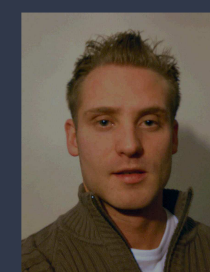
New Inapproximability Bounds for TSP

Marek Karpinski, Michael Lampis and Richard Schmied



ISAAC 2013

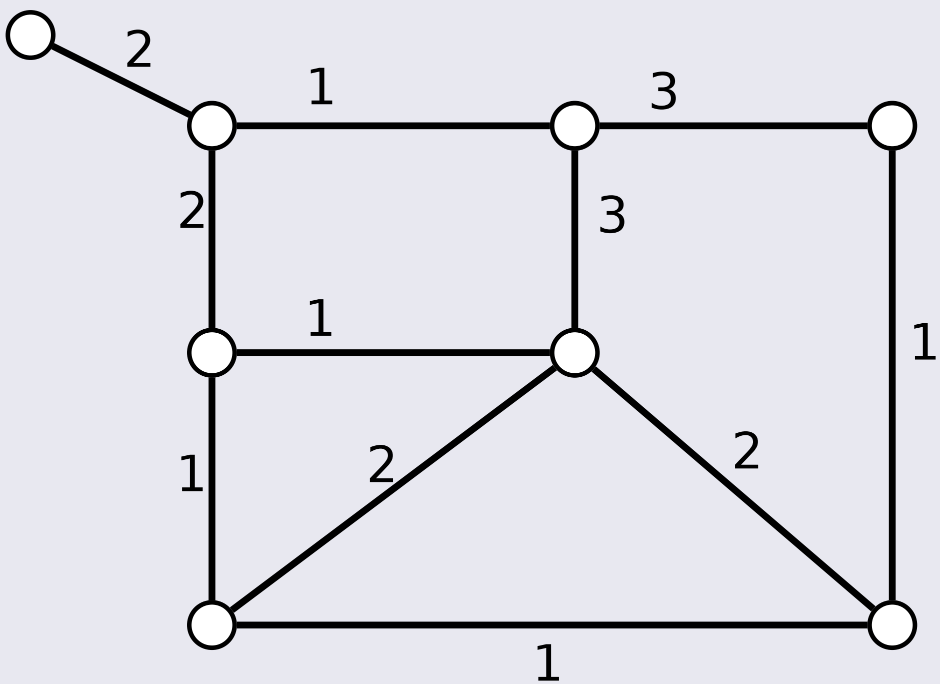
The Traveling Salesman Problem

Input:

- An edge-weighted graph $G(V, E)$

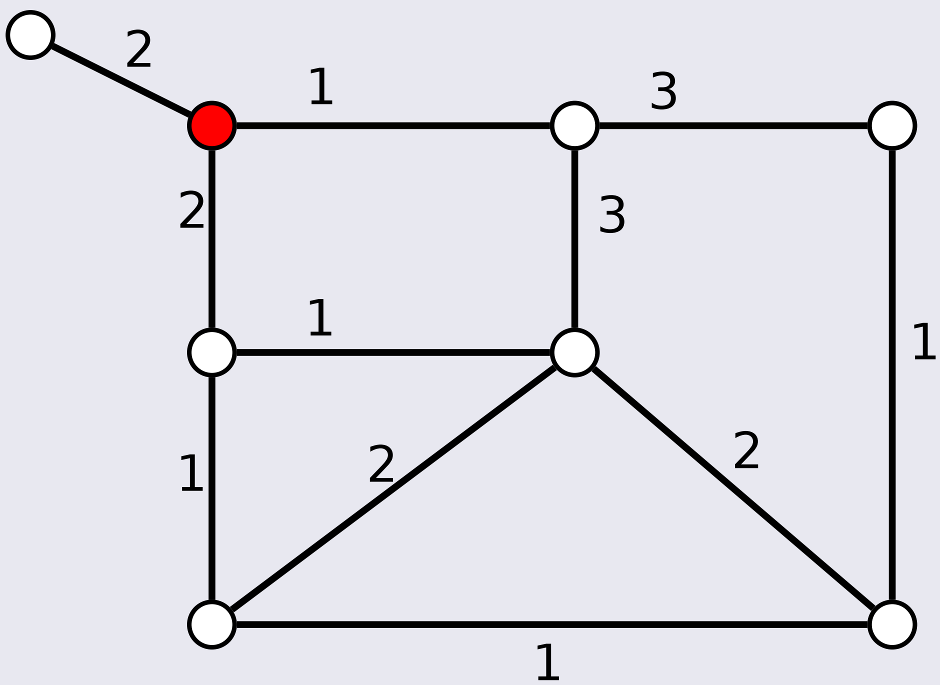
Objective:

- Find an ordering of the vertices $v_1, v_2, \ldots, v_n$ such that $d(v_1, v_2) + d(v_2, v_3) + \ldots + d(v_n, v_1)$ is minimized.
- $d(v_i, v_j)$ is the shortest-path distance of v_i, v_j on G

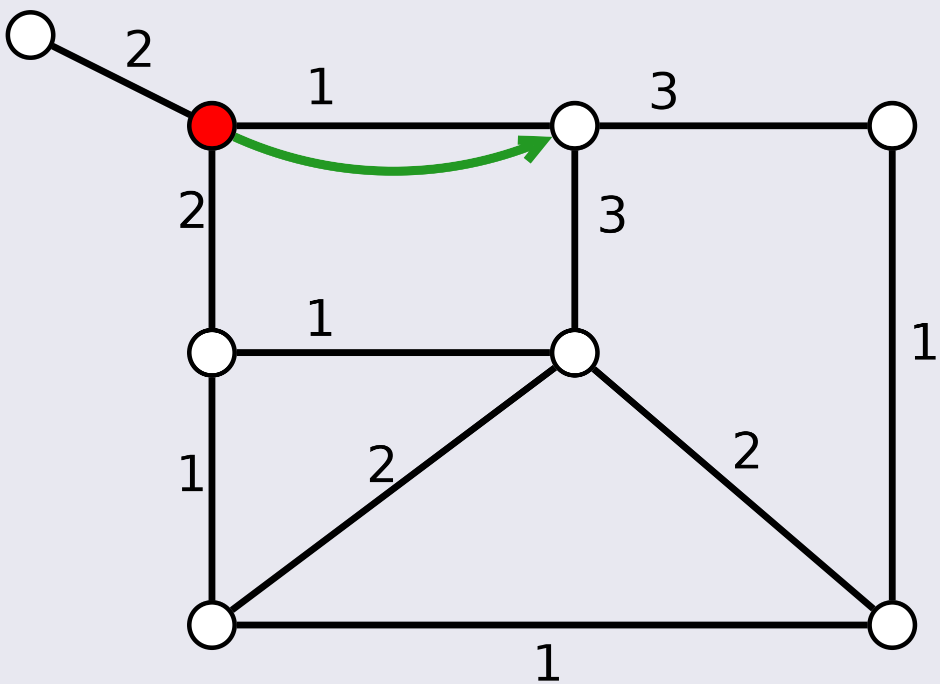

2
1
3
2
3
1
1
1
2
2
1


ACME
INSURANCE

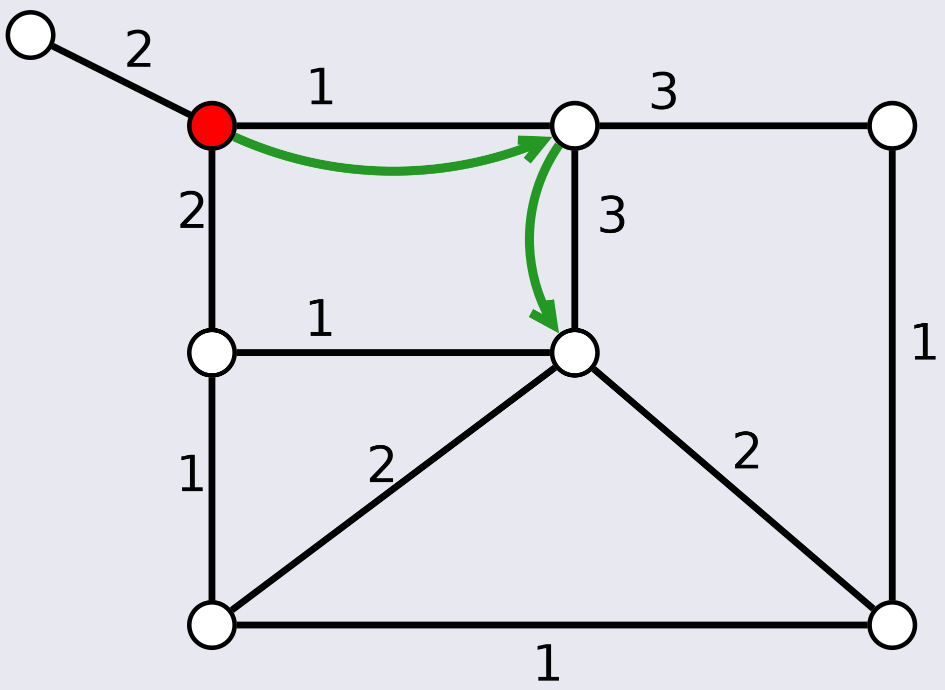
The Traveling Salesman Problem



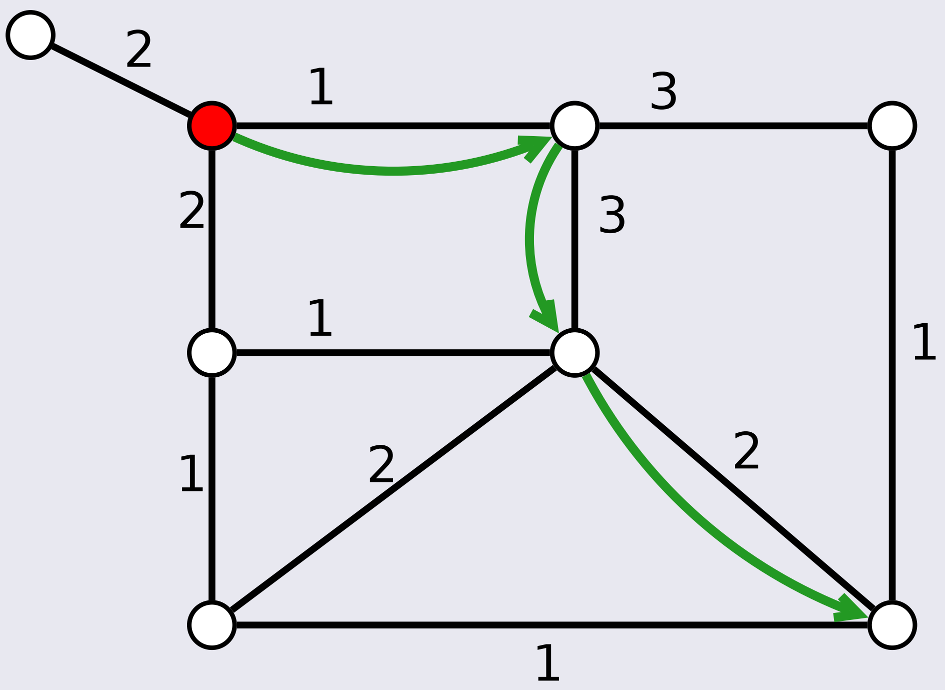
The Traveling Salesman Problem



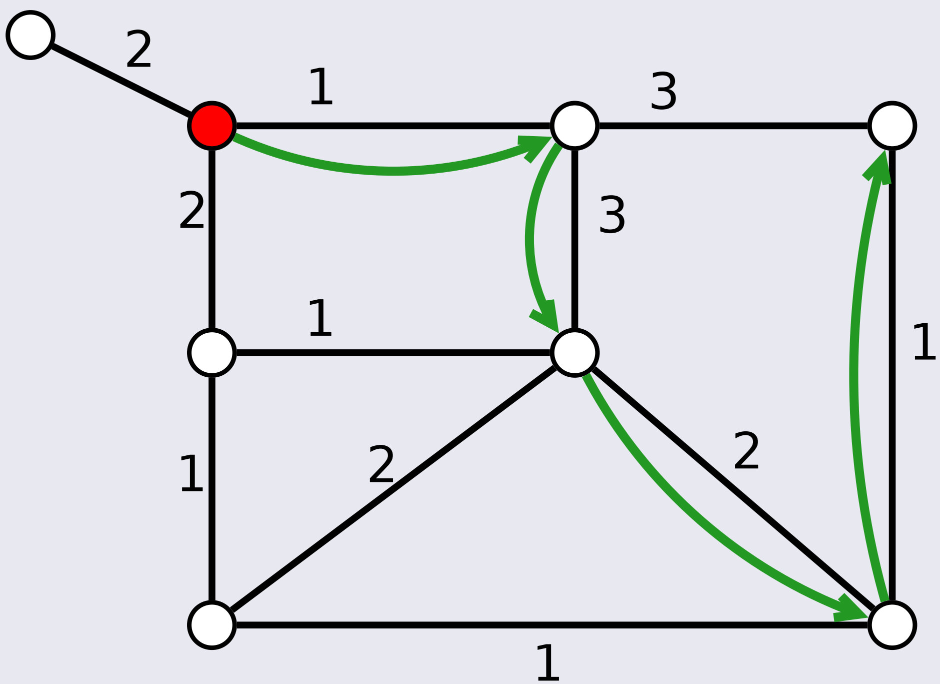
The Traveling Salesman Problem



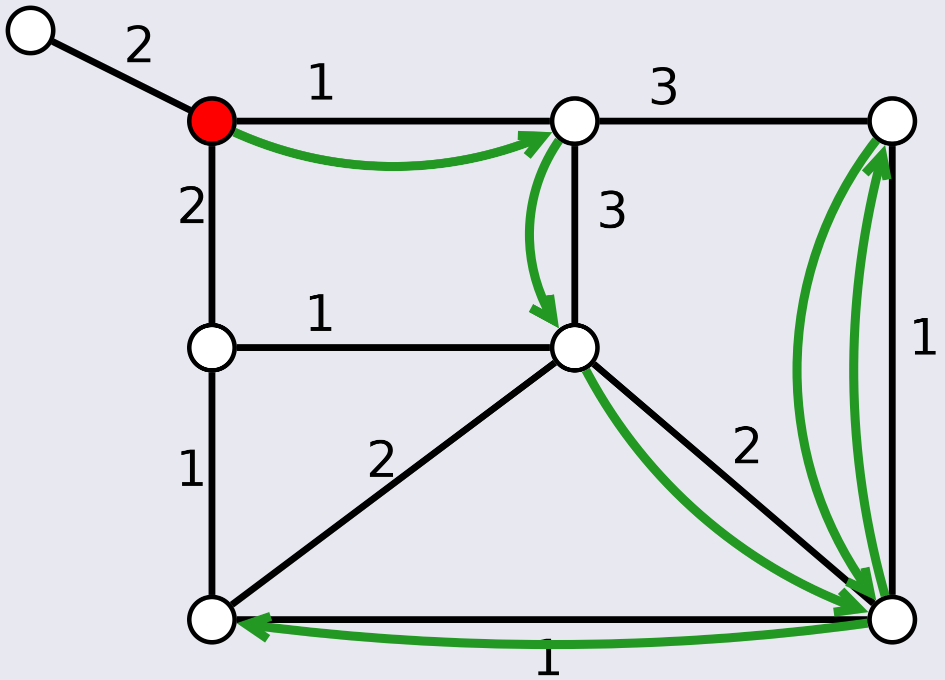
The Traveling Salesman Problem



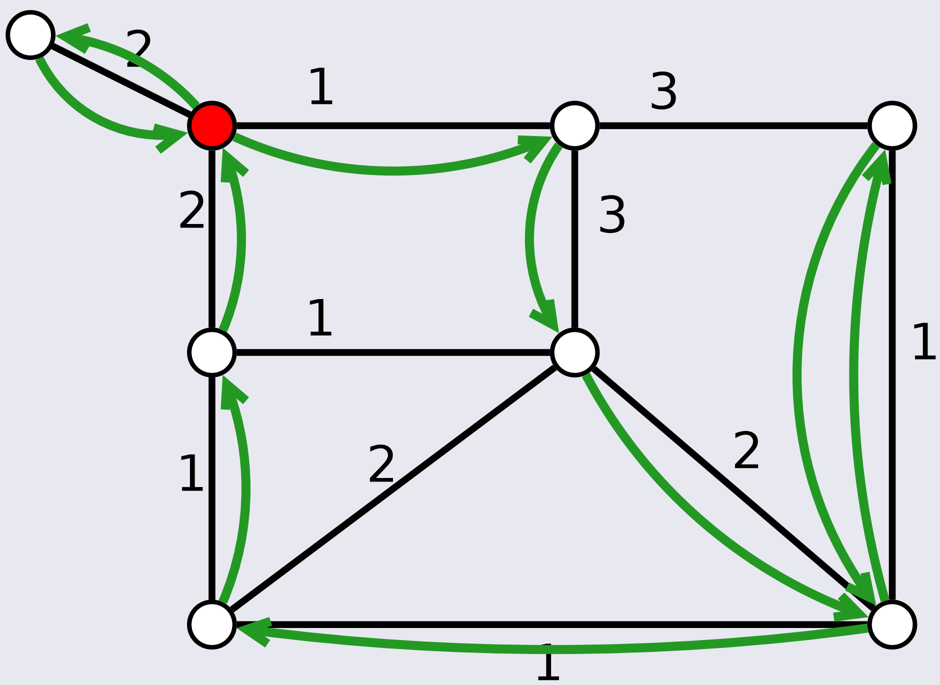
The Traveling Salesman Problem



The Traveling Salesman Problem



The Traveling Salesman Problem



TSP Approximations – Upper bounds

- $\frac{3}{2}$ approximation (Christofides 1976)

