A Study of Domino Parity and k-Parity Constraints for the TSP

William Cook Daniel Espinoza Marcos Goycoolea

School of Industrial and Systems Engineering
Georgia Institute of Technology

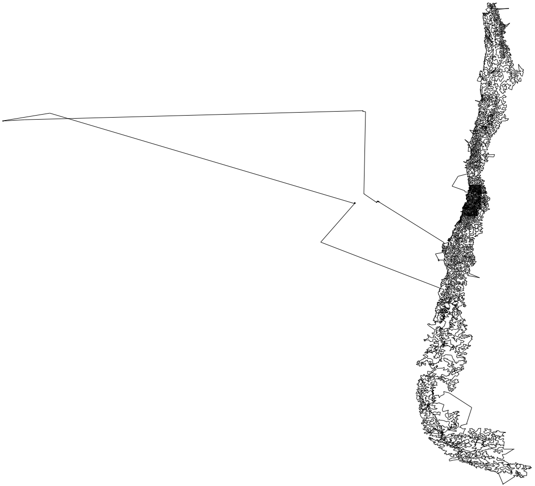
IPCO 2005

The Traveling Salesman Problem

$\min cx$

$$\sum_{e \in \delta(v)} x_e = 2 \quad \forall v \in V$$
$$\sum_{e \in \delta(S)} x_e \geq 2 \quad \forall S \subsetneq V$$

$$x_e \in \{0, 1\} \quad \forall e \in E$$



Background

“ Separating a Superclass of Comb Inequalities in Planar Graphs ”

- Adam E. Letchford [Math of OR, 2000]

- Introduces a class of inequalities called “Domino Parity Constraints” which generalize comb inequalities.
- Proves that these constraints can be separated in polynomial time in planar graphs

Outline

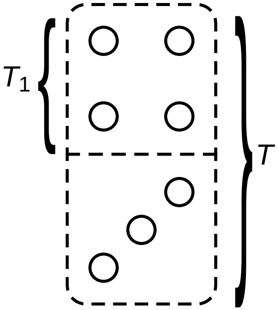
1. The Domino Parity Constraints
   - Introduction
   - Separation
   - Computational Results

2. The k-Parity Constraints
   - Validity
   - Separation
   - Computational Results

Dominoes

Definition (Domino)

A *domino* is a pair of sets $(T_1; T)$ such that $\emptyset \subsetneq T_1 \subsetneq T \subsetneq V$.

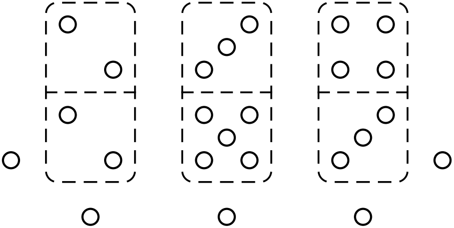


Lemma

For any valid tour $x \in \{0,1\}^E$,

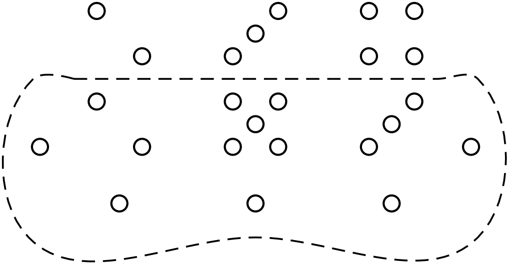
$$x(E(T_1 : T \setminus T_1)) + x(\delta(T)) \geq 3$$

The Domino Parity Constraint

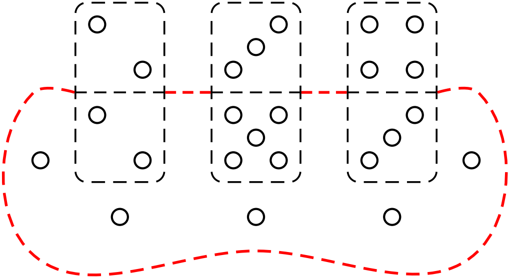


$$\sum_{T \in \mathcal{T}} x(E(T_1 : T \setminus T_1)) + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 3p$$

