On Approximability of Bounded Degree Instances of Selected Optimization Problems

Richard Schmied

Dept. of Computer Science and
Hausdorff Center for Mathematics,
University of Bonn

Ph.D. Thesis Defense
July 25, 2013

universitätbonn

Introduction

We connect two seemingly faraway problems:

Introduction

We connect two seemingly faraway problems:

- **Traveling Salesman Problem (TSP)**

Introduction

We connect two seemingly faraway problems:

- **Traveling Salesman Problem (TSP)**
- **Algebraic problem of satisfying the maximum number of equations in a given system of linear equations (over finite fields)**

Introduction

We connect two seemingly faraway problems:

- **Traveling Salesman Problem (TSP)**
- **Algebraic problem of satisfying the maximum number of equations in a given system of linear equations (over finite fields)**

The first problem belongs to the fundamental and most important problems in combinatorial optimization.

Introduction

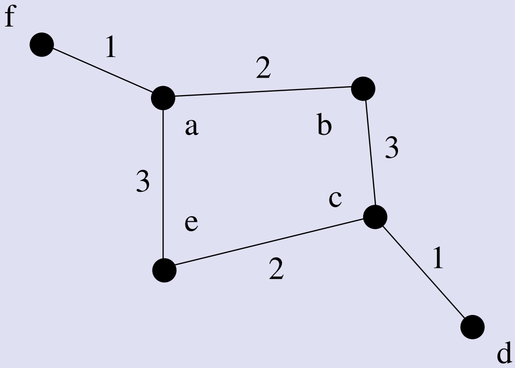
We connect two seemingly faraway problems:

- **Traveling Salesman Problem (TSP)**
- **Algebraic problem of satisfying the maximum number of equations in a given system of linear equations (over finite fields)**

The first problem belongs to the fundamental and most important problems in combinatorial optimization. (formal definition follows later for both problems)

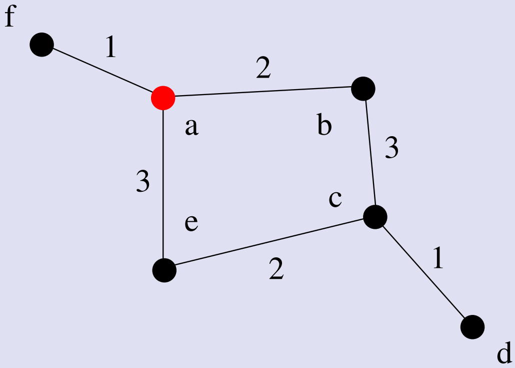
Introduction

An instance of the Metric TSP (shortest-path metric)



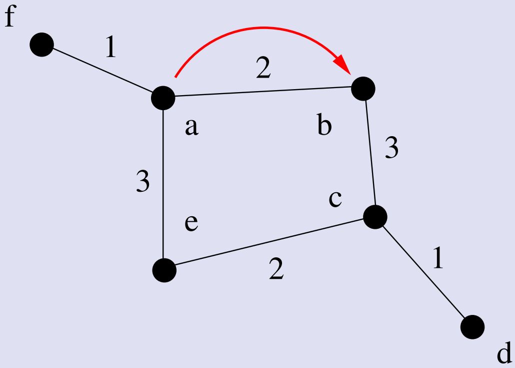
Introduction

An instance of the Metric TSP (shortest-path metric)



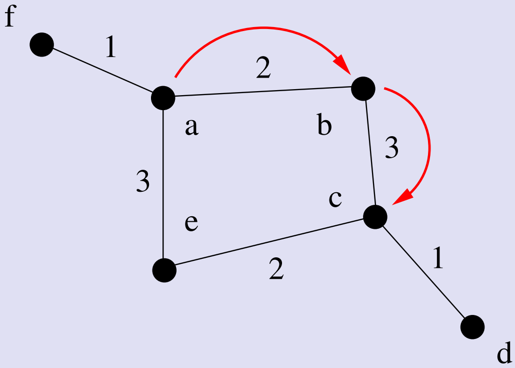
Introduction

An instance of the Metric TSP (shortest-path metric)

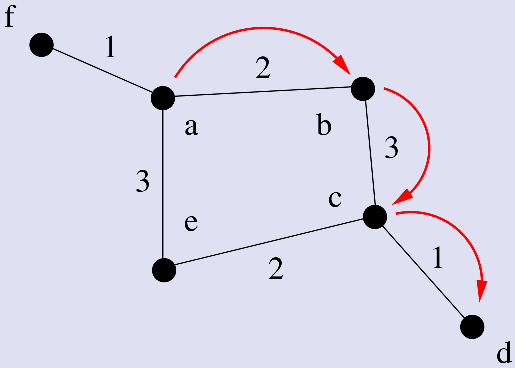


Introduction

An instance of the Metric TSP (shortest-path metric)

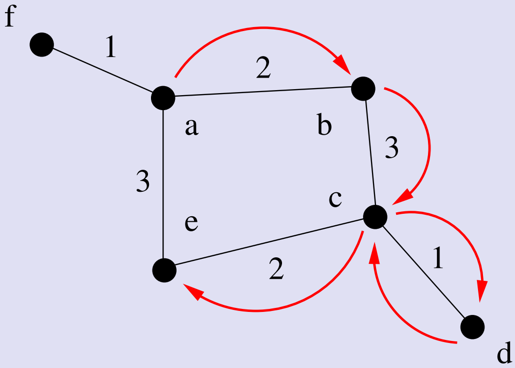


An instance of the Metric TSP (shortest-path metric)



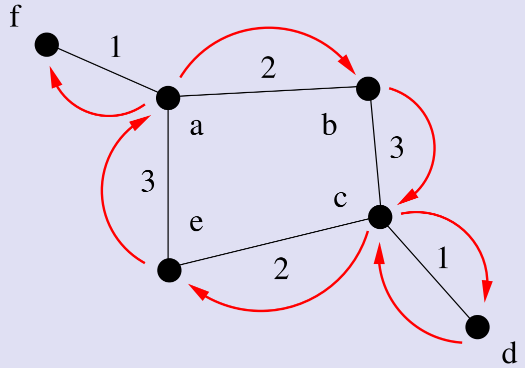
Introduction

An instance of the Metric TSP (shortest-path metric)