For graphic (un-weighted) case

- $\frac{3}{2} - \epsilon$ approximation (Oveis Gharan et al. FOCS '11)
- 1.461 approximation (Mömke and Svensson FOCS '11)
- $\frac{13}{9}$ approximation (Mucha STACS '12)
- 1.4 approximation (Sebö and Vygen arXiv '12)

- For ATSP the best ratio is $O(\log n / \log \log n)$ (Asadpour et al. SODA '10)

TSP Approximations – Lower bounds

- Problem is APX-hard (Papadimitriou and Yannakakis '93)
- TSP $\frac{5381}{5380}$-inapproximable, ATSP $\frac{2805}{2804}$ (Engebretsen STACS '99)
- TSP $\frac{3813}{3812}$-inapproximable (Böckenhauer et al. STACS '00)
- TSP $\frac{220}{219}$-inapproximable, ATSP $\frac{117}{116}$ (Papadimitriou and Vempala STOC '00, Combinatorica '06)
- TSP $\frac{185}{184}$-inapproximable (L. APPROX '12)

TSP Approximations – Lower bounds

- Problem is APX-hard (Papadimitriou and Yannakakis '93)
- TSP $\frac{5381}{5380}$-inapproximable, ATSP $\frac{2805}{2804}$ (Engebretsen STACS '99)
- TSP $\frac{3813}{3812}$-inapproximable (Böckenhauer et al. STACS '00)
- TSP $\frac{220}{219}$-inapproximable, ATSP $\frac{117}{116}$ (Papadimitriou and Vempala STOC '00, Combinatorica '06)
- TSP $\frac{185}{184}$-inapproximable (L. APPROX '12)

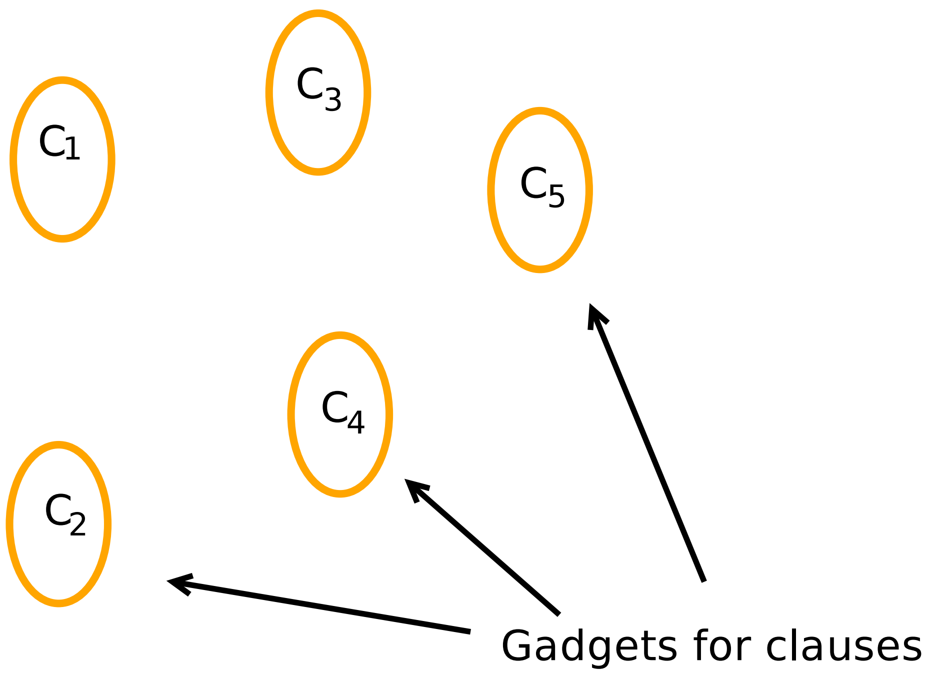
This talk:

Theorem

It is NP-hard to approximate TSP better than $\frac{123}{122}$ and ATSP better than $\frac{75}{74}$.

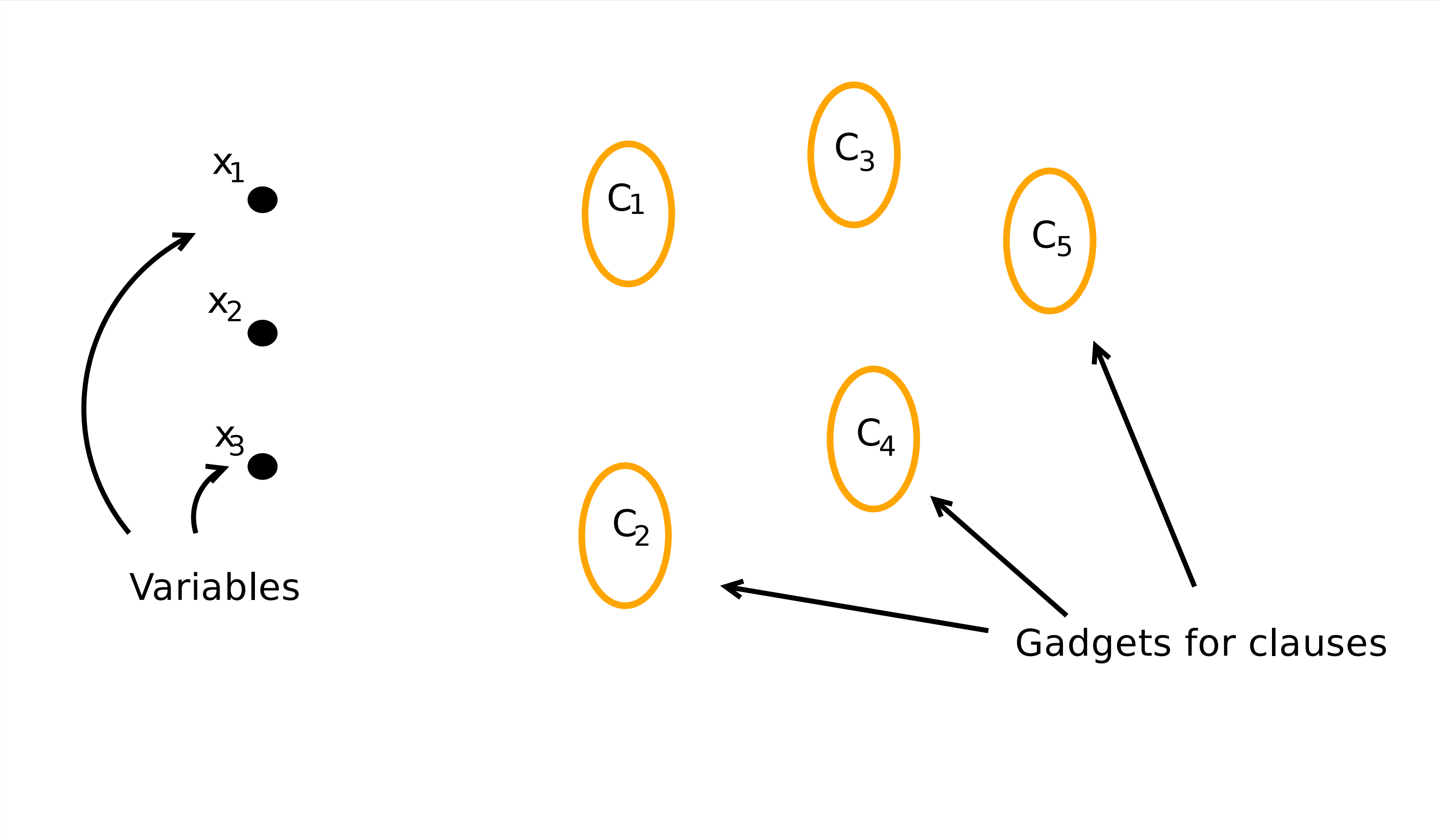
We reduce some inapproximable CSP (e.g. MAX-3SAT) to TSP.

Reduction Technique



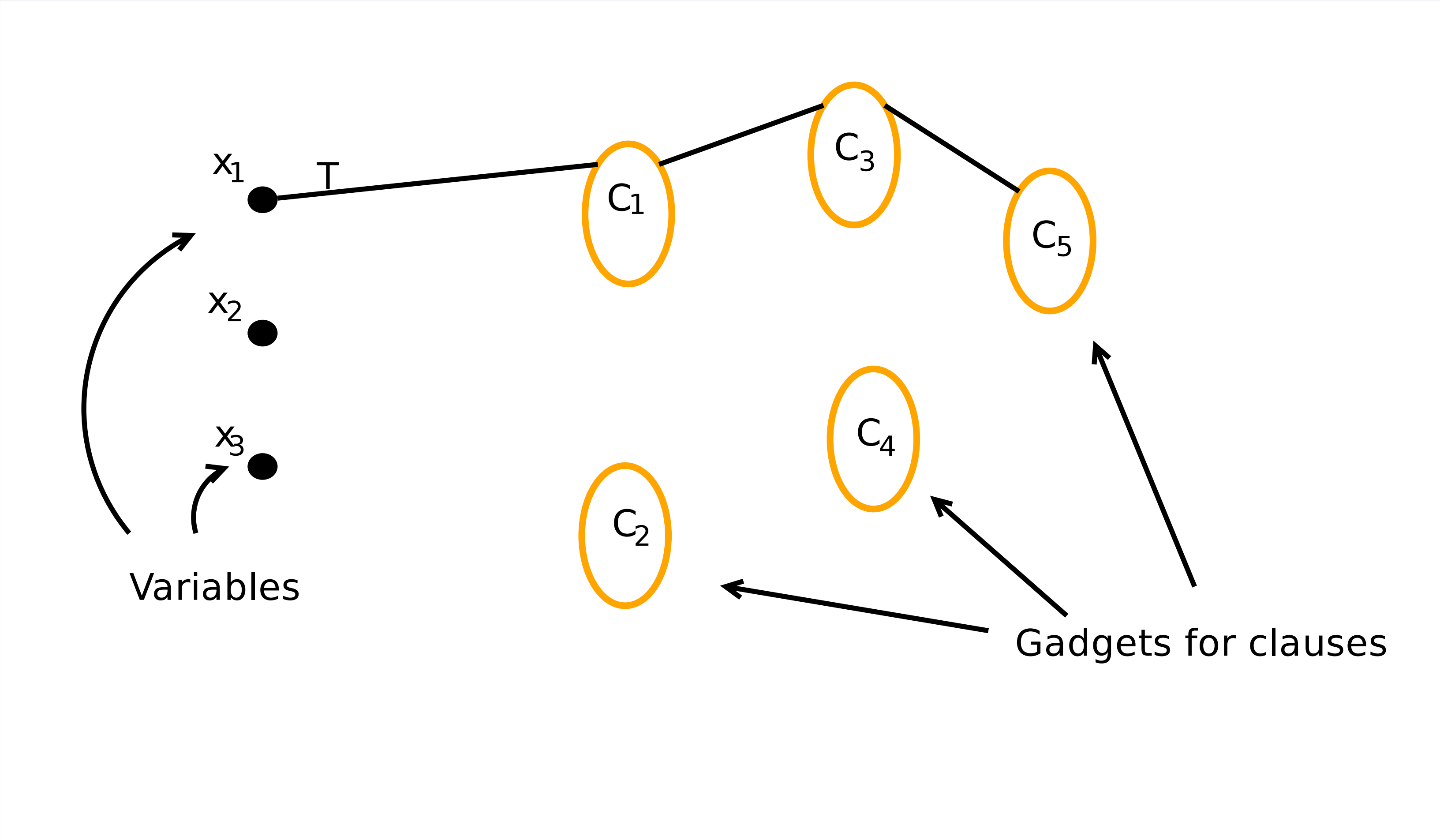
First, design some gadgets to represent the clauses