The Domino Parity Constraint

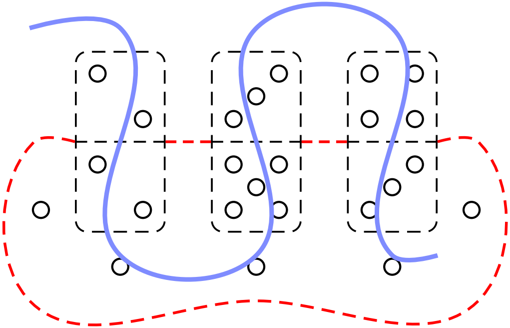


$$\sum_{T \in \mathcal{T}} x(E(T_1 : T \setminus T_1)) + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 3p$$



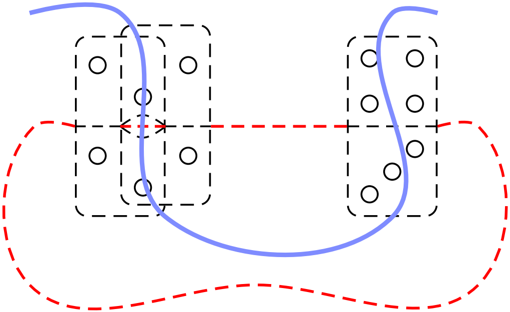
$$x(F_H) + \sum_{T \in \mathcal{T}} x(E(T_1 : T \setminus T_1)) + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 3p + 1$$

The Domino Parity Constraint



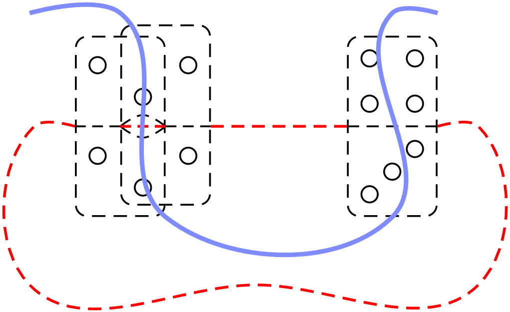
$$x(F_H) + \sum_{T \in \mathcal{T}} x(E(T_1 : T \setminus T_1)) + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 3p + 1$$

The Domino Parity Constraint II



$$x(F_H) + \sum_{T \in \mathcal{T}} x(E(T_1 : T \setminus T_1)) + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 3p + 1$$

The Domino Parity Constraint II



$$\mu^H x + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 3p + 1$$

Preliminaries

Definition (Support Graph)

Let,

$$E^* = \{e \in E : x_e^* > 0\}$$

The *support graph* of G is the sub-graph $G^* = (V, E^*)$.

Henceforth assume:

- $x^* \in \mathbb{Q}_+^E$ satisfi es all subtour elimination constraints.
- G^* is planar, and $\bar{G}^*$ is its dual.
- For $F \subseteq E^*$ let $\bar{F}$ be corresponding edges in $\bar{G}^*$.

Super Connectivity

Definition (Super-Connectivity)

A domino $(T_1; T)$ is *super-connected* if:

- T and $V \setminus T$ are connected in G^*.
- T_1 and $T \setminus T_1$ are connected in G^*.
- $x^*(E(T_1 : V \setminus T)) > 0$ and $x^*(E(T \setminus T_1 : V \setminus T)) > 0$.

Lemma

Every tooth $(T_1; T)$ in a violated domino-parity is super-connected.

Super Connectivity and Duality

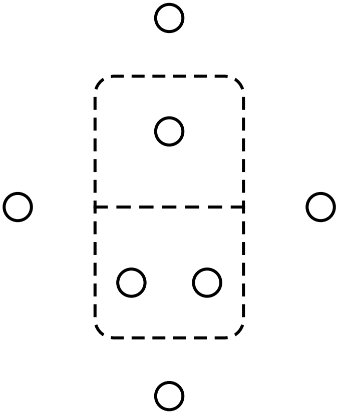
Lemma

If a tooth $(T_1; T)$ is super-connected, then

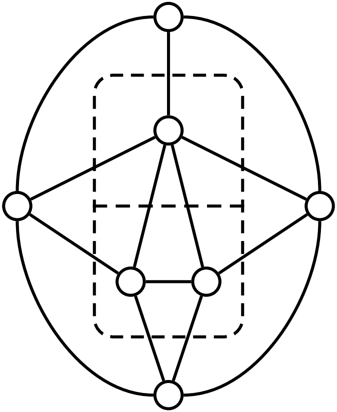
- $C = \overline{\delta(T)}$ *is a simple cycle in* $\bar{G}^*$.
- $P = \overline{E(T_1 : T \setminus T_1)}$ *is a simple path with end-points in* C.