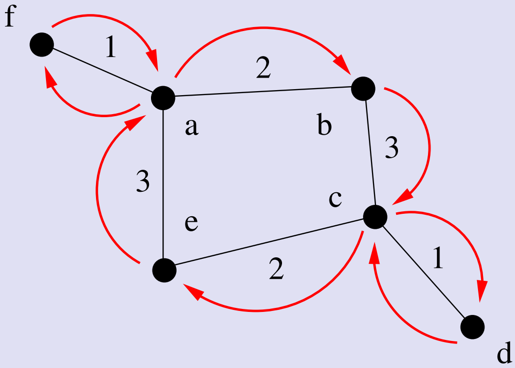
Introduction

An instance of the Metric TSP (shortest-path metric)



Introduction

An instance of the Metric TSP (shortest-path metric)



Introduction

Definition (Metric TSP)

Input:

Objective:

universitätbonn

Introduction

Definition (Metric TSP)

Input: A metric space (V, d) (weighted graph, shortest path metric)

Objective:

Introduction

Definition (Metric TSP)

Input: A metric space (V, d) (weighted graph, shortest path metric)

Objective: Find an ordering of the points $v_1, v_2, \ldots, v_n$ such that $d(v_1, v_2) + d(v_2, v_3) + \ldots + d(v_n, v_1)$ is minimized

Metric TSP

The Metric TSP

- NP-hardness proved by Karp in 1972 – Leaving less hope for efficient algorithms solving the Metric TSP to optimality

universitätbonn

Metric TSP

The Metric TSP

- NP-hardness proved by Karp in 1972 – Leaving less hope for efficient algorithms solving the Metric TSP to optimality
- Best known efficient approximation algorithm achieves a factor 3/2 [Christofides'76]

universitätbonn

Metric TSP

The Metric TSP

- NP-hardness proved by Karp in 1972 – Leaving less hope for efficient algorithms solving the Metric TSP to optimality
- Best known efficient approximation algorithm achieves a factor 3/2 [Christofides'76]
- APX-hard [Papadimitriou & Yannakakis'93]

Metric TSP

The Metric TSP

- NP-hardness proved by Karp in 1972 – Leaving less hope for efficient algorithms solving the Metric TSP to optimality
- Best known efficient approximation algorithm achieves a factor 3/2 [Christofides'76]
- APX-hard [Papadimitriou & Yannakakis'93]
- First explicit inapproximability bound: 5381/5380 [Engebretsen'00]

Metric TSP

Explicit inapproximability constants for the Metric TSP

- 3813/3812 [Böckenhauer et al.'00]

universitätbonn

Metric TSP

Explicit inapproximability constants for the Metric TSP

- 3813/3812 [Böckenhauer et al.'00]
- 389/388 [Engebretsen & Karpinski'01]

universitätbonn

Metric TSP

Explicit inapproximability constants for the Metric TSP

- 3813/3812 [Böckenhauer et al.'00]
- 389/388 [Engebretsen & Karpinski'01]
- 220/119 [Papadimitriou & Vempala'06]

Metric TSP

Explicit inapproximability constants for the Metric TSP

- 3813/3812 [Böckenhauer et al.'00]
- 389/388 [Engebretsen & Karpinski'01]
- 220/119 [Papadimitriou & Vempala'06]
- 185/184 [Lampis'12]

universitätbonn

Metric TSP

Explicit inapproximability constants for the Metric TSP

- 3813/3812 [Böckenhauer et al.'00]
- 389/388 [Engebretsen & Karpinski'01]
- 220/119 [Papadimitriou & Vempala'06]
- 185/184 [Lampis'12]

Theorem (Karpinski, Lampis & S.'13)

It is NP-hard to approximate the Metric TSP within any factor less than 123/122.

The Reduction (Metric TSP)

The Reduction (Metric TSP)

Starting point: Inapproximability result for MAX-E3LIN2

universitätbonn

The Reduction (Metric TSP)

Starting point: Inapproximability result for MAX-E3LIN2

Definition (MAX-E3LIN2)

Input: A system $\mathscr{L}$ of linear equations mod 2, in which equations are of the form $x_i \oplus x_j \oplus x_k = b$ with $b \in \{0, 1\}$

Output: An assignment to the variables in $\mathscr{L}$ that maximizes the number of satisfied equations

The Reduction (Metric TSP)

Starting point: Inapproximability result for MAX-E3LIN2

Definition (MAX-E3LIN2)

Input: A system $\mathscr{L}$ of linear equations mod 2, in which equations are of the form $x_i \oplus x_j \oplus x_k = b$ with $b \in \{0, 1\}$

Output: An assignment to the variables in $\mathscr{L}$ that maximizes the number of satisfied equations

Approximation lower bound: MAX-E3LIN2 is NP-hard to approximate to within any factor less than 2. [Håstad'01]

universitätbonn

The Reduction (Metric TSP)

High-level view of the reduction:

- Construct a reduction from MAX-E3LIN2 to Metric TSP ($\mathscr{L} \rightarrow$ TSP instance)

The Reduction (Metric TSP)

High-level view of the reduction:

- Construct a reduction from MAX-E3LIN2 to Metric TSP ($\mathscr{L} \rightarrow$ TSP instance)
- Reduction is easier if the number of occurrences of each variable in $\mathscr{L}$ is bounded by a constant (to control the consistency of variable gadgets)

universitätbonn

The Reduction (Metric TSP)

High-level view of the reduction:

- $\rightarrow$ We need inapproximability results for MAX-E3LIN2 with bounded number of occurrences of variables

The Reduction (Metric TSP)

High-level view of the reduction:

- $\rightarrow$ We need inapproximability results for MAX-E3LIN2 with bounded number of occurrences of variables **(Intermediate problem: MAX-E3occ-LIN2)**

universitätbonn

The Reduction (Metric TSP)

High-level view of the reduction:

- $\rightarrow$ We need inapproximability results for MAX-E3LIN2 with bounded number of occurrences of variables **(Intermediate problem: MAX-E3occ-LIN2)**
- Sparse instance methods **(amplifier graphs)** [Berman&Karpinski'99]

The Reduction (Metric TSP)

High-level view of the reduction:

- $\rightarrow$ We need inapproximability results for MAX-E3LIN2 with bounded number of occurrences of variables **(Intermediate problem: MAX-E3occ-LIN2)**
- Sparse instance methods **(amplifier graphs)** [Berman&Karpinski'99]
  $\rightarrow$ Prove inapproximability for MAX-E3occ-LIN2

The Reduction (Metric TSP)

High-level view of the reduction:

- $\rightarrow$ We need inapproximability results for MAX-E3LIN2 with bounded number of occurrences of variables **(Intermediate problem: MAX-E3occ-LIN2)**
- Sparse instance methods **(amplifier graphs)** [Berman&Karpinski'99]
  $\rightarrow$ Prove inapproximability for MAX-E3occ-LIN2
  $\rightarrow$ Prove our result for Metric TSP

universitätbonn

Sparse Instance Methods

Sparse Instance Methods

First approach: Use expander graphs to decrease the number of occurrences of variables:

Sparse Instance Methods

First approach: Use expander graphs to decrease the number of occurrences of variables:

- Restrict ourselves to expander with **maximum degree** Δ bounded by a small constant

universitätbonn

Sparse Instance Methods

First approach: Use expander graphs to decrease the number of occurrences of variables:

- Restrict ourselves to expander with **maximum degree** Δ bounded by a small constant
- Main property: In any partition of the vertices into two sets $(S, V \backslash S)$, there are **many edges crossing** from S to $V \backslash S$

universitätbonn

Sparse Instance Methods

First approach: Use expander graphs to decrease the number of occurrences of variables:

- Restrict ourselves to expander with **maximum degree** Δ bounded by a small constant
- Main property: In any partition of the vertices into two sets $(S, V \backslash S)$, there are **many edges crossing** from S to $V \backslash S$

This is achieved even though the graph has only few edges!

universitätbonn

Sparse Instance Methods

First approach: Use expander graphs to decrease the number of occurrences of variables:

- Restrict ourselves to expander with **maximum degree** Δ bounded by a small constant
- Main property: In any partition of the vertices into two sets $(S, V\setminus S)$, there are **many edges crossing** from S to $V\setminus S$

This is achieved even though the graph has only few edges!

Definition (Strong expander)

A graph $G = (V, E)$ is a **strong expander** if for all $S \subseteq V$ with $|S| \leq |V|/2$, we have that $|\{e \in E \mid |e \cap S| = 1\}| \geq |S|$.

universitätbonn

Sparse Instance Methods

Construction for reducing the number of occurrences

For each variable x:

Sparse Instance Methods

Construction for reducing the number of occurrences

For each variable x:

- Let n be the number of occurrences of x in $\mathscr{L}$:
  Replace the ith occurrence of the variable x with a new variable x_i

universitätbonn

Sparse Instance Methods

Construction for reducing the number of occurrences

For each variable x:

- Let n be the number of occurrences of x in $\mathscr{L}$:
  Replace the ith occurrence of the variable x with a new variable x_i
- Construct a strong expander G with vertices $\{1, 2, \ldots, n\}$

universitätbonn

Sparse Instance Methods

Construction for reducing the number of occurrences

For each variable x:

- Let n be the number of occurrences of x in $\mathscr{L}$:
  Replace the ith occurrence of the variable x with a new variable x_i
- Construct a strong expander G with vertices $\{1, 2, \ldots, n\}$
- For each edge $\{i, j\}$ in G, add the equation: $x_i \oplus x_j = 0$

universitätbonn

Sparse Instance Methods

Construction for reducing the number of occurrences

For each variable x:

- Let n be the number of occurrences of x in $\mathscr{L}$:
  Replace the ith occurrence of the variable x with a new variable x_i
- Construct a strong expander G with vertices $\{1, 2, \ldots, n\}$
- For each edge $\{i, j\}$ in G, add the equation: $x_i \oplus x_j = 0$

Note: $x \oplus y = 0$ if and only if $x = y$ (**equality equation**)

universitätbonn

Optimal assignments are consistent

1. Suppose that in the new instance the optimal assignment sets some of the x_i's to 0 and others to $\mathbf{1}$ $\rightarrow$ partition of the strong expander
2.
3.

universitätbonn

Optimal assignments are consistent

1. Suppose that in the new instance the optimal assignment sets some of the x_i's to 0 and others to $\mathbf{1} \rightarrow$ partition of the strong expander (Note: Each edge crossing the partition corresponds to an unsatisfied equation!)

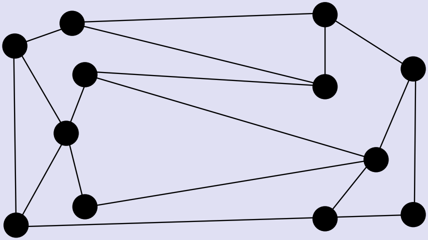
2.

3.

universitätbonn

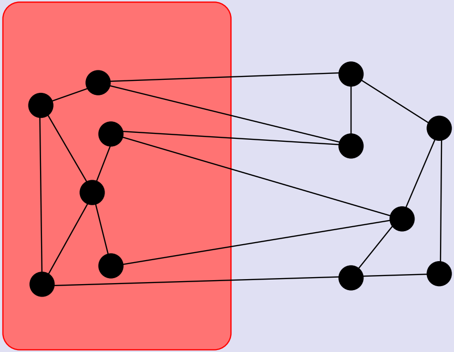
Sparse Instance Methods

Edges crossing from S to $V \setminus S$



Sparse Instance Methods

Edges crossing from S to $V \setminus S$



Optimal assignments are consistent

1. Suppose that in the new instance the optimal assignment sets some of the x_i's to 0 and others to $\mathbf{1}$ $\rightarrow$ partition of the strong expander (Note: Each edge crossing the partition corresponds to an unsatisfied equation!)