Reduction Technique



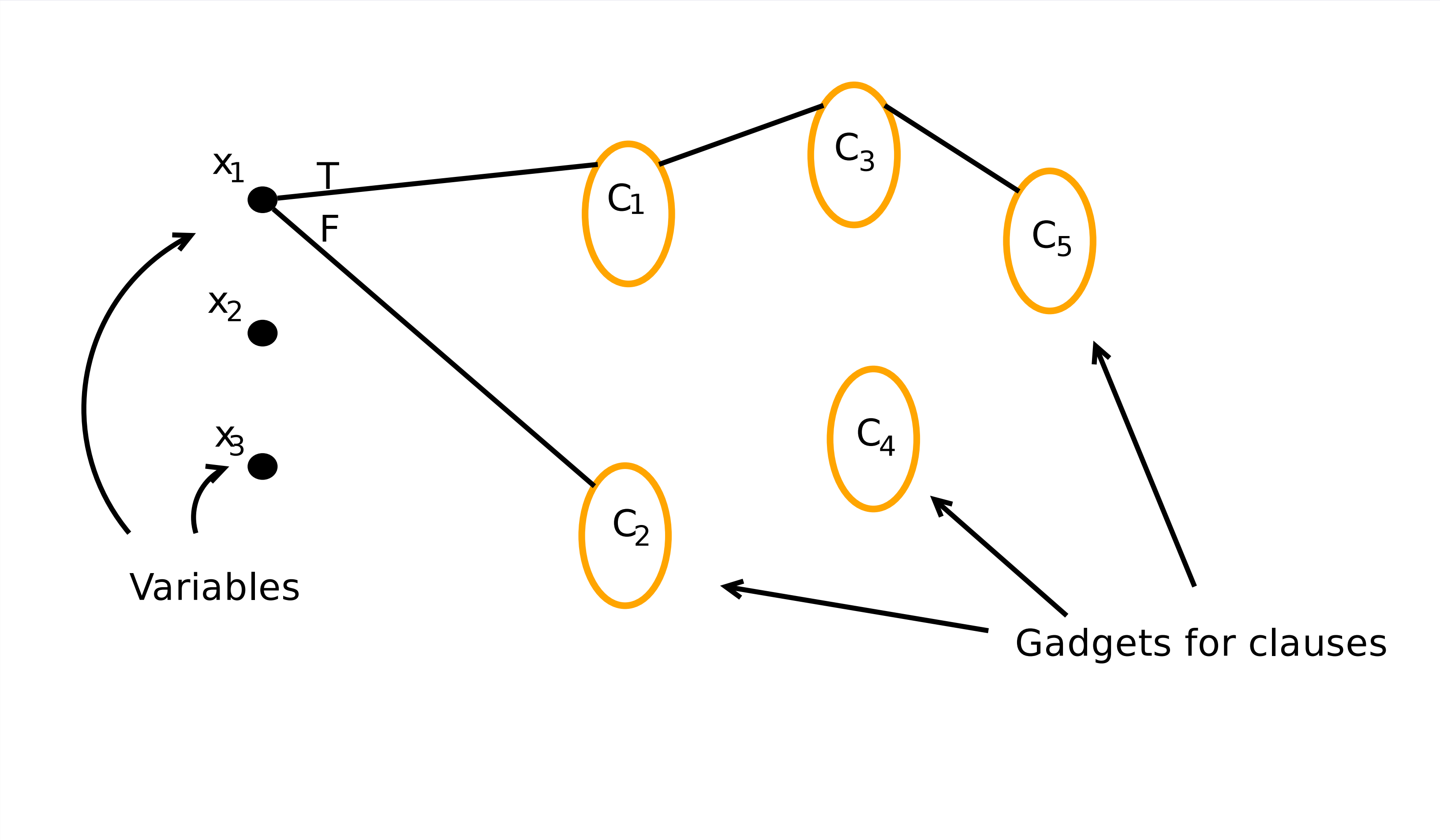
Then, add some choice vertices to represent truth assignments to variables

Reduction Technique



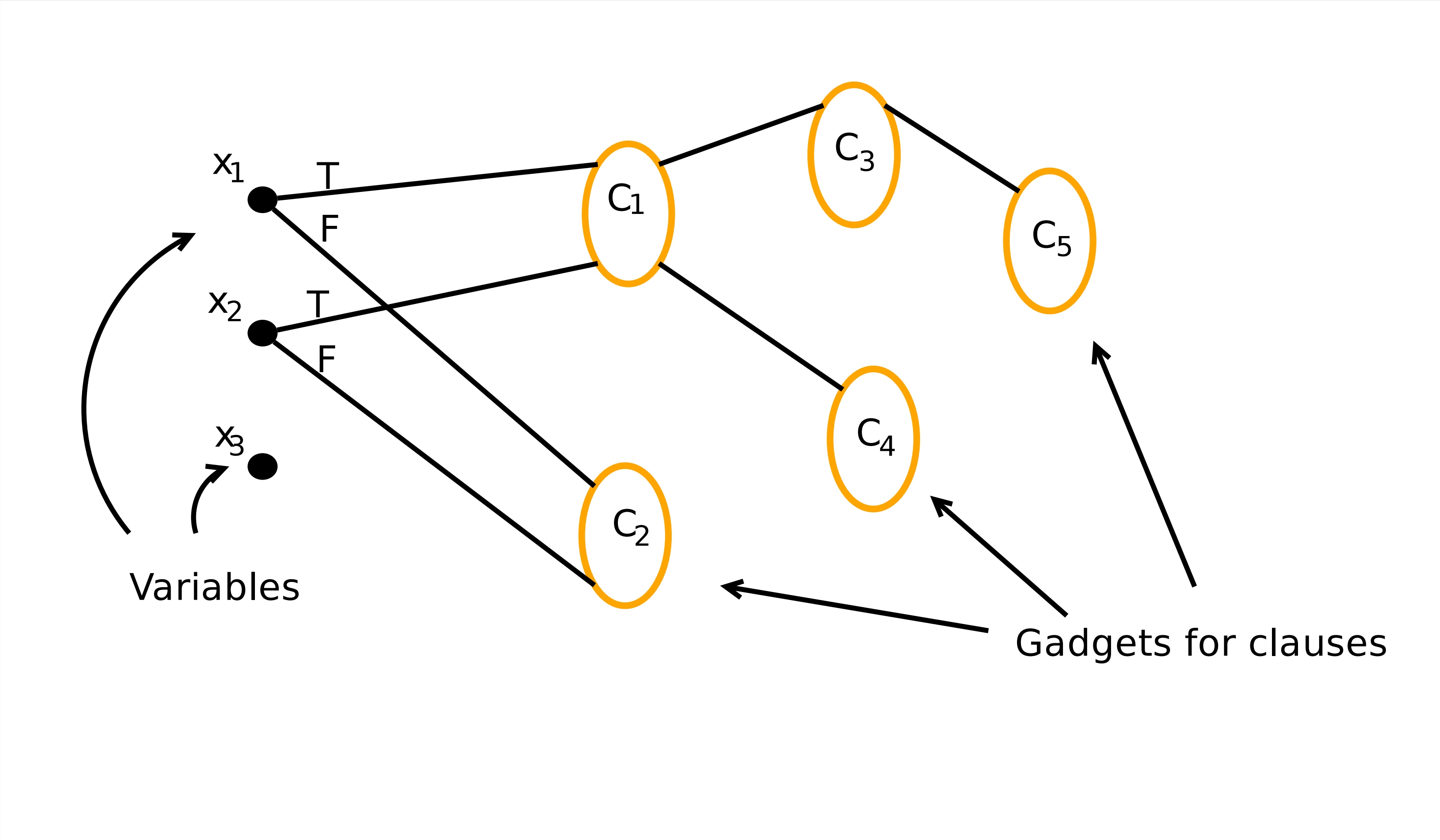
For each variable, create a path through clauses where it appears positive