Observation:
There exist two nodes $s, t \in V(\bar{G}^*)$ such that $C \cup P$ defi ne three disjoint paths in $\bar{G}^*$.

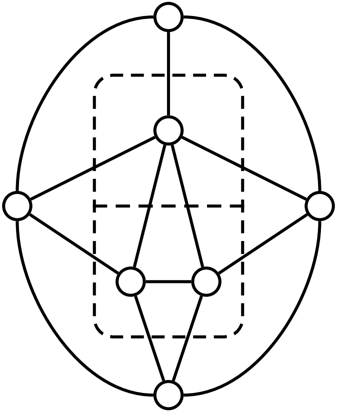
Super Connectivity : An Example

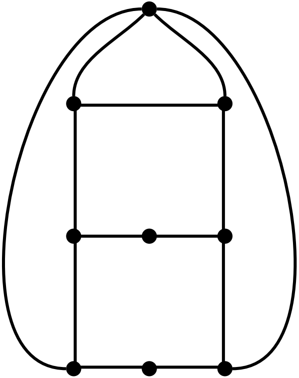


Super Connectivity : An Example

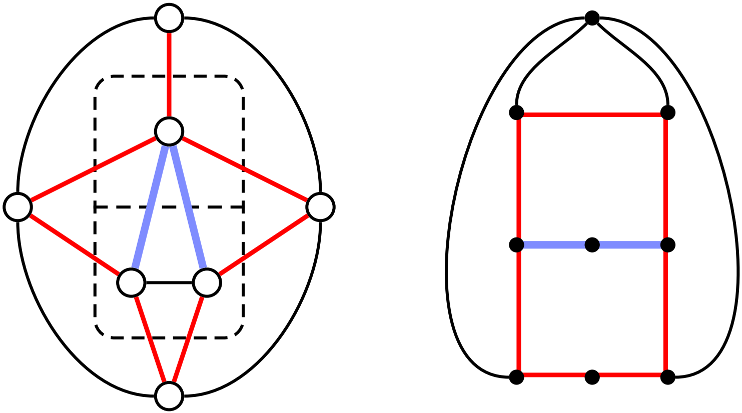


Super Connectivity : An Example

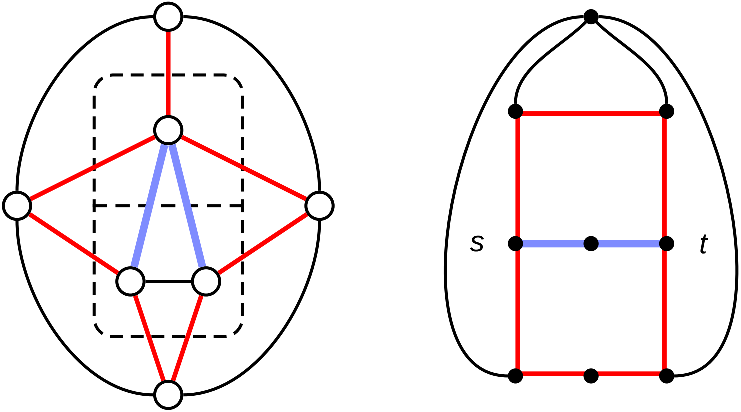




Super Connectivity : An Example



Super Connectivity : An Example



Min Weight Odd Circuit Problem

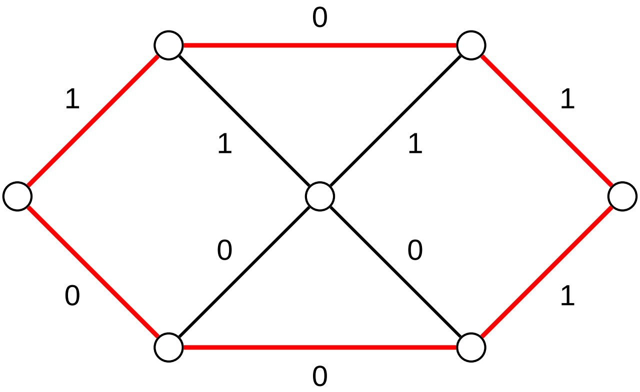
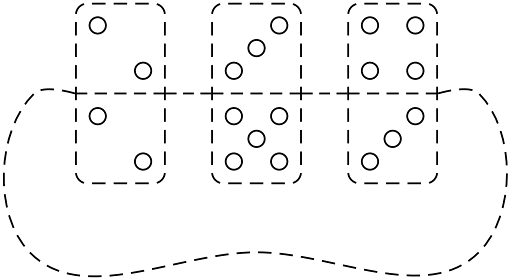