2. $\rightarrow$ it is always optimal to give the same value to all x_i's. Because bounded-degree expander graphs are sparse, we added only a linear number of equations.

universitätbonn

Optimal assignments are consistent

1. Suppose that in the new instance the optimal assignment sets some of the x_i's to 0 and others to $\mathbf{1}$ $\rightarrow$ partition of the strong expander (Note: Each edge crossing the partition corresponds to an unsatisfied equation!)
2. $\rightarrow$ it is always optimal to give the same value to all x_i's. Because bounded-degree expander graphs are sparse, we added only a linear number of equations.
3. This gives some inapproximability factor

universitätbonn

Sparse Instance Methods

Unfortunately:

For $\Delta < 6$, strong expander are yet not known to exist!

universitätbonn

Sparse Instance Methods

Unfortunately:

For $\Delta < 6$, strong expander are yet not known to exist!

Second approach: we use amplifier graphs instead

universitätbonn

Sparse Instance Methods

Unfortunately:

For $\Delta < 6$, strong expander are yet not known to exist!

Second approach: we use amplifier graphs instead

Amplifier graphs

- **Amplifier graphs** are strong expander graphs for a certain subset of vertices (contact vertices)

Sparse Instance Methods

Unfortunately:

For $\Delta < 6$, strong expander are yet not known to exist!

Second approach: we use amplifier graphs instead

Amplifier graphs

- **Amplifier graphs** are strong expander graphs for a certain subset of vertices (contact vertices)
- The other vertices are thrown in to make consistency easier to achieve (checker vertices)

Sparse Instance Methods

Unfortunately:

For $\Delta < 6$, strong expander are yet not known to exist!

Second approach: we use amplifier graphs instead

Amplifier graphs

- **Amplifier graphs** are strong expander graphs for a certain subset of vertices (contact vertices)
- The other vertices are thrown in to make consistency easier to achieve (checker vertices)
- This allows us to get smaller Δ

universitätbonn

Special class of amplifier graphs with $\Delta = 3$:

Wheel amplifier graphs [Berman & Karpinski'99]

Construction:

- Start with a cycle on $7n$ vertices

Sparse Instance Methods

Special class of amplifier graphs with $\Delta = 3$:

Wheel amplifier graphs [Berman & Karpinski'99]

Construction:

- Start with a cycle on $7n$ vertices
- Every seventh vertex is a contact vertex

universitätbonn

Sparse Instance Methods

Special class of amplifier graphs with $\Delta = 3$:

Wheel amplifier graphs [Berman & Karpinski'99]

Construction:

- Start with a cycle on $7n$ vertices
- Every seventh vertex is a contact vertex
- Other vertices are checker vertices

Sparse Instance Methods

Special class of amplifier graphs with $\Delta = 3$:

Wheel amplifier graphs [Berman & Karpinski'99]

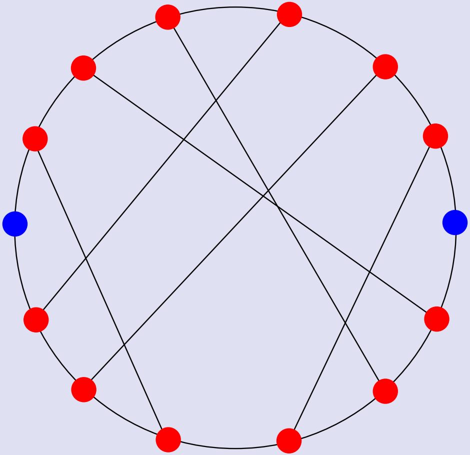
Construction:

- Start with a cycle on $7n$ vertices
- Every seventh vertex is a contact vertex
- Other vertices are checker vertices
- There is a perfect matching on the set of checker vertices

universitätbonn

Sparse Instance Methods

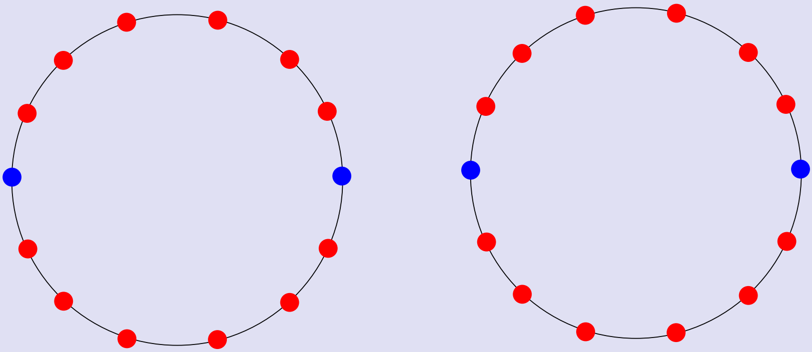
Wheel amplifier graph



Sparse Instance Methods

Amplifier graphs tailored for the Metric TSP:

Bi-wheel amplifier graph [Karpinski, Lampis & S.'13]

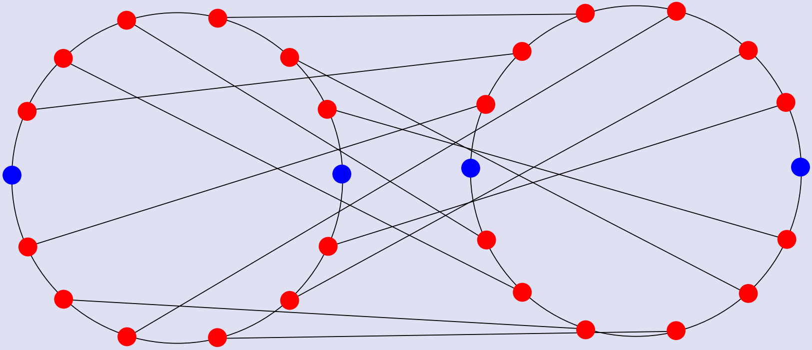


universitätbonn

Sparse Instance Methods

Amplifier graphs tailored for the Metric TSP:

Bi-wheel amplifier graph [Karpinski, Lampis & S.'13]



universitätbonn

Sparse Instance Methods

Construct the intermediate instance $\mathscr{L}_{3occ}$, in which each variable appears in exactly 3 equations:

- Variables x corresponding to contact vertices appear in equations with three variables $x \oplus y \oplus z = b$

Sparse Instance Methods

Construct the intermediate instance $\mathscr{L}_{3occ}$, in which each variable appears in exactly 3 equations:

- Variables x corresponding to contact vertices appear in equations with three variables $x \oplus y \oplus z = b$
- Each cycle edge $\{i, i+1\}$: $x_i \oplus x_{i+1} = 0$ (equality eqn)

Sparse Instance Methods

Construct the intermediate instance $\mathscr{L}_{3occ}$, in which each variable appears in exactly 3 equations:

- Variables x corresponding to contact vertices appear in equations with three variables $x \oplus y \oplus z = b$
- Each cycle edge $\{i, i+1\}$: $x_i \oplus x_{i+1} = 0$ (equality eqn)
- Each matching edge $\{i, j\}$: $x_i \oplus x_j = 1$ (inequality eqn)

universitätbonn

Sparse Instance Methods

Construct the intermediate instance $\mathscr{L}_{3occ}$, in which each variable appears in exactly 3 equations:

- Variables x corresponding to contact vertices appear in equations with three variables $x \oplus y \oplus z = b$
- Each cycle edge $\{i, i+1\}$: $x_i \oplus x_{i+1} = 0$ (equality eqn)
- Each matching edge $\{i, j\}$: $x_i \oplus x_j = 1$ (inequality eqn)

Note: $x \oplus y = 1$ if and only if $x \neq y$ (**inequality equation**)

universitätbonn

The Reduction (Metric TSP) cont'd

The Reduction (Metric TSP)

Construction ($\mathscr{L}_{3occ} \rightarrow$ TSP instance):

Given an instance $\mathscr{L}_{3occ}$,

- For each variable, create a vertex

universitätbonn

The Reduction (Metric TSP)

Construction ($\mathscr{L}_{3occ} \rightarrow$ TSP instance):

Given an instance $\mathscr{L}_{3occ}$,

- For each variable, create a vertex
- For each equality equation, create an edge

The Reduction (Metric TSP)

Construction ($\mathscr{L}_{3occ} \rightarrow$ TSP instance):

Given an instance $\mathscr{L}_{3occ}$,

- For each variable, create a vertex
- For each equality equation, create an edge
- For each inequality equation, add an **inequality gadget**

The Reduction (Metric TSP)

Construction ($\mathscr{L}_{3occ} \to$ TSP instance):

Given an instance $\mathscr{L}_{3occ}$,

- For each variable, create a vertex
- For each equality equation, create an edge
- For each inequality equation, add an **inequality gadget** (will be shown on the next slide)

The Reduction (Metric TSP)

Construction ($\mathscr{L}_{3occ} \to$ TSP instance):

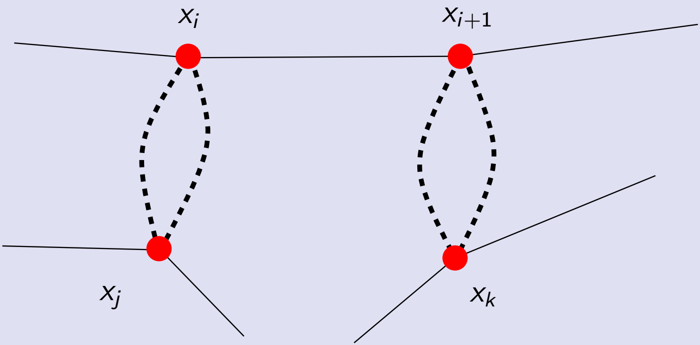
Given an instance $\mathscr{L}_{3occ}$,

- For each variable, create a vertex
- For each equality equation, create an edge
- For each inequality equation, add an **inequality gadget** (will be shown on the next slide)
- Add gadgets for equations with 3 variables (containing the contact vertices)

The Reduction (Metric TSP)

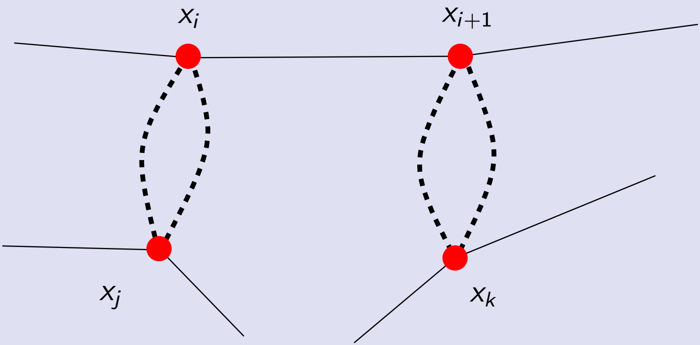
Forced edge: any tour is forced to use this edge at least once

Inequality gadgets for $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



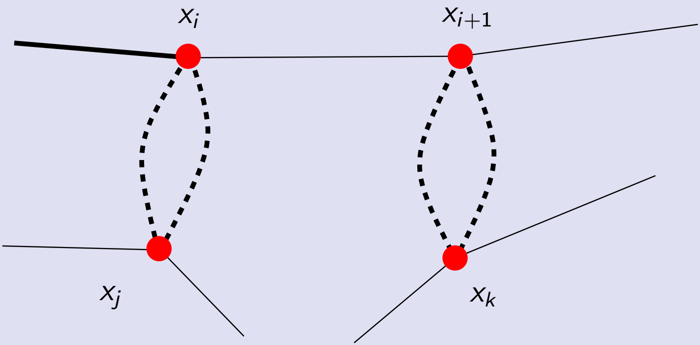
The Reduction (Metric TSP)

Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



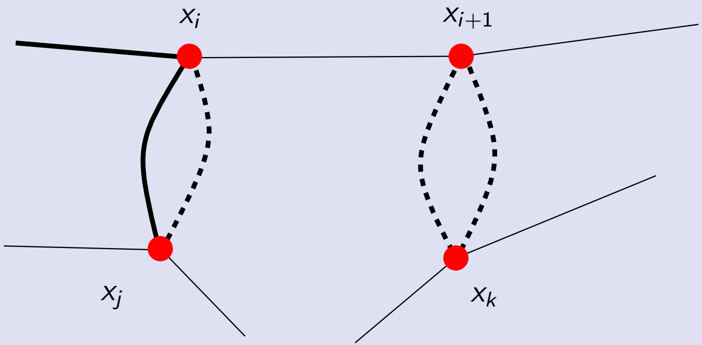
The Reduction (Metric TSP)

Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



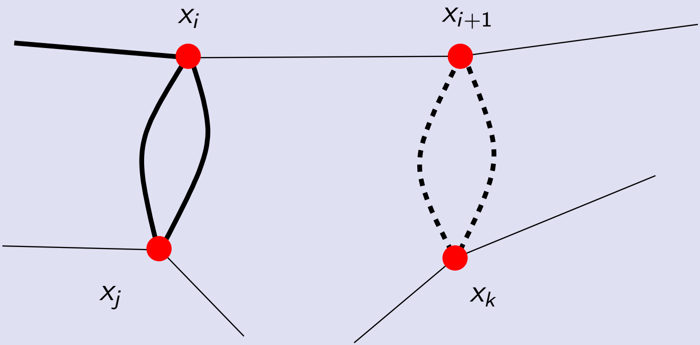
The Reduction (Metric TSP)

Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



The Reduction (Metric TSP)

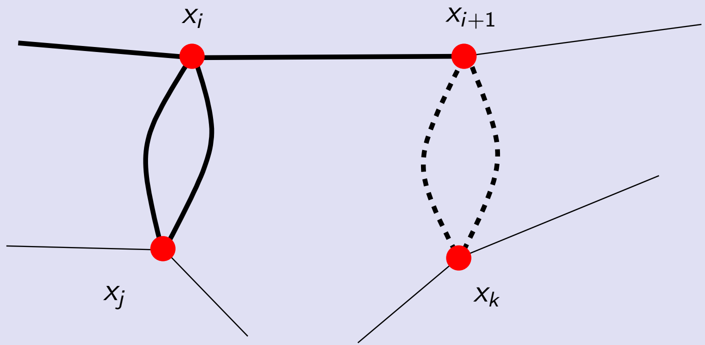
Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



universitätbonn

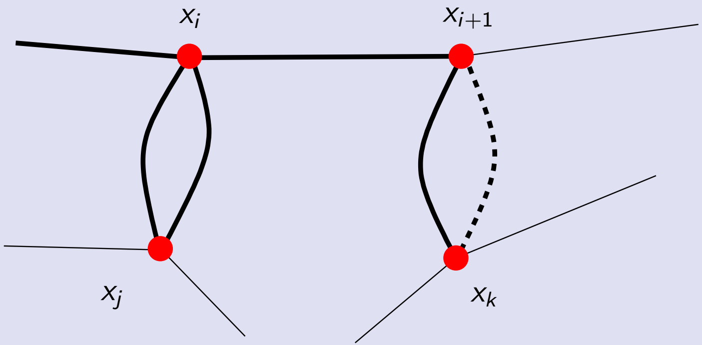
The Reduction (Metric TSP)

Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



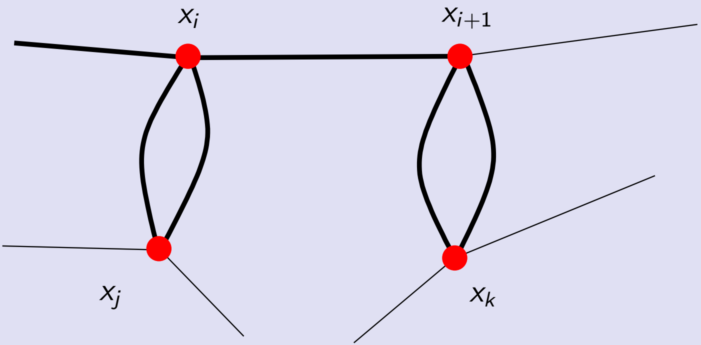
The Reduction (Metric TSP)

Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



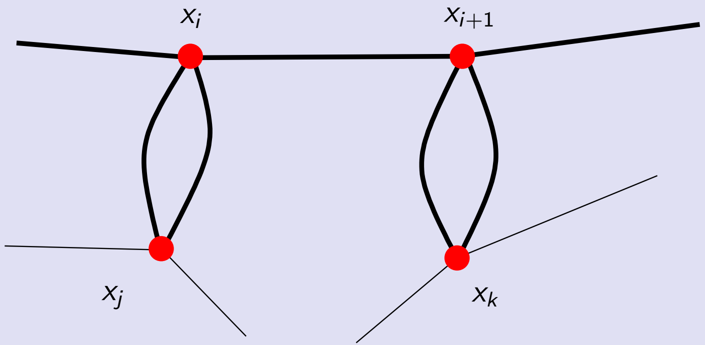
The Reduction (Metric TSP)

Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ and $x_{i+1} \oplus x_k = 1$



The Reduction (Metric TSP)