Reduction Technique

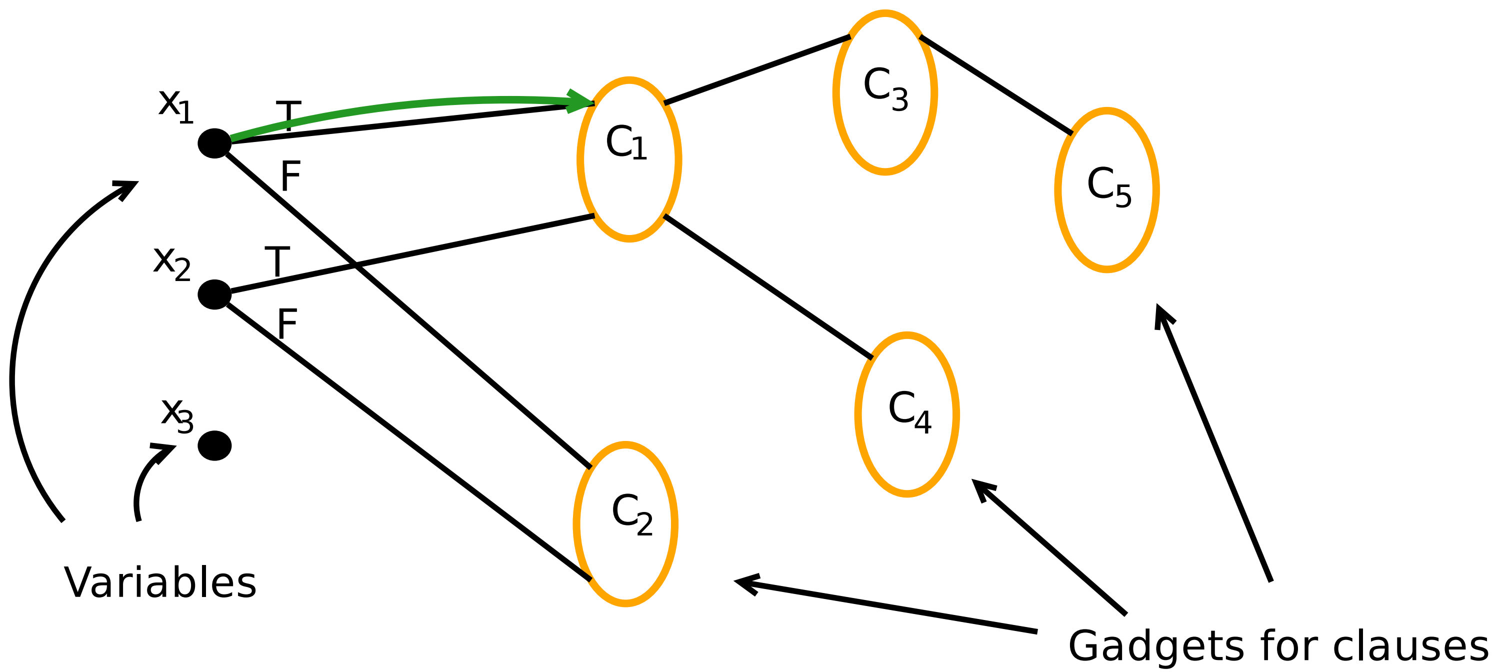


. . . and another path for its negative appearances

Reduction Technique

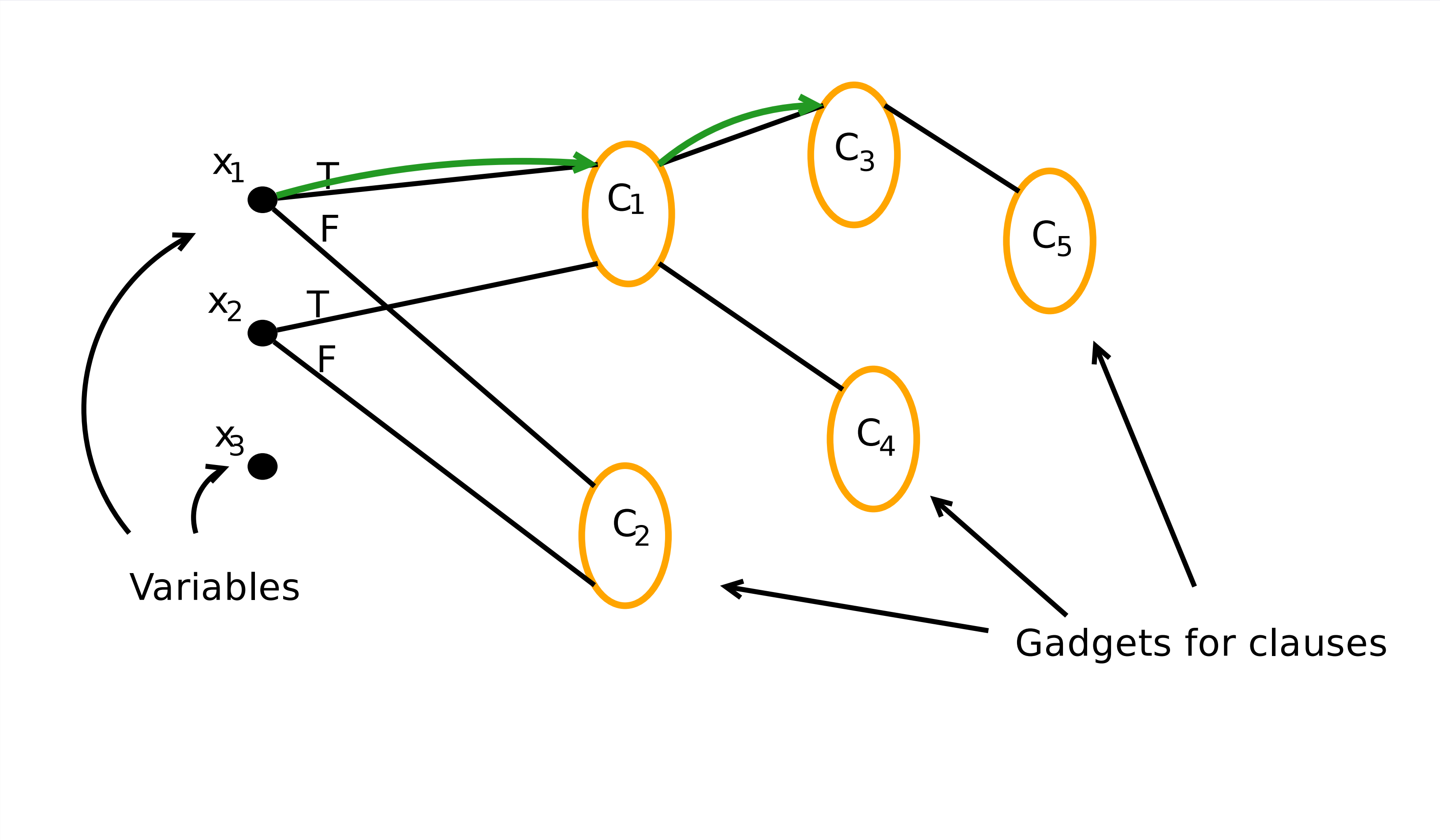


Reduction Technique

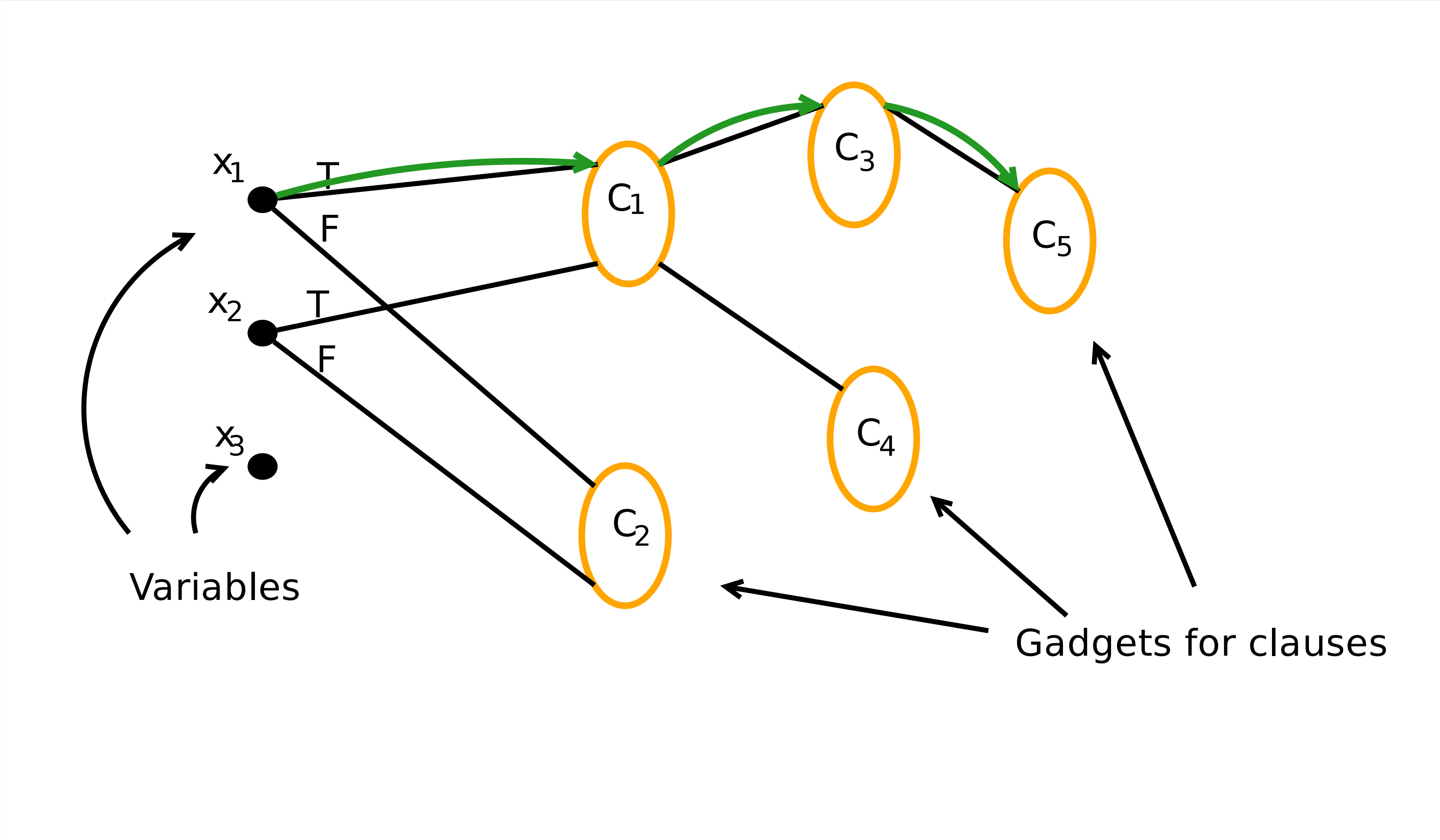


A truth assignment dictates a general path

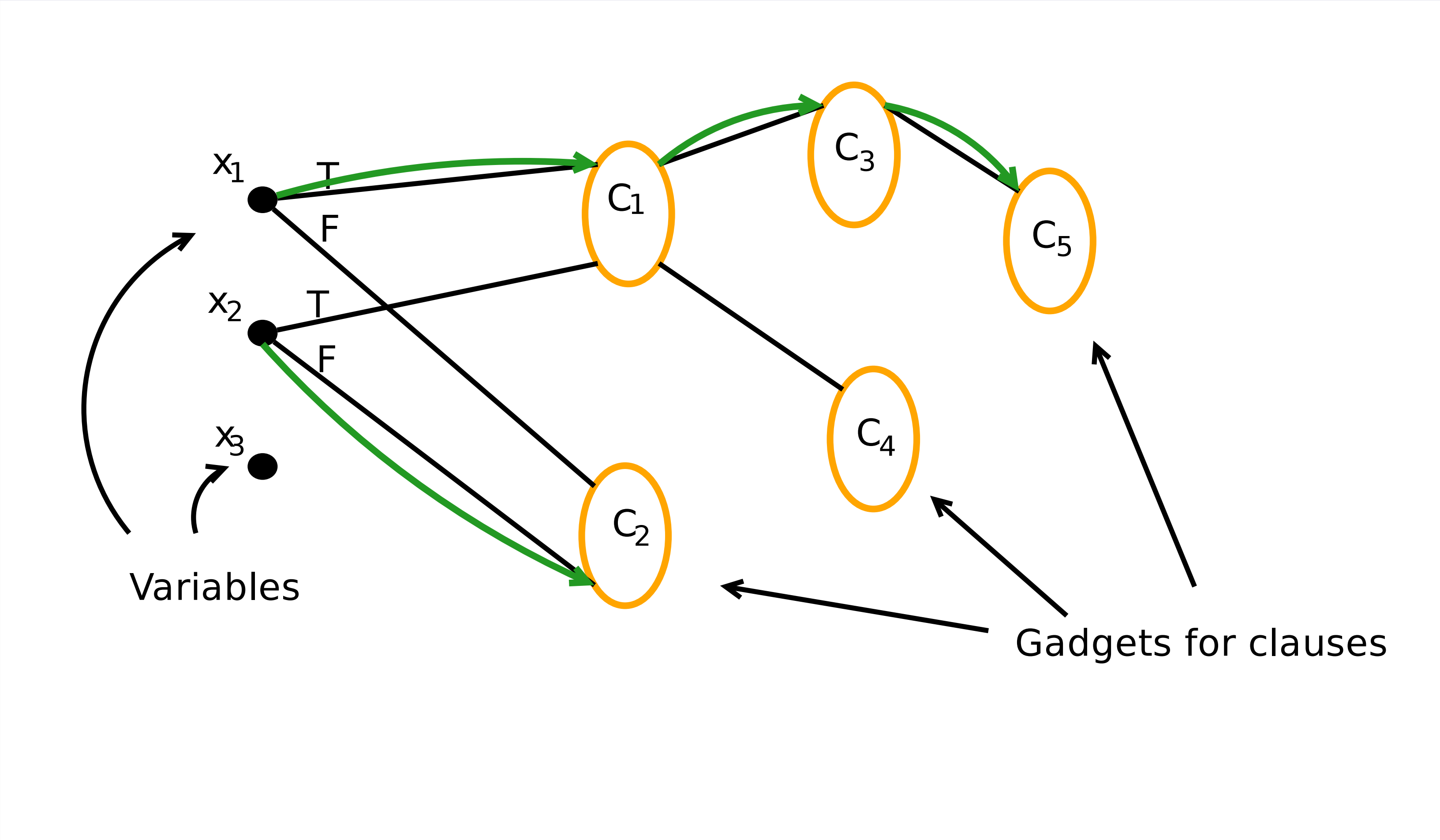
Reduction Technique



Reduction Technique

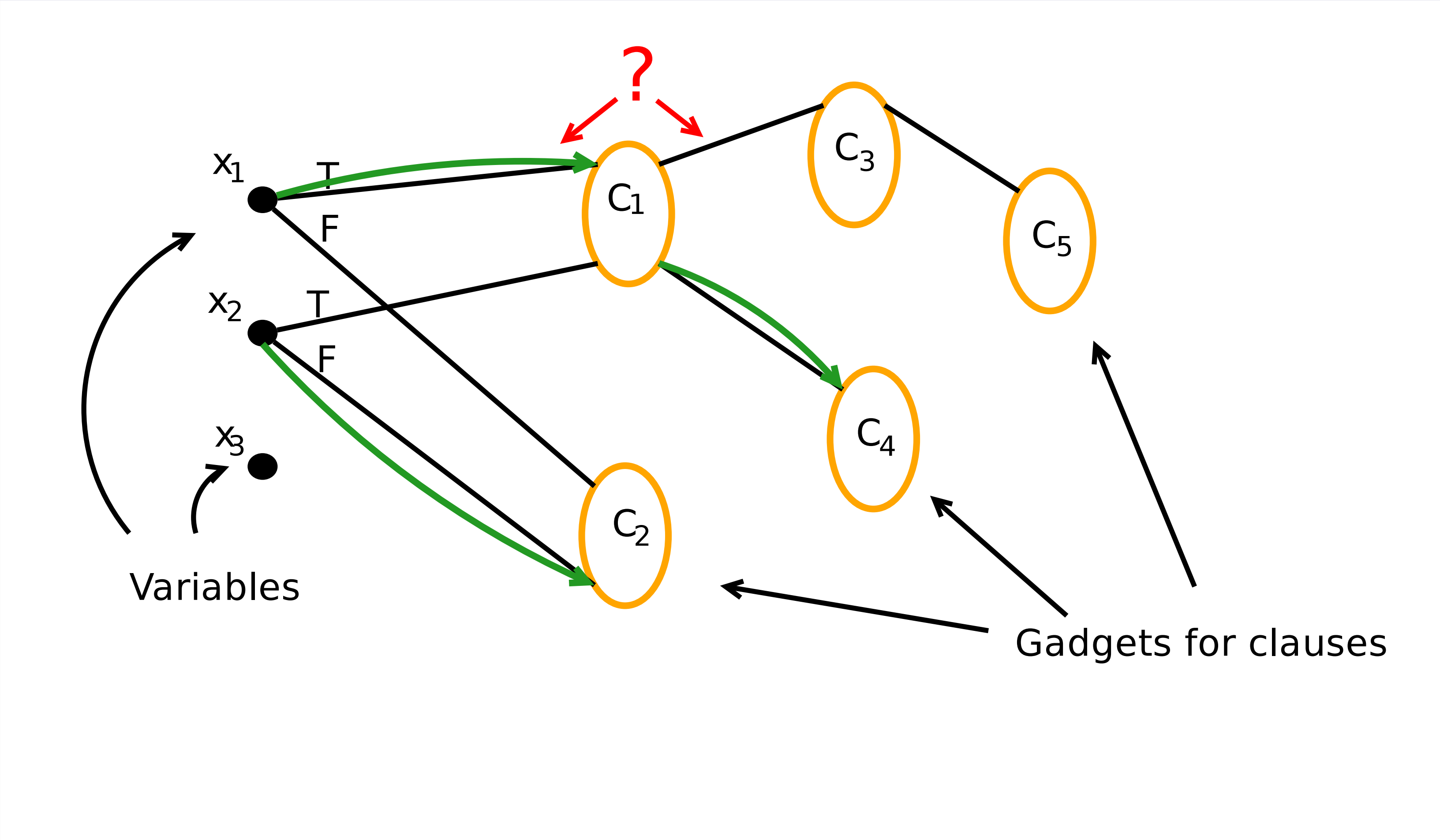


Reduction Technique



We must make sure that gadgets are cheaper to traverse if corresponding clause is satisfied

Reduction Technique



For the converse direction we must make sure that ”cheating” tours are not optimal!

How to ensure consistency

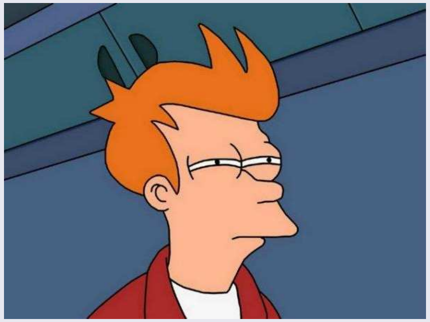
- Basic idea here: consistency would be easy if each variable occurred at most c times, c a constant.
  - Cheating would only help a tour ”fix” a bounded number of clauses.

How to ensure consistency

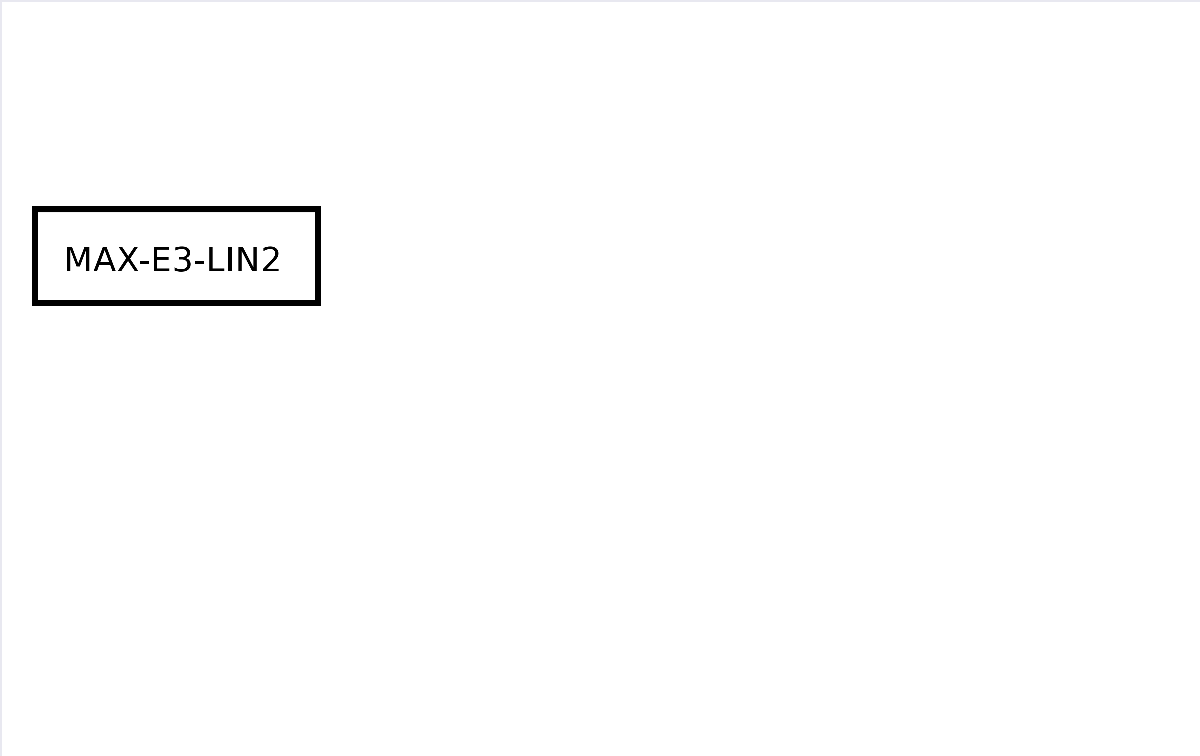
- Basic idea here: consistency would be easy if each variable occurred at most c times, c a constant.
  - Cheating would only help a tour ”fix” a bounded number of clauses.
- We will rely on techniques and tools used to prove inapproximability for bounded-occurrence CSPs.
  - Main tool: “amplifier graph” constructions due to Berman and Karpinski.
  - We introduce a new bi-wheel amplifier.

How to ensure consistency

- Basic idea here: consistency would be easy if each variable occurred at most c times, c a constant.
  - Cheating would only help a tour ”fix” a bounded number of clauses.
- We will rely on techniques and tools used to prove inapproximability for bounded-occurrence CSPs.
  - Main tool: “amplifier graph” constructions due to Berman and Karpinski.
  - We introduce a new bi-wheel amplifier.
- Result: modular proof, improved bounds
- Potential for further improvements: parts of the reduction have no overhead!