Figure: An Odd Ciruit

Characterization of Domino Parity Constraints

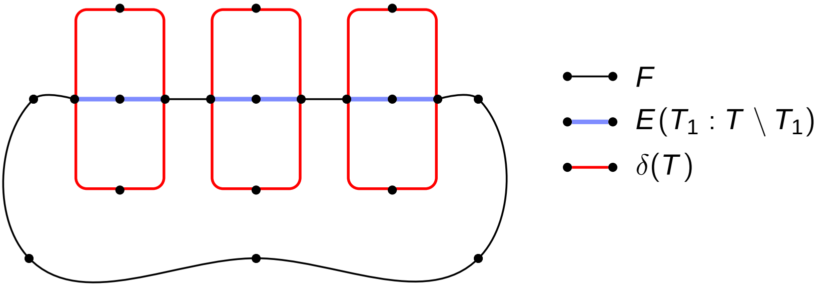
Domino Parity Constraint: Primal Form.



Idea: Reduce cut-generation to min-weight odd circuit problem.

Characterization of Domino Parity Constraints

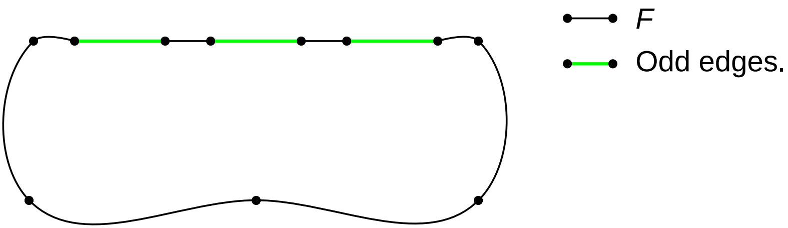
Domino Parity Constraint: Dual Form.



Idea: Reduce cut-generation to min-weight odd circuit problem.

Characterization of Domino Parity Constraints

Domino Parity Constraint: Extended Dual Form.



Black edges + odd edges defi ne an odd circuit.

Implementation: Planarization

If graph is not planar, identify a forbidden minor. Then:

- Contract edges.
  Contracting nodes which are not connected in the minor eliminates it from the shrunk graph.
- Delete edges.
  Careful! Deleting edges means loosing validity of Subtour elimination constraints.

Implementation: Speed-Ups

- Pruning to restrict search space.
  Allows us to build cuts in BB tree, even in largest instances.
- Safe shrinking to reduce problem size.
  70% - 90% smaller graphs obtained.
- Random walk to generate more cuts.
  Thousands of cuts generated per run.
- Tighten to re-utilize old cuts.
  Thousands of cuts generated per round.
  Generates cuts not visible after planarization.

Tests on TSPLIB

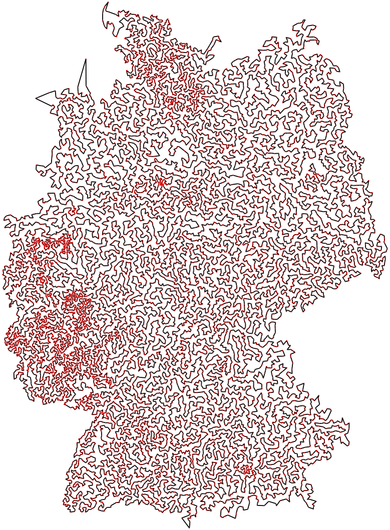
Improvements to Root LP on 8 largest TSPLIB problems:

Name	GAP Closed
pla7397	67%
rl11849	66%
usa13509	55%
brd14051	52%
d15112	47%
d18512	40%
pla33810	38%
pla85900*	27%

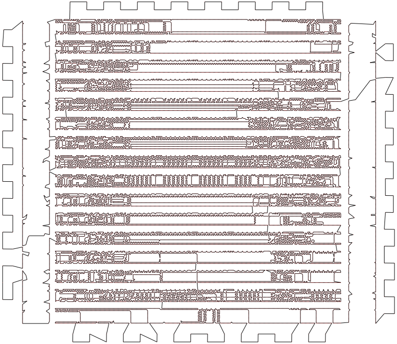
(*) Problem is not optimally solved as of yet.

Previously Unsolved Problems: d18512.

