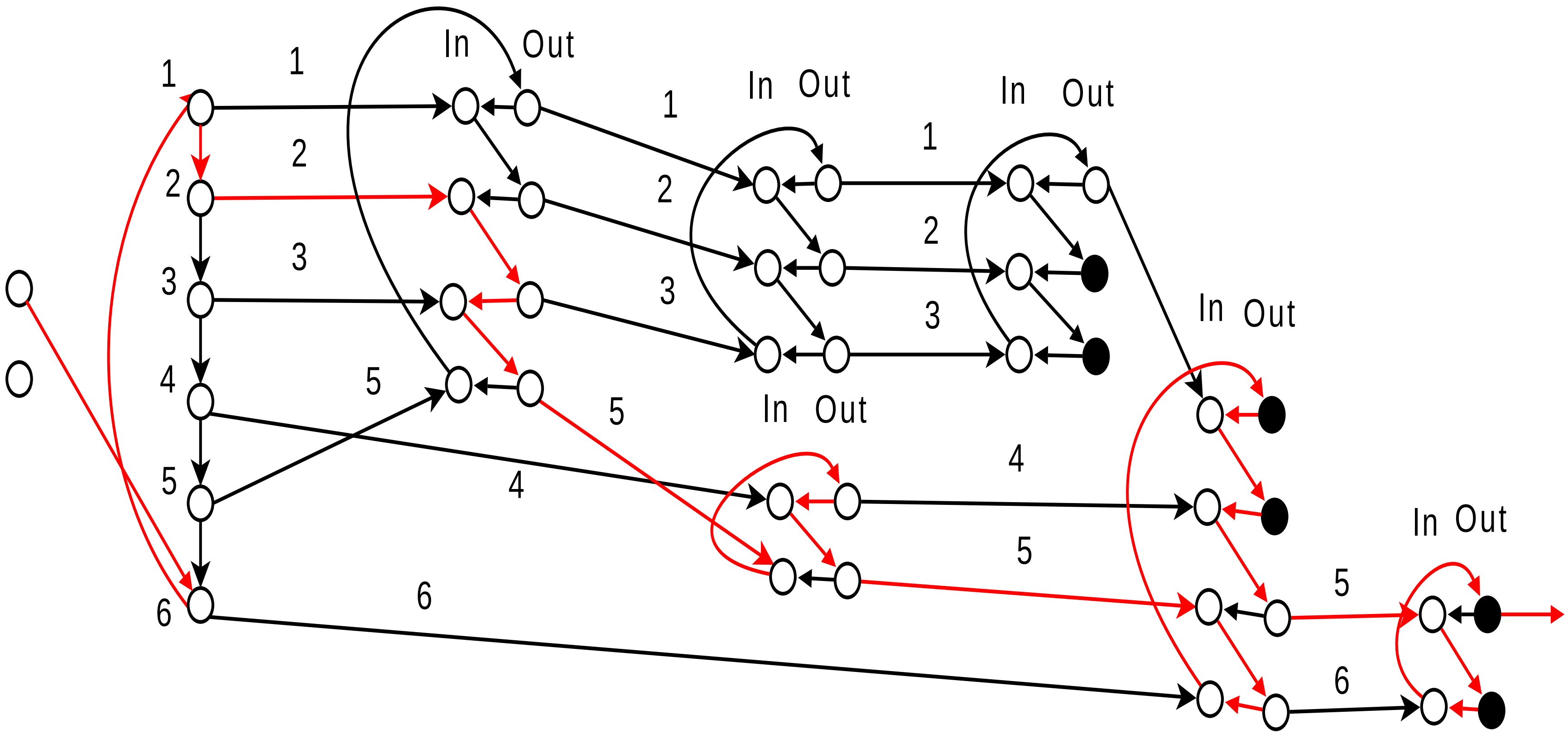
- Why this is important: The gadgets maintain the choices made in the second part of the graph.