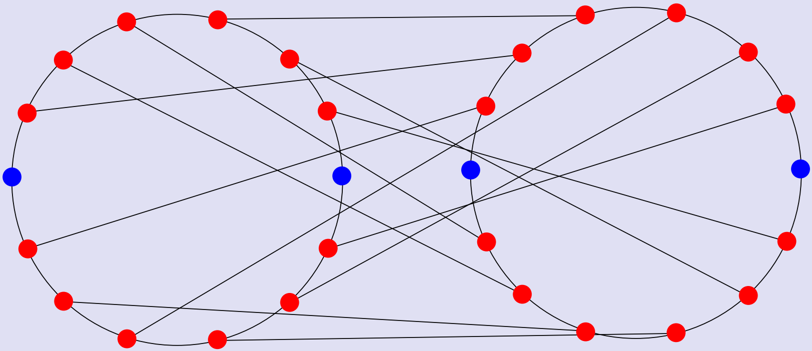
Simulating $x_i \oplus x_{i+1} = 0$, $x_i \oplus x_j = 1$ **and** $x_{i+1} \oplus x_k = 1$



universitätbonn

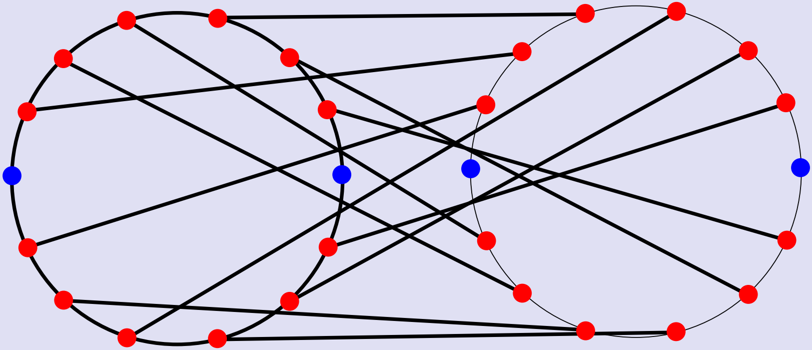
The Reduction (Metric TSP)

Sub-tour visiting all contact vertices of one cycle



The Reduction (Metric TSP)

Sub-tour visiting all contact vertices of one cycle



The Reduction (Metric TSP)

Final remarks:

- For equations with 3 variables, we construct a more efficient gadget (not shown)

universitätbonn

The Reduction (Metric TSP)

Final remarks:

- For equations with 3 variables, we construct a more efficient gadget (not shown)
- Some work needs to be done to ensure connectivity

The Reduction (Metric TSP)

Final remarks:

- For equations with 3 variables, we construct a more efficient gadget (not shown)
- Some work needs to be done to ensure connectivity
- Similar ideas can be used for Asymmetric TSP

universitätbonn

Asymmetric TSP

Asymmetric TSP

Definition (Asymmetric TSP)

Input:

Objective:

Asymmetric TSP

Definition (Asymmetric TSP)

Input: An asymmetric metric space (V, d) (arc weighted digraph)

Objective:

universitätbonn

Asymmetric TSP

Definition (Asymmetric TSP)

Input: An asymmetric metric space (V, d) (arc weighted digraph)

Objective: Find a tour in (V, d) with minimum length

universitätbonn

Asymmetric TSP

Theorem (Papadimitriou & Vempala (STOC'00))

It is NP-hard to approximate the Asymmetric TSP to within any factor less than 117/116.

By using our bi-wheel amplifier methods, we obtain:

universitätbonn

Asymmetric TSP

Theorem (Papadimitriou & Vempala (STOC'00))

It is NP-hard to approximate the Asymmetric TSP to within any factor less than 117/116.

By using our bi-wheel amplifier methods, we obtain:

Theorem (Karpinski, Lampis & S.'13)

It is NP-hard to approximate the Asymmetric TSP to within any factor less than 75/74.

Asymmetric TSP

Theorem (Papadimitriou & Vempala (STOC'00))

It is NP-hard to approximate the Asymmetric TSP to within any factor less than 117/116.

By using our bi-wheel amplifier methods, we obtain:

Theorem (Karpinski, Lampis & S.'13)

It is NP-hard to approximate the Asymmetric TSP to within any factor less than 75/74.

First improvement after more than a decade!

TSP with Distances 1 and 2

TSP with Distances 1 and 2

Definition (TSP with distances 1 and 2 ((1,2)-TSP))

Input:

Objective:

TSP with Distances 1 and 2

Definition (TSP with distances 1 and 2 ((1,2)-TSP))

Input: A graph $G = (V, E)$

Objective:

TSP with Distances 1 and 2

Definition (TSP with distances 1 and 2 ((1,2)-TSP))

Input: A graph $G = (V, E)$

Objective: Find a tour with minimum length

$d(v_i, v_j) = 1$ if $\{v_i, v_j\} \in E$ and 2 otherwise

universitätbonn

TSP with Distances 1 and 2

Theorem (Engebretsen & Karpinski (ICALP'01))

It is NP-hard to approximate the (1,2)-TSP to within any factor less than 741/740.

By using wheel amplifier graphs combined with "parity gadgets", we obtain:

universitätbonn

TSP with Distances 1 and 2

Theorem (Engebretsen & Karpinski (ICALP'01))

It is NP-hard to approximate the (1,2)-TSP to within any factor less than 741/740.

By using wheel amplifier graphs combined with "parity gadgets", we obtain:

Theorem (Karpinski & S.'12)

It is NP-hard to approximate the (1,2)-TSP to within any factor less than 535/534.