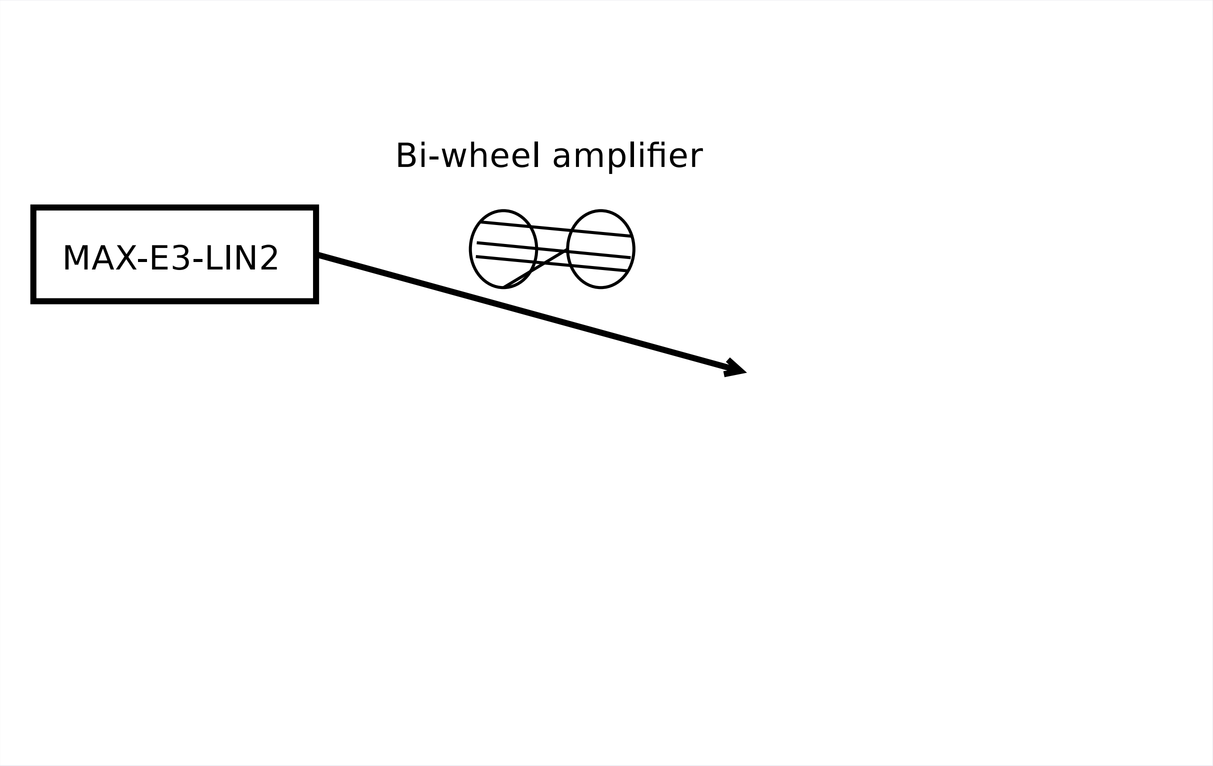


Overview



We start from an instance of MAX-E3-LIN2. Given a set of linear equations (mod 2) each of size three satisfy as many as possible. Problem known to be 2-inapproximable (Håstad ’01)

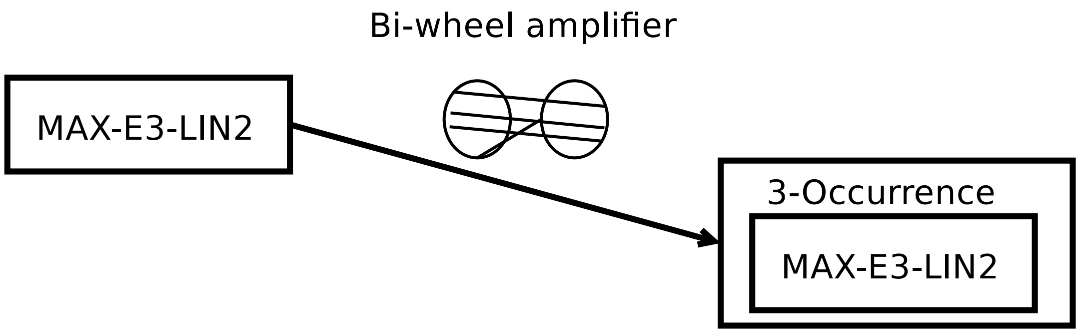
Overview



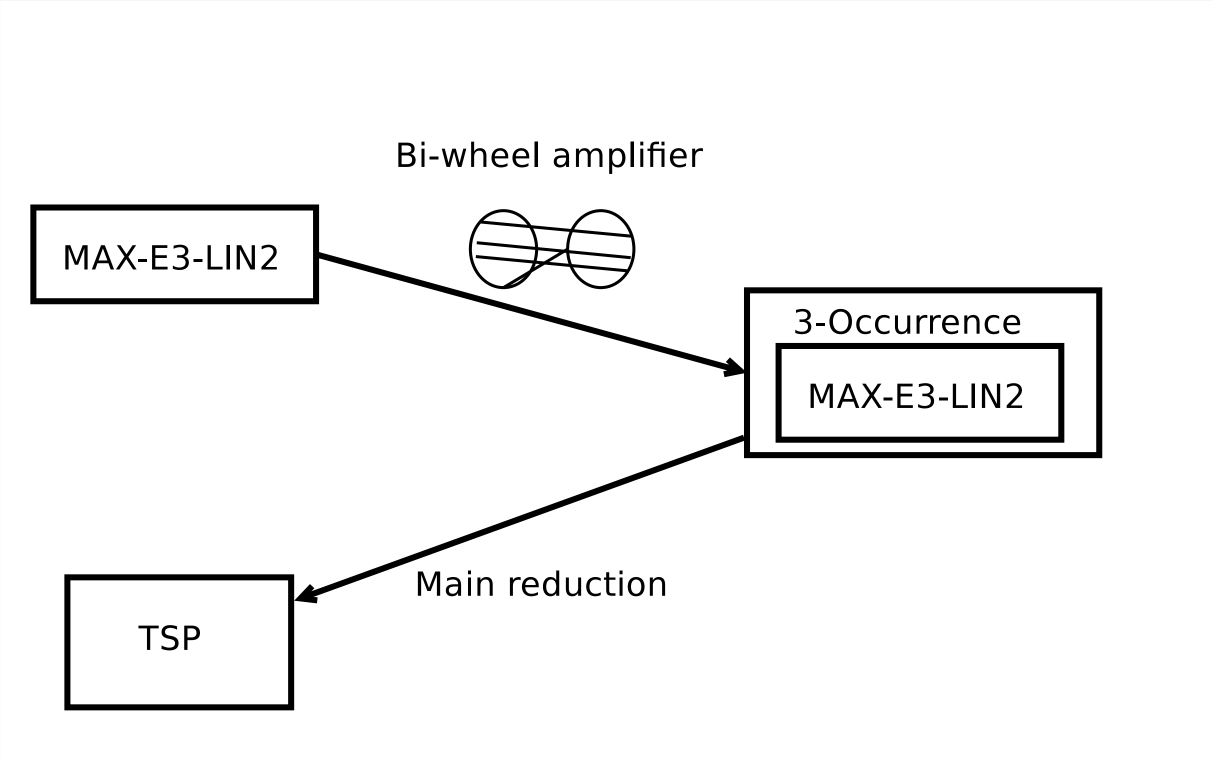
We use a new version of the Berman-Karpinski wheel amplifier: the **bi-wheel**.

We obtain an instance where each variable appears exactly 3 times (and most equations have size 2).

Overview



Overview



From this instance we construct a TSP/ATSP graph instance.

Amplifiers and Bounded Occurrences

Amplifiers

What is an amplifier?

Amplifiers

What is an amplifier?

Amplifiers

An amplifier is a graph with edge expansion 1 for a subset of its vertices.

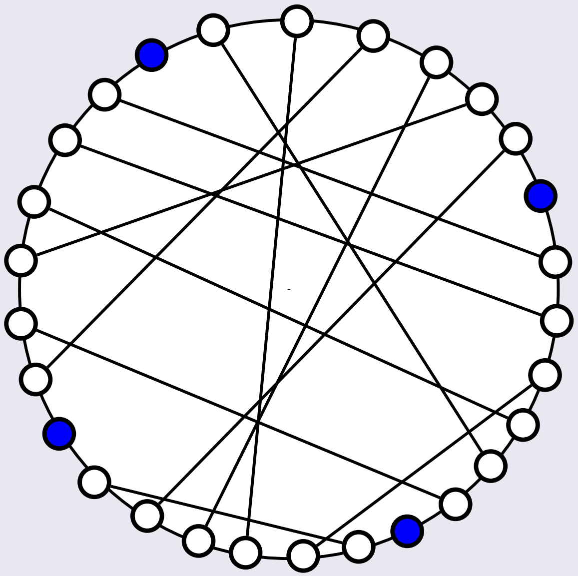
Amplifiers

An amplifier is a graph with edge expansion 1 for a subset of its vertices.

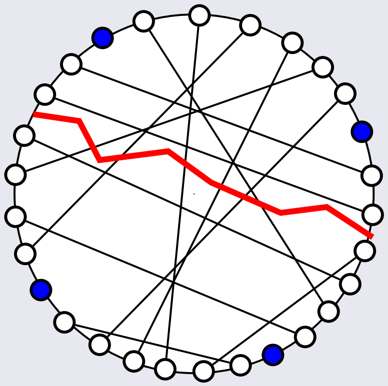
3-regular wheel amplifier [Berman Karpinski 01]

- Start with a cycle on $7n$ vertices.
- Every seventh vertex is a contact vertex. Other vertices are checkers.
- Take a random perfect matching of checkers.

- Crucial Property: whp any partition cuts more edges than the number of contact vertices on the smaller set.

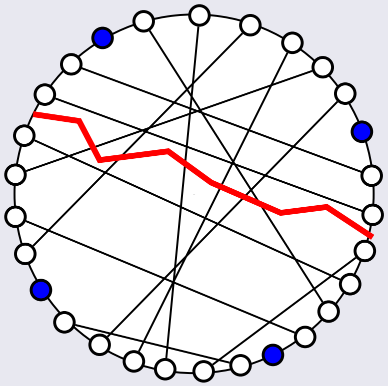


How to use amplifiers



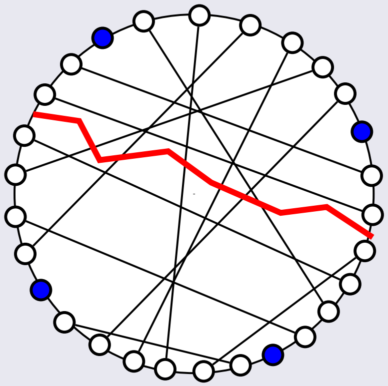
- Input: MAX-E3-LIN2, variables appear B times.
  - For each variable x construct an amplifier.
  - For each vertex construct a variable x_i, y_i
  - For each edge of the amplifier make an equality constraint ($y_i + y_j = 0$).
  - Use the x_i's in the original constraints.

How to use amplifiers



- Input: MAX-E3-LIN2, variables appear B times.
  - For each variable x construct an amplifier.
  - For each vertex construct a variable x_i, y_i
  - For each edge of the amplifier make an equality constraint $(y_i + y_j = 0)$.
  - Use the x_i's in the original constraints.
- Inconsistent assignments $\rightarrow$ partition of vertices
  - But cut edges $\rightarrow$ violated equalities
  - Large cut $\rightarrow$ Flipping the minority part is always good
  - $\rightarrow$ Consistent assignment is optimal

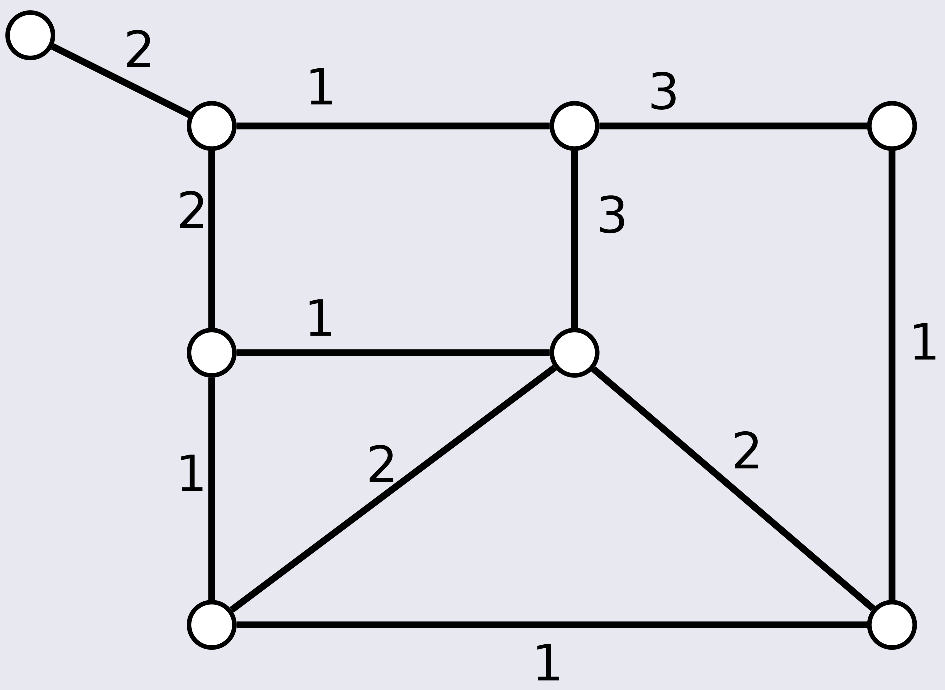
How to use amplifiers

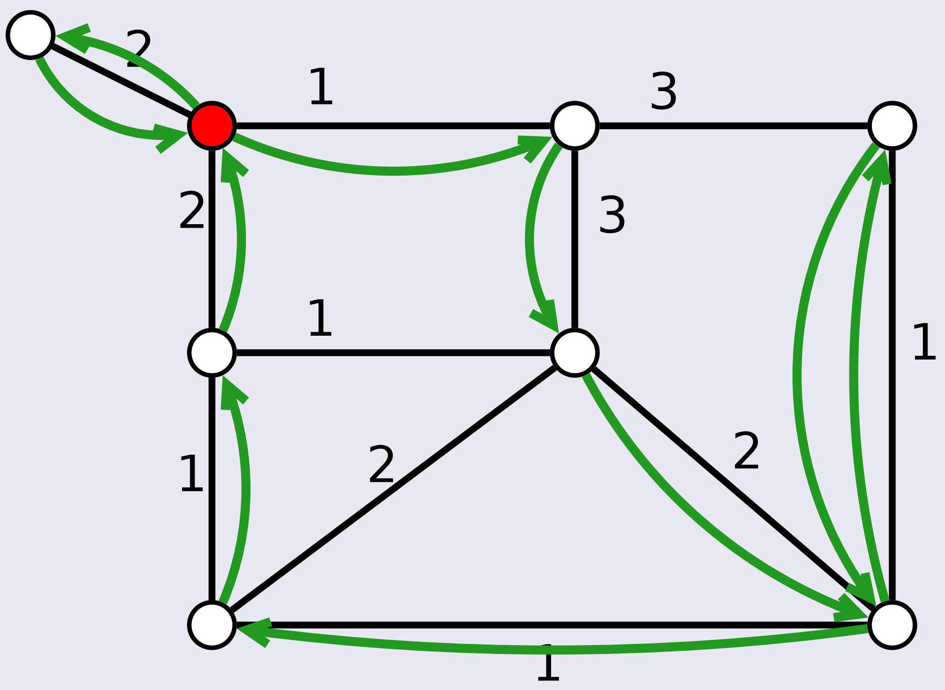


- Input: MAX-E3-LIN2, variables appear B times.
  - For each variable x construct an amplifier.
  - For each vertex construct a variable x_i, y_i
  - For each edge of the amplifier make an equality constraint $(y_i + y_j = 0)$.
  - Use the x_i's in the original constraints.
- Inconsistent assignments $\rightarrow$ partition of vertices
  - But cut edges $\rightarrow$ violated equalities
  - Large cut $\rightarrow$ Flipping the minority part is always good
  - $\rightarrow$ Consistent assignment is optimal
- Problem: New equations are pure overhead! (always satisfiable)