Number of Cities: 18,512.
Origin: Cities of Germany.
Optimal Solution: 645,238.
BB Nodes: 424,241.
Approx. Sol. Time: 57.5 years.



Previously Unsolved Problems: pla33810.



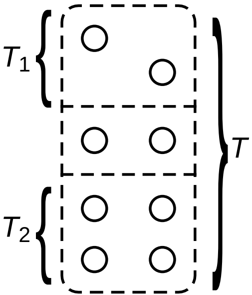
Number of Cities: 33,810.
Origin: Circuit Board Design.
Optimal Solution: 66,048,945.
BB Nodes: 577.
Approx. Sol. Time: 15.7 years.
Largest TSP instance solved!

k-Dominoes

Definition (k-Domino)

A *k-domino* consists of $k+1$ sets $(T_1, T_2, \ldots, T_k; T)$ such that:

- $\emptyset \subsetneq T_i \subsetneq T \subsetneq V$ for all $i = 1, \ldots, k$
- The edges $\bigcup\{E(T_i : T \setminus T_i) : i \in I\}$ define a $|I| + 1$ cut in the subgraph of G induced by T, for all $\emptyset \subsetneq I \subset \{1, \ldots, k\}$.

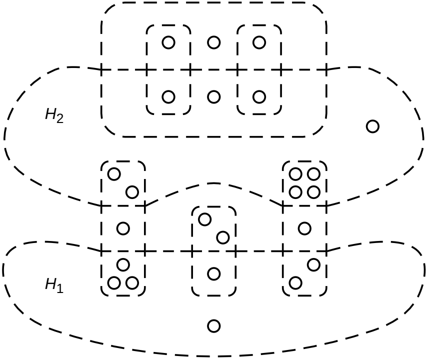


Lemma

For any valid tour $x \in \{0,1\}^E$,

$$\frac{x(\delta(T)) - 2}{2} + \sum_{i=1}^{k} x(E(T_i : T \setminus T_i)) \geq k$$

The k-Parity Constraint



- $\Lambda(T) \subseteq \mathcal{H} \quad \forall T \in \mathcal{T}$
- $\Lambda(H) \subseteq \mathcal{T} \quad \forall H \in \mathcal{H}$
- $|\Lambda(H)|$ odd $\quad \forall H \in \mathcal{H}$
- $T \leftarrow |\Lambda(T)|$-domino $\forall T \in \mathcal{T}$

$$\sum_{H \in \mathcal{H}} \mu^H x + \sum_{T \in \mathcal{T}} x(\delta(T)) \geq 2|\mathcal{T}| + |\mathcal{H}| + \sum_{H \in \mathcal{H}} |\Lambda(H)|$$

Super Connectivity and k-Dominoes

Definition (Super-Connectivity)

A k-domino $(T_1, T_2, \ldots, T_k; T)$ is *super-connected* if:

- T and $V \setminus T$ are connected in G^*.
- T_i and $T \setminus T_i$ are connected in $G^* \qquad \forall i = 1, \ldots, k$.
- $x^*(E(T_i : V \setminus T)) > 0$ and $x^*(E(T \setminus T_i : V \setminus T)) > 0 \quad \forall 0 \leq i \leq k$.

Question

Is every k-domino in a violated constraint super-connected?

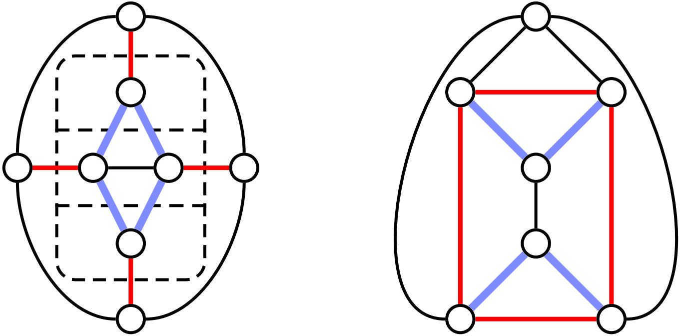
Super Connectivity and Duality

Lemma

If a k-domino $(T_1, T_2, \ldots, T_k; T)$ *is super-connected, then*

- $C = \overline{\delta(T)}$ *is a simple cycle in* $\bar{G}^*$.
- $P_i = \overline{E(T_i : T \setminus T_i)}$ *is a simple path with end-points in* C.
- *All of the paths* P_i *are on the same side of* C *with regard to the planar embedding.*