Full example



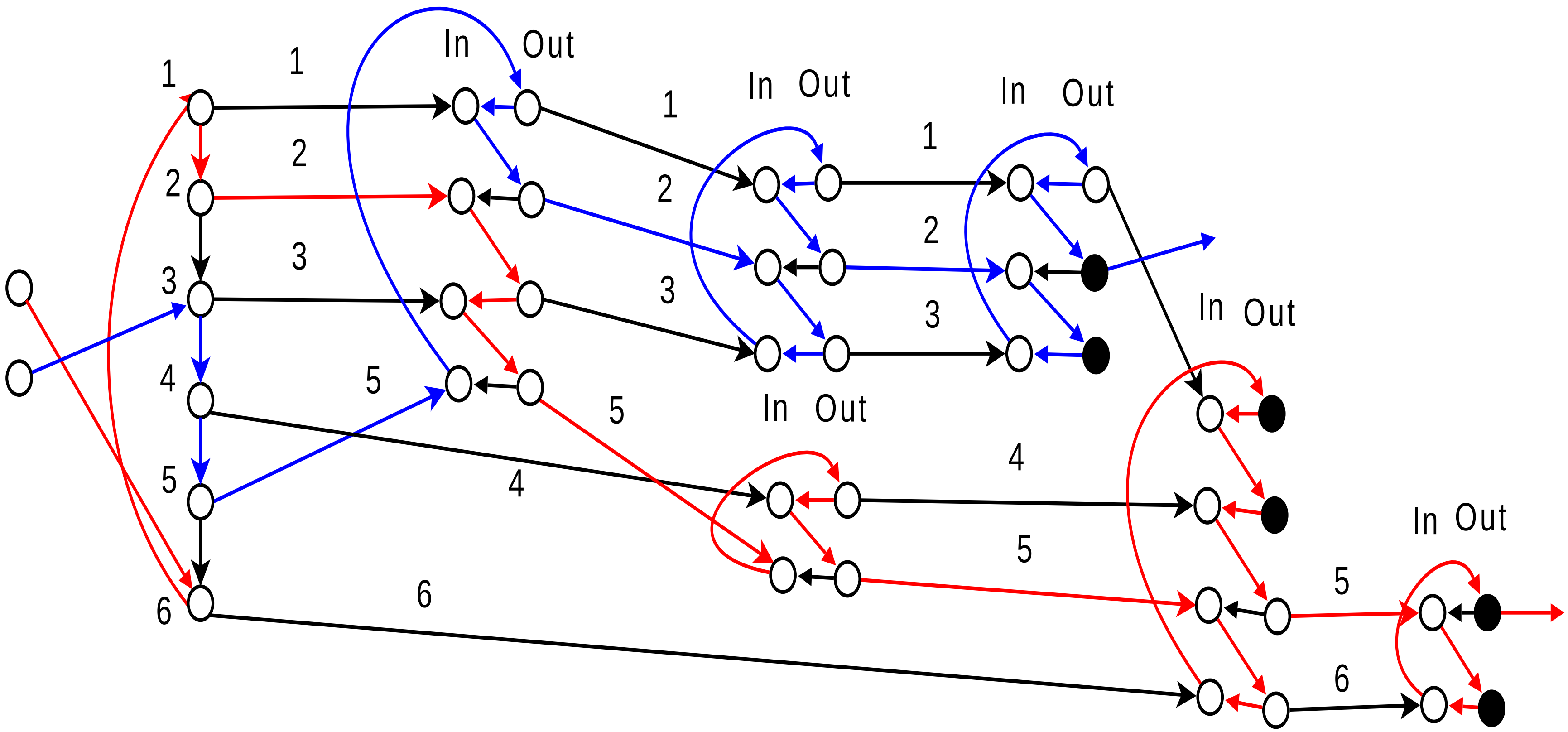
- Full construction.
- G' has Hamiltonian cycle for dominating set $2, 5$.

Full example



- Full construction.
- G' has Hamiltonian cycle for dominating set $2, 5$.

Full example



- Full construction.
- G' has Hamiltonian cycle for dominating set $2, 5$.

Completing the proof

- What remains is to show that G' has small width.
- If we remove the k vertices of the first part, we are left with an ordered set of $n + 1$ directed cycles.
- Each of these has small width.

Summary of results

	Hamiltonian Cycle	MaxDiCut
Treewidth	FPT	FPT
Dir. Treewidth	XP	
DAG-width	XP	
Kelly-width	XP	
Dir. Pathwidth	XP	
Cycle rank	XP	

Summary of results

	Hamiltonian Cycle	MaxDiCut
Treewidth	FPT	FPT
Dir. Treewidth	XP	
DAG-width	XP	
Kelly-width	XP	
Dir. Pathwidth	XP	
Cycle rank	XP $W[2]$-hard	NP-complete

Summary of results

	Hamiltonian Cycle	MaxDiCut
Treewidth	FPT	FPT
Dir. Treewidth	XP $W[2]$-hard	NP-complete
DAG-width	XP $W[2]$-hard	NP-complete
Kelly-width	XP $W[2]$-hard	NP-complete
Dir. Pathwidth	XP $W[2]$-hard	NP-complete
Cycle rank	XP $W[2]$-hard	NP-complete

Conclusion

- Currently known digraph decompositions don’t work as well as treewidth.
- Why?
  - Perhaps DAGs are not a good starting point.
  - Perhaps different cops-and-robber games could reveal something interesting.
    - What if we allow the robber to move backwards sometimes?
- Finding a good treewidth for digraphs is an interesting open problem.

Thank You!

References

[Hunter and Kreutzer, 2007] Hunter, P. and Kreutzer, S. (2007). Digraph measures: Kelly decompositions, games, and orderings. In Bansal, N., Pruhs, K., and Stein, C., editors, *SODA*, pages 637–644. SIAM.

[Johnson et al., 2001] Johnson, T., Robertson, N., Seymour, P. D., and Thomas, R. (2001). Directed tree-width. *J. Comb. Theory, Ser. B*, 82(1):138–154.

[Obdrzálek, 2006] Obdrzálek, J. (2006). Dag-width: connectivity measure for directed graphs. In *SODA*, pages 814–821. ACM Press.