The reduction

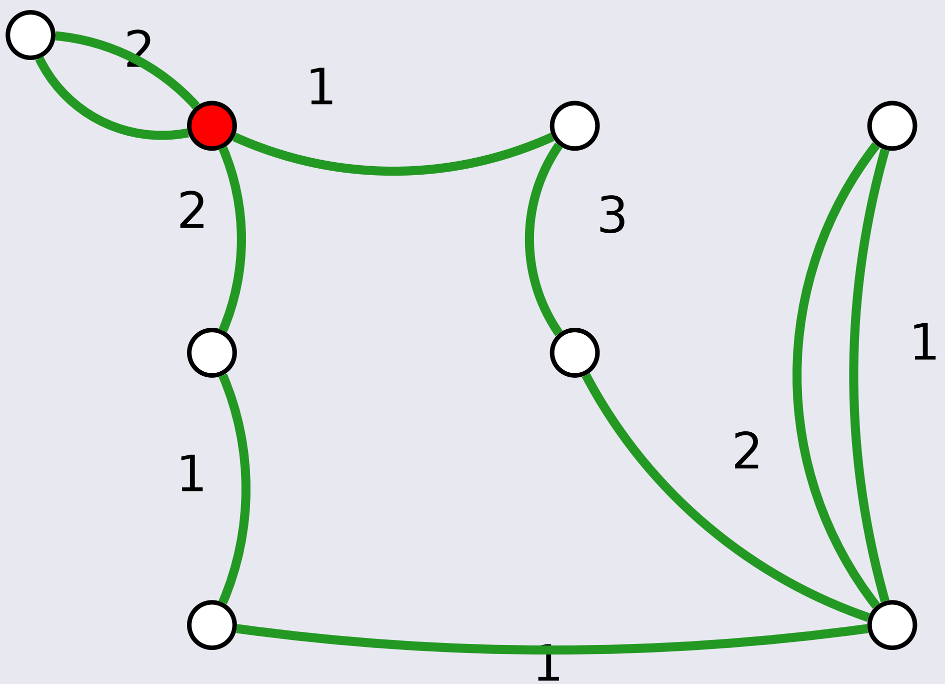
TSP and Euler tours




2
1
3
2
3
1
1
1
2
2
1


ACME
INSURANCE

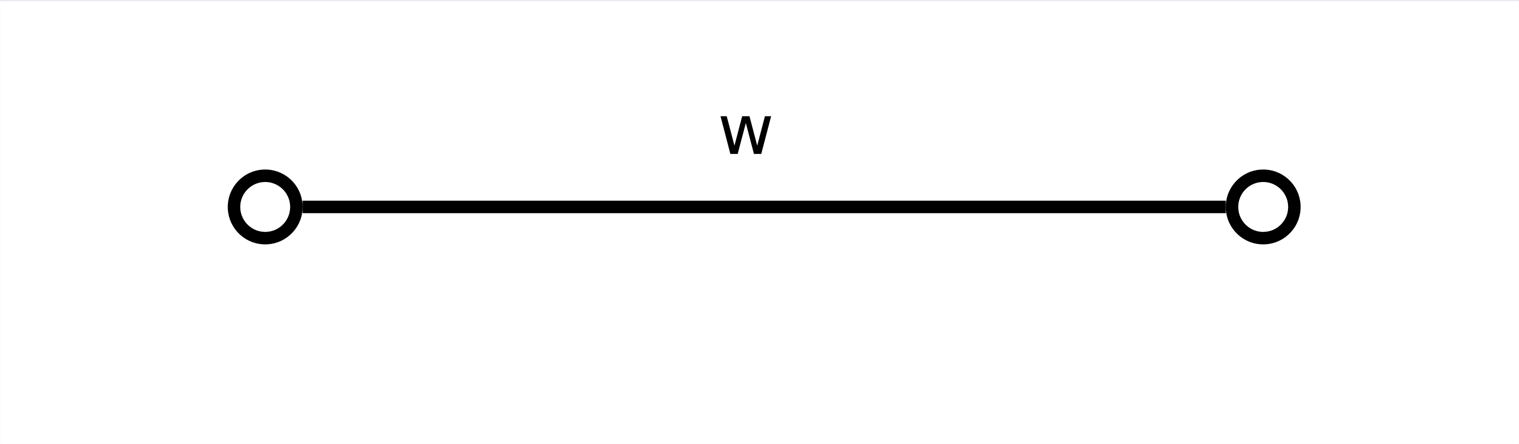
TSP and Euler tours



TSP and Euler tours

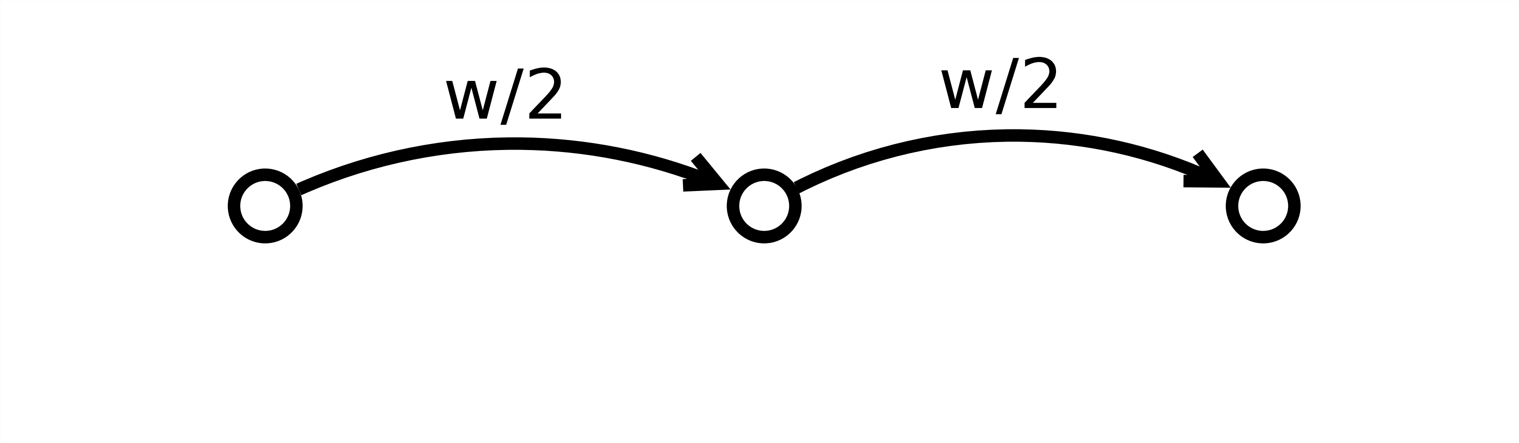
- A TSP tour gives an Eulerian multi-graph composed with edges of G.
- An Eulerian multi-graph composed with edges of G gives a TSP tour.
  - TSP $\equiv$ Select a multiplicity for each edge so that the resulting multi-graph is Eulerian and total cost is minimized
  - **Note**: no edge is used more than twice

Gadget – Forced Edges

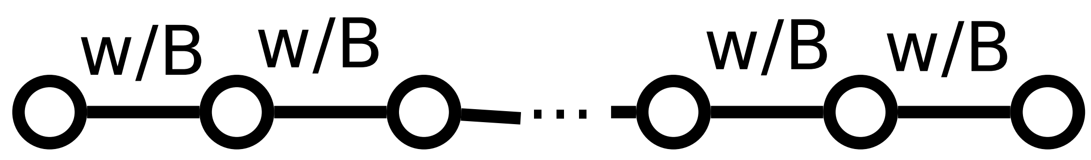


We would like to be able to dictate in our construction that a certain edge has to be used at least once.

Gadget – Forced Edges

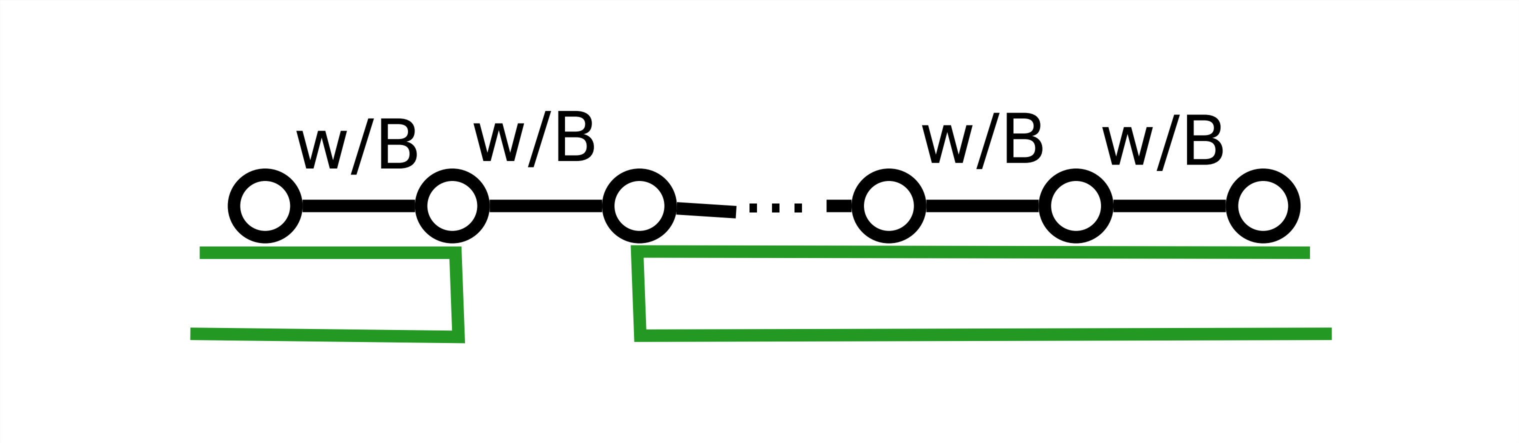


If we had directed edges, this could be achieved by adding a dummy intermediate vertex



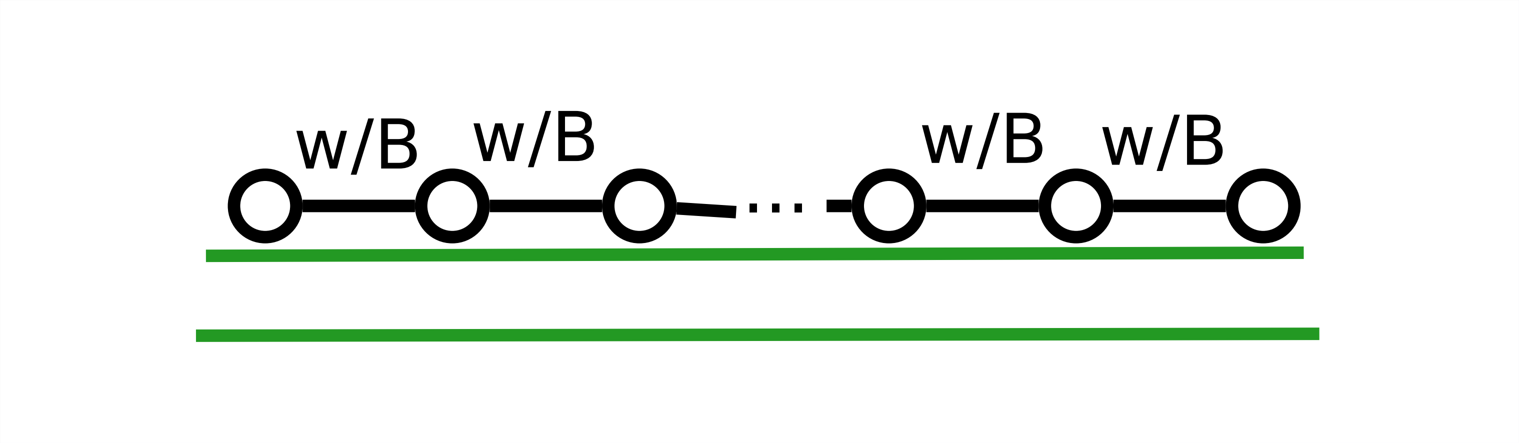
Here, we add many intermediate vertices and evenly distribute the weight w among them. Think of B as very large.

Gadget – Forced Edges



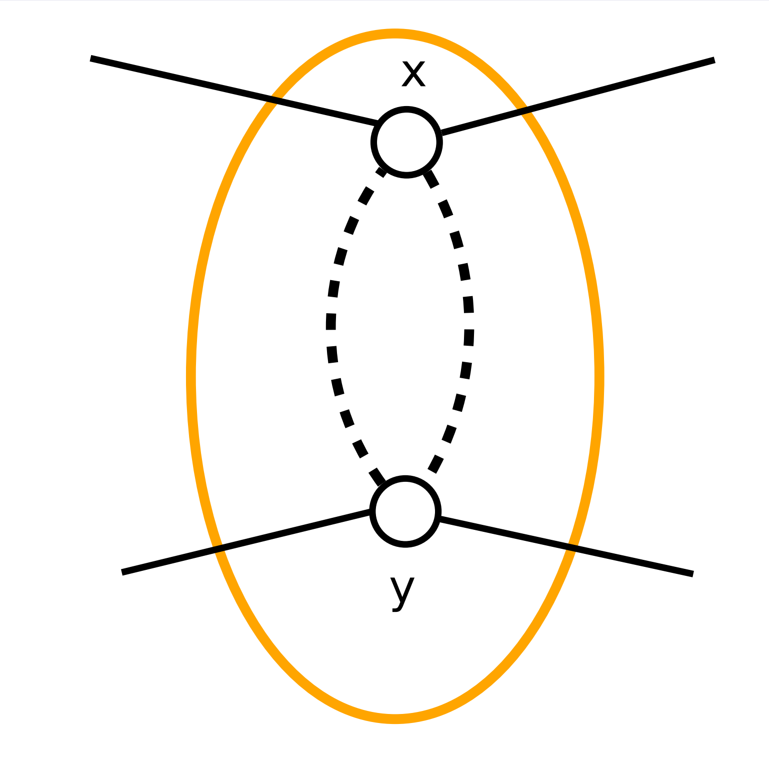
At most one of the new edges may be unused, and in that case all others are used twice.

Gadget – Forced Edges



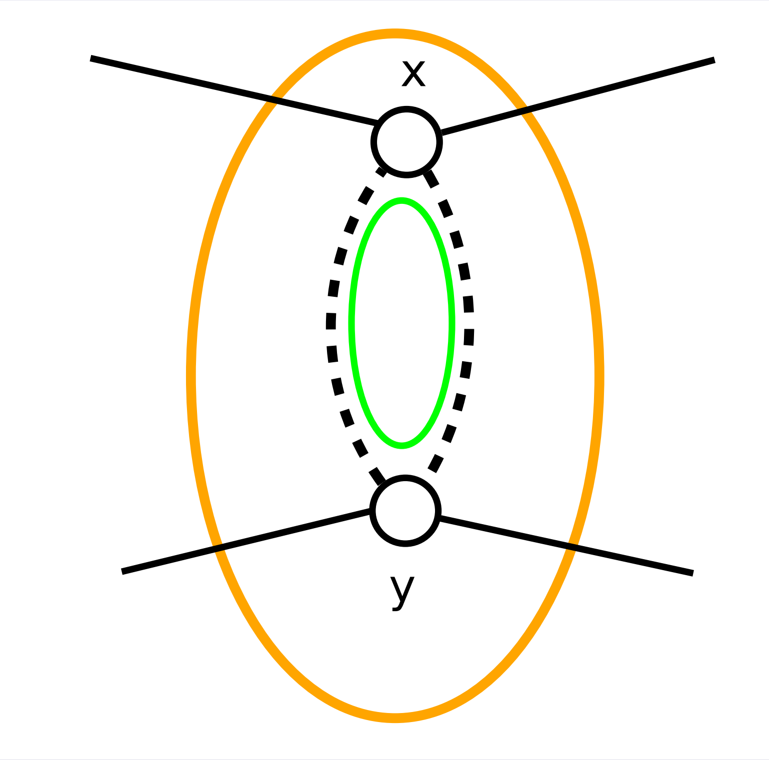
In that case, adding two copies of that edge to the solution doesn't hurt much (for B sufficiently large).