Graphic TSP

Graphic TSP

Definition (Graphic TSP)

Input:

Objective:

Graphic TSP

Definition (Graphic TSP)

Input: A graph $G = (V, E)$

Objective:

Graphic TSP

Definition (Graphic TSP)

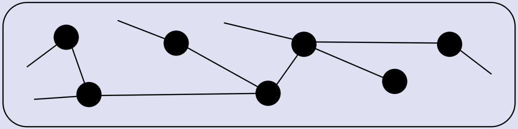
Input: A graph $G = (V, E)$

Objective: Find a tour with minimum length

$d(v_i, v_j)$ is defined by the shortest-path distance of v_i and v_j in G

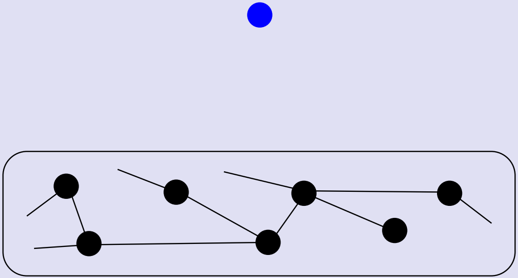
Graphic TSP

Reducing (1,2)-TSP to Graphic TSP



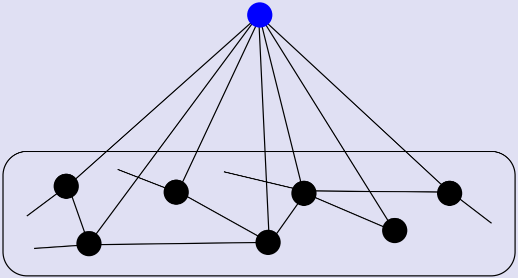
Graphic TSP

Reducing (1,2)-TSP to Graphic TSP



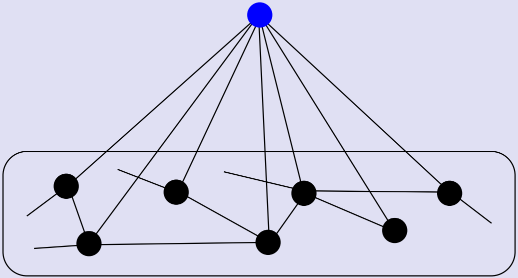
Graphic TSP

Reducing (1,2)-TSP to Graphic TSP



Graphic TSP

Reducing (1,2)-TSP to Graphic TSP



Implying the best up to now inapproximability factor for Graphic TSP

Subcubic & Cubic Instances

Comparison of inapproximability results

Problem	Previously best known	Our Result
(1,2)-TSP on cubic graphs	1291/1290 [CKK02]	
(1,2)-TSP on subcubic graphs	787/786 [CKK02]	
Graphic TSP on cubic graphs	–	
Graphic TSP on subcubic graphs	–	

Subcubic & Cubic Instances

Comparison of inapproximability results

Problem	Previously best known	Our Result
(1,2)-TSP on cubic graphs	1291/1290 [CKK02]	1141/1140
(1,2)-TSP on subcubic graphs	787/786 [CKK02]	
Graphic TSP on cubic graphs	–	
Graphic TSP on subcubic graphs	–	

universitätbonn

Subcubic & Cubic Instances

Comparison of inapproximability results

Problem	Previously best known	Our Result
(1,2)-TSP on cubic graphs	1291/1290 [CKK02]	1141/1140
(1,2)-TSP on subcubic graphs	787/786 [CKK02]	673/672
Graphic TSP on cubic graphs	–	
Graphic TSP on subcubic graphs	–	

universitätbonn

Subcubic & Cubic Instances

Comparison of inapproximability results

Problem	Previously best known	Our Result
(1,2)-TSP on cubic graphs	1291/1290 [CKK02]	1141/1140
(1,2)-TSP on subcubic graphs	787/786 [CKK02]	673/672
Graphic TSP on cubic graphs	–	1153/1152
Graphic TSP on subcubic graphs	–	

First inapproximability results at all!

universitätbonn

Subcubic & Cubic Instances

Comparison of inapproximability results

Problem	Previously best known	Our Result
(1,2)-TSP on cubic graphs	1291/1290 [CKK02]	1141/1140
(1,2)-TSP on subcubic graphs	787/786 [CKK02]	673/672
Graphic TSP on cubic graphs	–	1153/1152
Graphic TSP on subcubic graphs	–	685/684

First inapproximability results at all!

Further Results

Further Results

Further related results on approximability of the problems of:

- Shortest Superstring

Further Results

Further related results on approximability of the problems of:

- Shortest Superstring
- Maximum Compression

Further Results

Further related results on approximability of the problems of:

- Shortest Superstring
- Maximum Compression
- Steiner Tree with distances 1 and 2

Further Results

Further related results on approximability of the problems of:

- Shortest Superstring
- Maximum Compression
- Steiner Tree with distances 1 and 2
- Metric Dimension

Further Results

Further related results on approximability of the problems of:

- Shortest Superstring
- Maximum Compression
- Steiner Tree with distances 1 and 2
- Metric Dimension
- Hypergraph Vertex Cover

Conclusions:

- Reduction method for several TSP problems leading to improved inapproximability thresholds

universitätbonn

Conclusions:

- Reduction method for several TSP problems leading to improved inapproximability thresholds
- But, the inapproximability constants are still very low!

Conclusions:

- Reduction method for several TSP problems leading to improved inapproximability thresholds
- But, the inapproximability constants are still very low!

Further research:

- Improving the inapproximability bounds for the TSP and the Steiner Tree problem?

universitätbonn

Conclusions:

- Reduction method for several TSP problems leading to improved inapproximability thresholds
- But, the inapproximability constants are still very low!

Further research:

- Improving the inapproximability bounds for the TSP and the Steiner Tree problem?
- Better amplifier constructions?

universität**bonn**

Conclusions:

- Reduction method for several TSP problems leading to improved inapproximability thresholds
- But, the inapproximability constants are still very low!

Further research:

- Improving the inapproximability bounds for the TSP and the Steiner Tree problem?
- Better amplifier constructions?
- New global PCP-system constructions for TSP?

Further research:

On upper bounds side:

- Improving general upper approximation bound for metric TSP below 3/2 (1.50)?

∎

universitätbonn

Further research:

On upper bounds side:

- Improving general upper approximation bound for metric TSP below 3/2 (1.50)?
- Improving upper approximation bound for cubic Graphic TSP below 4/3 (1.33)?

∎

Thank You!

universitätbonn