Super Connectivity : An Example



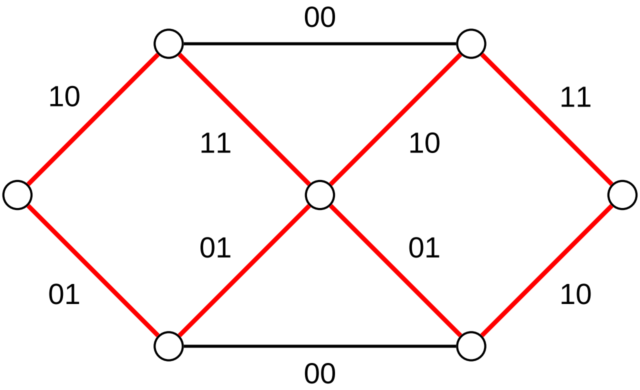
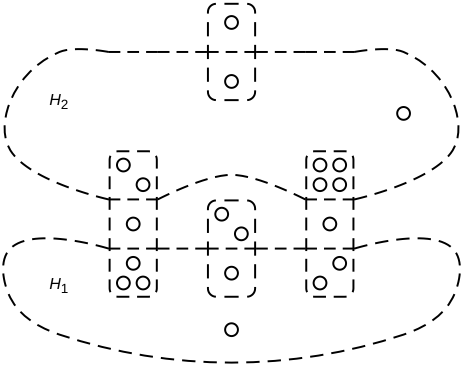


Figure: A 11-Parity Euler Subgraph

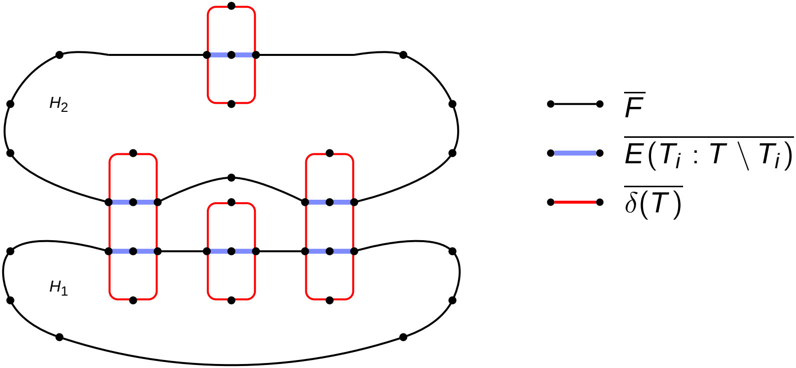
For Clique Trees:
Problem Reduces to Min-Weight-1^k-Parity Euler Subgraph Problem

Characterization of k-Parity Constraints



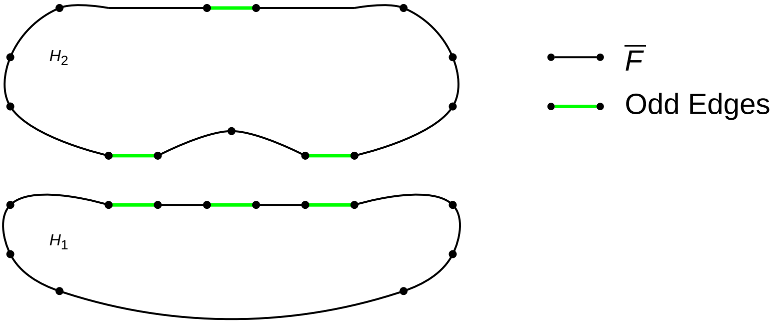
Idea: Each handle can be modelled as an odd circuit.

Characterization of k-Parity Constraints



Here we consider relevant edges in the dual of the support graph.

Characterization of k-Parity Constraints



We observe two odd eulerian graphs: One for each handle.

A Separation Heuristic for $k = 2$.

Heuristic:

- Find a Domino-Parity constraint with slack as small as possible.
- Grow a second handle by solving a constrained odd circuit problem.

Other Implementation Details:

- Planarization, Pruning.
- Shrinking, Tighten.

Tests on TSPLIB

How much do these new cuts close the gap?

Instance	DP	2P
pcb3038	75.38%	14.06%
fnl4461	45.46%	19.05%
rl5914	61.76%	10.58%
rl5934	64.98%	26.32%

Each test instance was ran ten times.
Results are computed from average solutions.

Thank you! Questions?