Gadget for Inequality



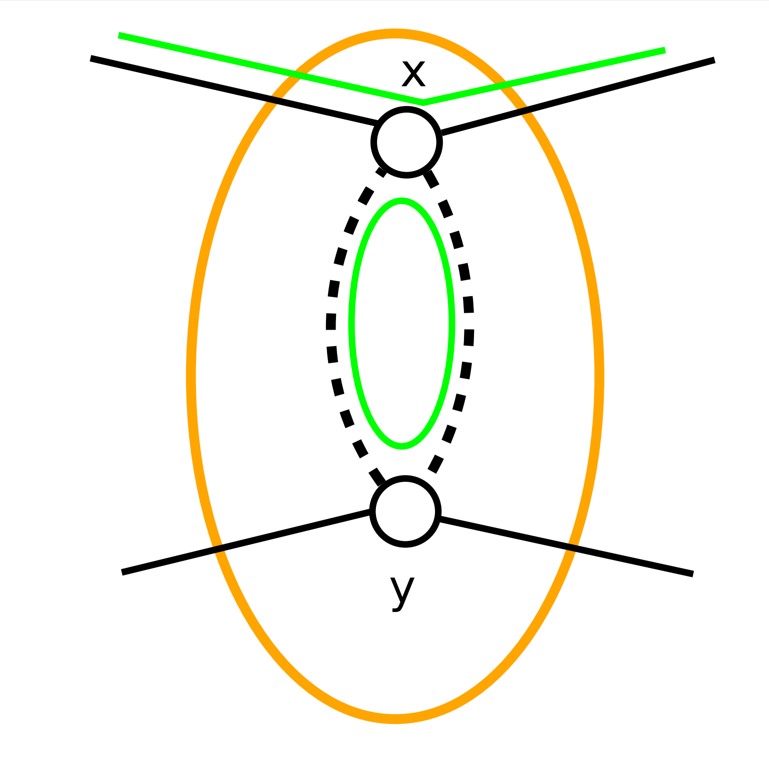
We can encode $x + y = 1$ with two parallel forced edges

Gadget for Inequality



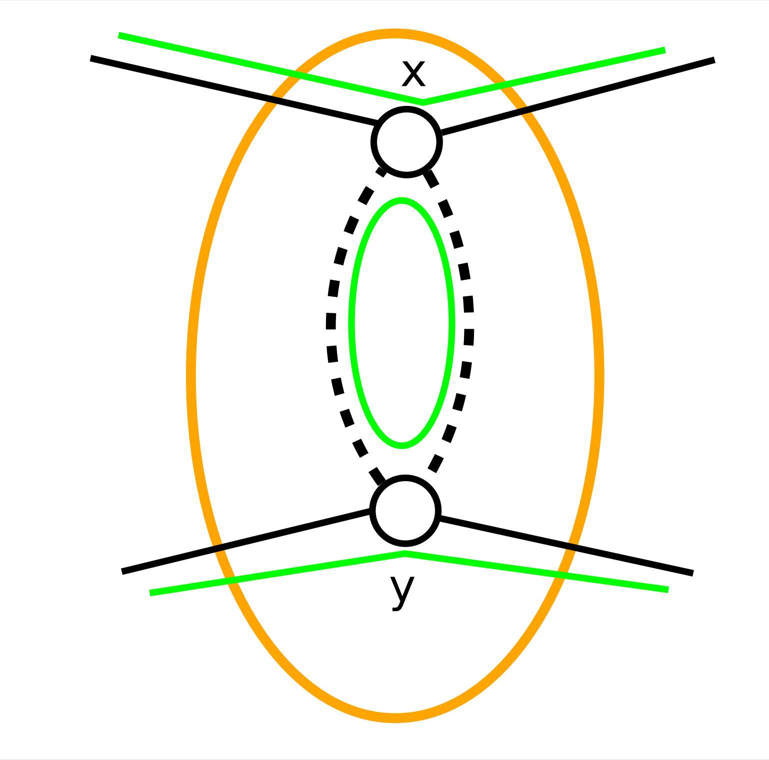
These are a connected component in any tour

Gadget for Inequality



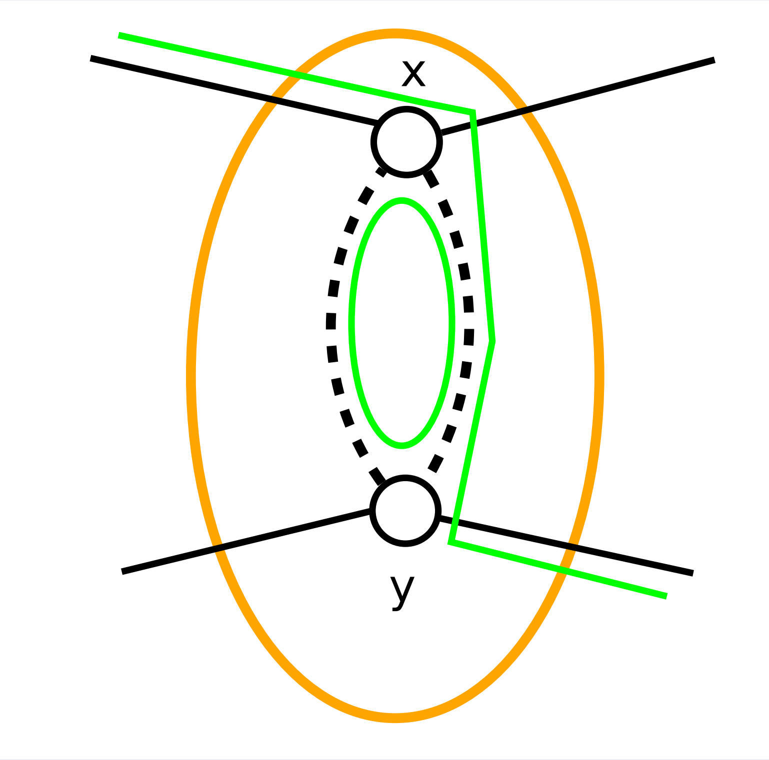
This is a good and honest assignment

Gadget for Inequality



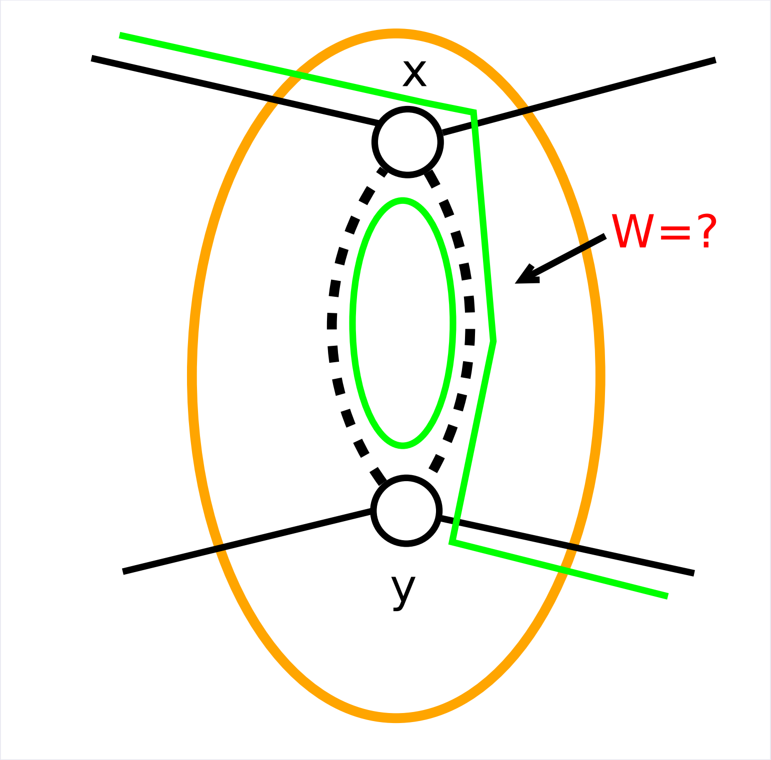
This is a bad and honest assignment

Gadget for Inequality



This is a PROBLEM!

Gadget for Inequality



Good news: Making this edge expensive fixes the problem.
Bad news: making this edge expensive adds overhead to the construction.

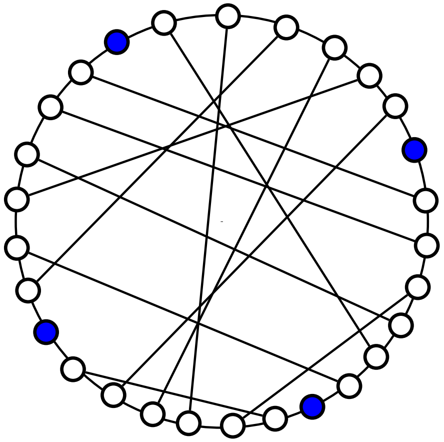
What is the smallest possible W?

The problem with inequality

- We want to use an inequality gadget to represent the matching edges of the amplifier.
- Normally, amplifier edges become equalities.

The problem with inequality

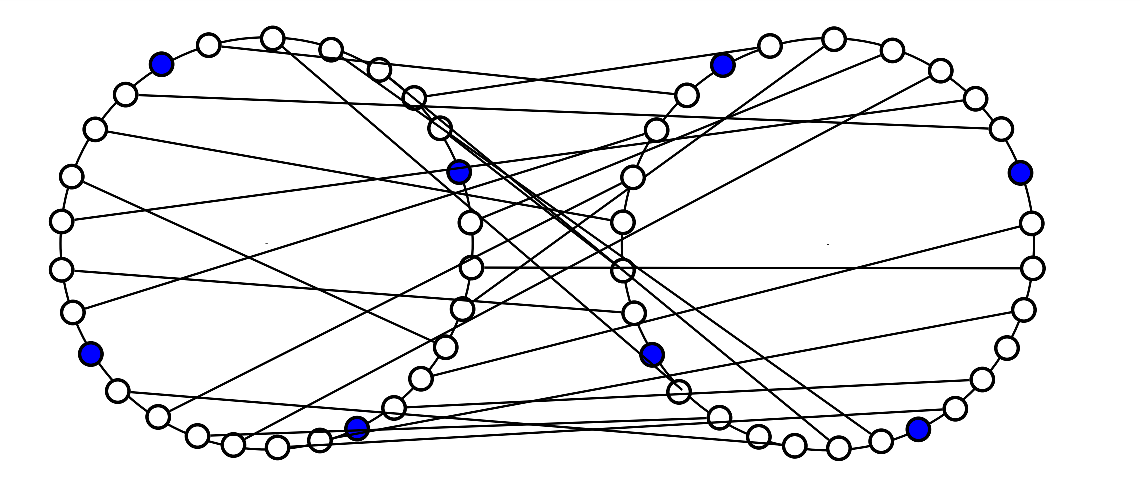
- We want to use an inequality gadget to represent the matching edges of the amplifier.
- Normally, amplifier edges become equalities.



We want cycle edges to remain equalities.

The problem with inequality

- We want to use an inequality gadget to represent the matching edges of the amplifier.
- Normally, amplifier edges become equalities.



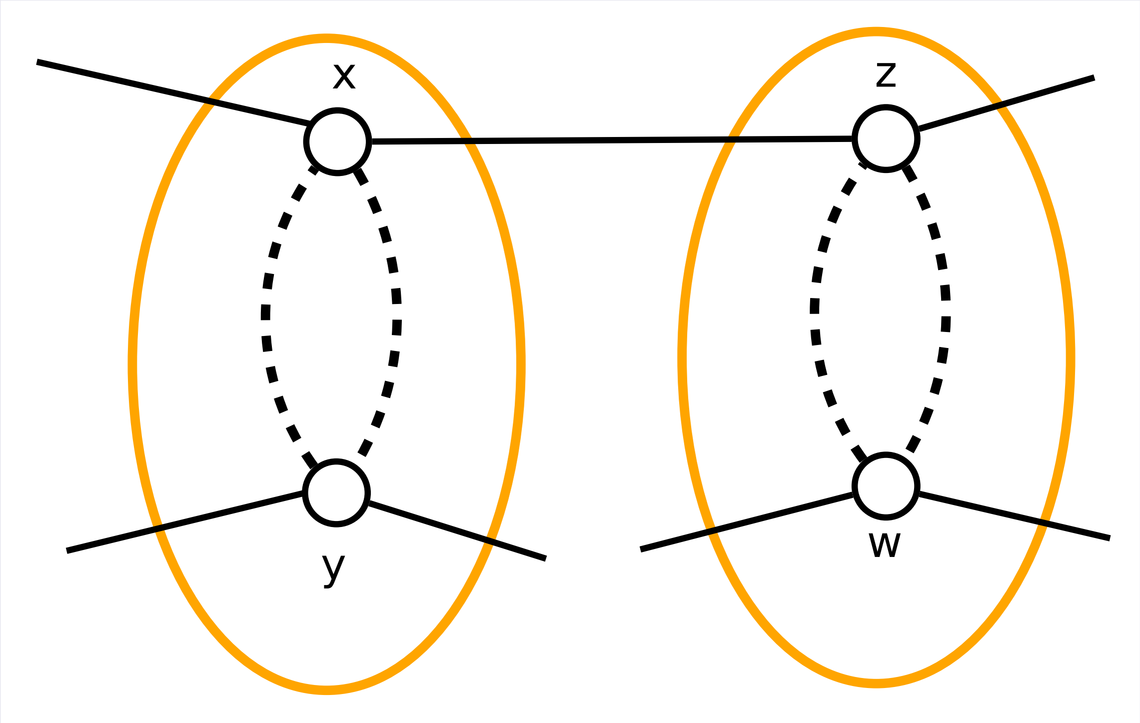
Solution: the bi-wheel!

Free equations!

Main idea: honesty gives equality

Free equations!

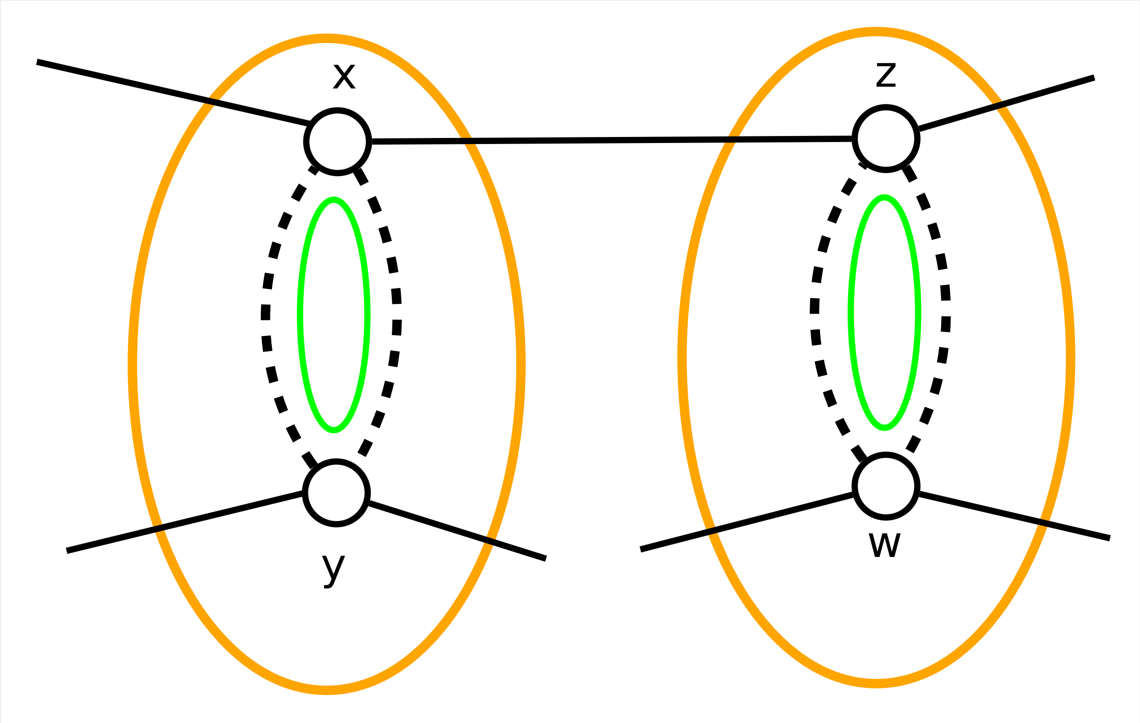
Main idea: honesty gives equality



Consider two vertices consecutive in one cycle (x, z)

Free equations!

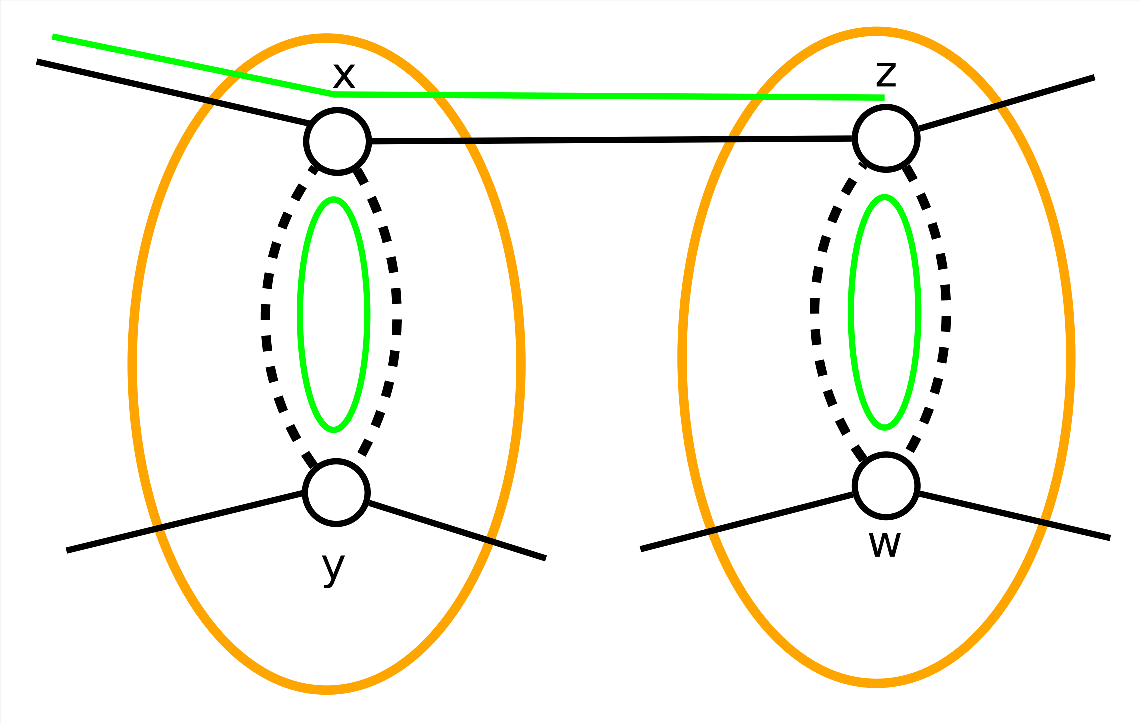
Main idea: honesty gives equality



Suppose that their matching gadgets are honest

Free equations!

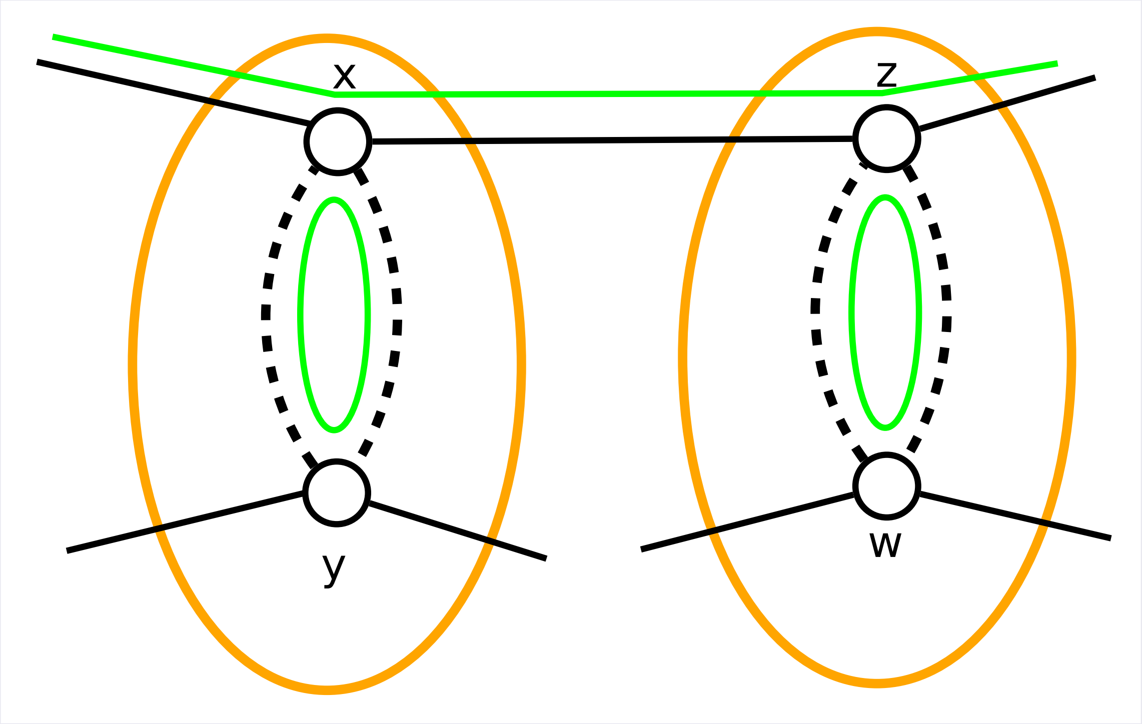
Main idea: honesty gives equality



Then if one is traversed as True. . .

Free equations!

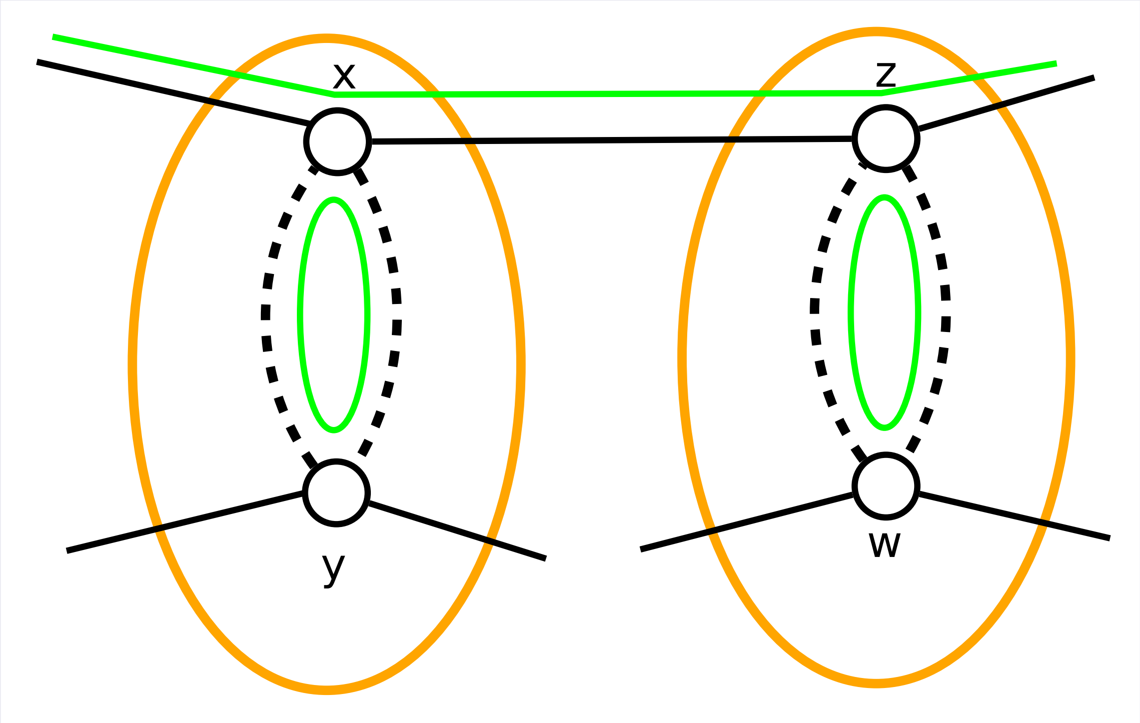
Main idea: honesty gives equality



. . . the other is also!

Free equations!

Main idea: honesty gives equality

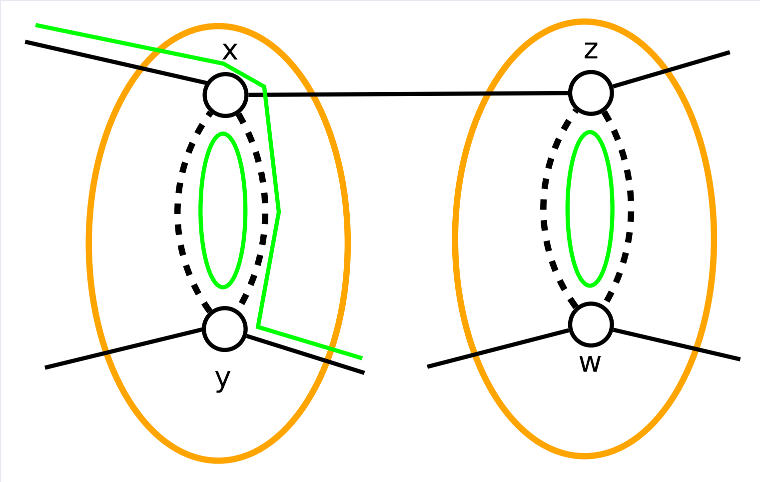


. . . the other is also!

- In other words, we extract an assignment for x by setting it to 1 iff both its incident non-forced edges are used.

Some handwaving

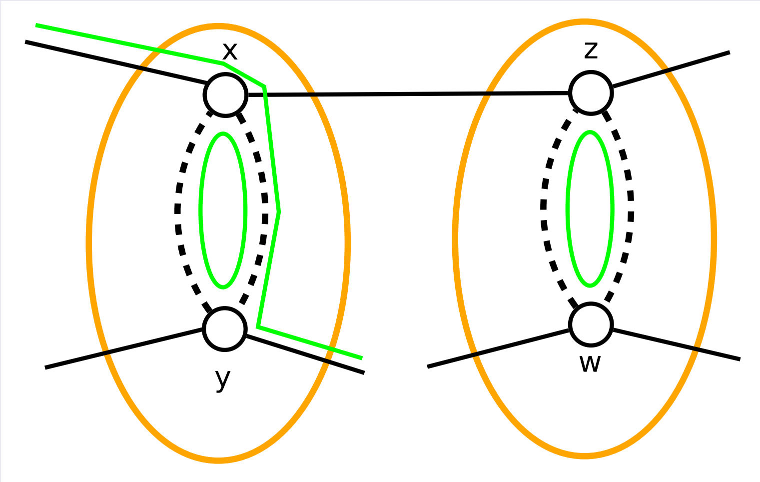
What is the cost of the forced edges?



- In case of dishonest traversal we must make the tour pay for all unsatisfied equations.

Some handwaving

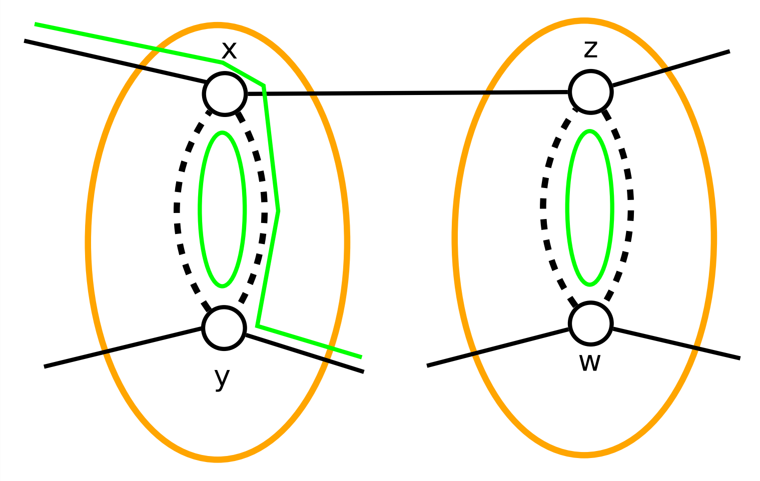
What is the cost of the forced edges?



- In case of dishonest traversal we must make the tour pay for all unsatisfied equations.
- There are 5 affected equation.

Some handwaving

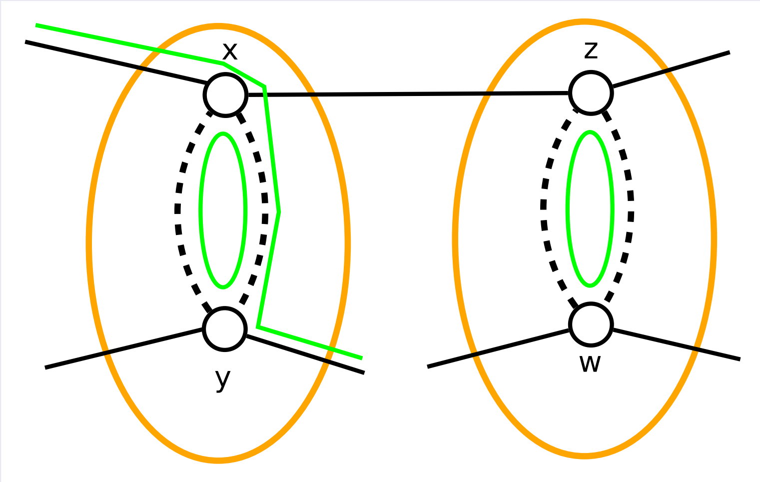
What is the cost of the forced edges?



- In case of dishonest traversal we must make the tour pay for all unsatisfied equations.
- There are 5 affected equation.
- We can always satisfy 3.

Some handwaving

What is the cost of the forced edges?



- In case of dishonest traversal we must make the tour pay for all unsatisfied equations.
- There are 5 affected equation.
- We can always satisfy 3.
- Hence, cost of forced edges is 2.

More handwaving

- For size-three equations we come up with some gadget (not shown).
- Some work needs to be done to ensure connectivity.
- Similar ideas can be used for ATSP.

More handwaving

- For size-three equations we come up with some gadget (not shown).
- Some work needs to be done to ensure connectivity.
- Similar ideas can be used for ATSP.

Theorem:
There is no $\frac{123}{122} - \epsilon$ approximation algorithm for TSP, unless P=NP.
There is no $\frac{75}{74} - \epsilon$ approximation algorithm for ATSP, unless P=NP.

Conclusions – Open problems

- A modular reduction for TSP and a better inapproximability threshold
  - But, constant still very low!

Future work

- Applications to other problems (Steiner Tree, Max 3-DM)
- Better amplifier constructions?

Conclusions – Open problems

- A modular reduction for TSP and a better inapproximability threshold
  - But, constant still very low!

Future work

- Applications to other problems (Steiner Tree, Max 3-DM)
- Better amplifier constructions?
- ... **Reasonable** inapproximability for TSP?

The end

Questions?