Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope

Colloquium "Großer Beleg"

Maximilian Moeller

21.11.2023

2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
Colloquium "Großer Beleg"
Maximilian Moeller
21.11.2023

Agenda

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└ Agenda

Agenda
Introduction
Separating Inequalities
Data
Empirical Results
Conclusion

Introduction

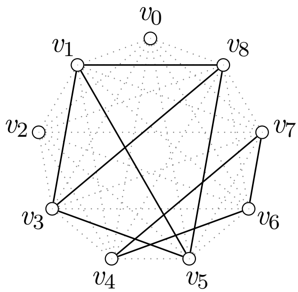
2023-11-21

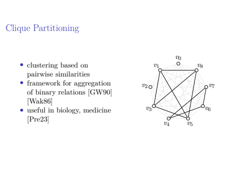
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Introduction

Introduction

Clique Partitioning

- clustering based on pairwise similarities
- framework for aggregation of binary relations [GW90] [Wak86]
- useful in biology, medicine [Pre23]





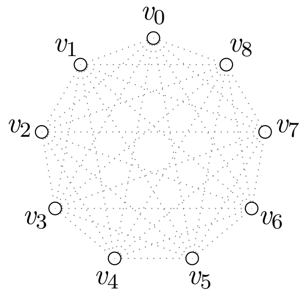
1. biology: taxonomy of animals (*e.g.*, whales and dolphins)
2. medicine: clustering of organoids in light microscopy images
3. probably also operations research

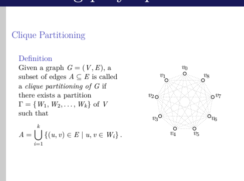
Clique Partitioning

Definition

Given a graph $G = (V, E)$, a subset of edges $A \subseteq E$ is called a *clique partitioning of* G if there exists a partition $\Gamma = \{W_1, W_2, \ldots, W_k\}$ of V such that

$$A = \bigcup_{i=1}^{k} \{(u, v) \in E \mid u, v \in W_i\}.$$





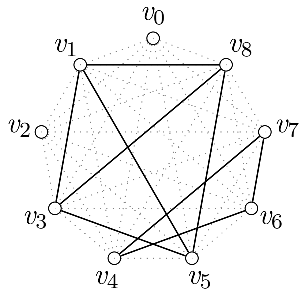
1. show which partition is induced
2. typically only complete graphs considered (indicates for every pair of nodes)
3. then one-to-one corresponds to an equivalence relation
4. induces complete subgraphs *clique* partitioning

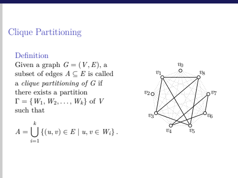
Clique Partitioning

Definition

Given a graph $G = (V, E)$, a subset of edges $A \subseteq E$ is called a *clique partitioning of* G if there exists a partition $\Gamma = \{W_1, W_2, \ldots, W_k\}$ of V such that

$$A = \bigcup_{i=1}^{k} \{(u, v) \in E \mid u, v \in W_i\}.$$




Clique Partitioning
Definition
Given a graph $G = (V, E)$, a subset of edges $A \subseteq E$ is called a *clique partitioning of* G if there exists a partition $\Gamma = \{W_1, W_2, \ldots, W_k\}$ of V such that
$A = \bigcup_{i=1}^{k} \{(u, v) \in E \mid u, v \in W_i\}.$

1. show which partition is induced
2. typically only complete graphs considered (indicates for every pair of nodes)
3. then one-to-one corresponds to an equivalence relation
4. induces complete subgraphs *clique* partitioning

The Clique Partitioning Problem

Input: complete graph $K_n = (V_n, E_n)$ and edge costs $w \in \mathbb{R}^{E_n}$

Compute: a clique partitioning of minimum weight

$$\begin{aligned} \min \quad & \sum_{e \in E_n} w_e x_e \\ \text{s.t.} \quad & \forall e \in E_n \colon x_e \in \{0,1\} \\ & x \text{ characterizes a clique partitioning} \end{aligned}$$

The Clique Partitioning Problem
Input: complete graph $K_n = (V_n, E_n)$ and edge costs $w \in \mathbb{R}^{E_n}$
Compute: a clique partitioning of minimum weight
min $\sum_{e \in E_n} w_e x_e$
s.t. $\forall e \in E_n \colon x_e \in \{0,1\}$
x characterizes a clique partitioning

1. characterize clique partitionings by binary vectors
2. 1 means it is in the clique partitioning and 0 means it is not in it
3. exactly opposite to multicut (equivalent problem)

The Clique Partitioning Problem

- NP-complete [Wak86]

The Clique Partitioning Problem
- NP-complete [Wak86]

1. blue drawn are edges with negative weights, all others non-negative
2. interesting instances need to have ‘conflicts’

The Clique Partitioning Problem

- NP-complete [Wak86]
- Trivial cases:

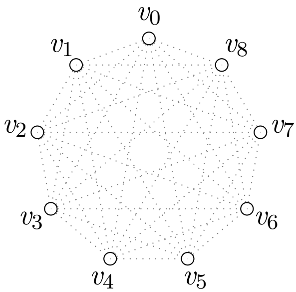
The Clique Partitioning Problem

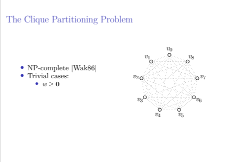
- NP-complete [Wak86]
- Trivial cases:

1. blue drawn are edges with negative weights, all others non-negative
2. interesting instances need to have ‘conflicts’

The Clique Partitioning Problem

- NP-complete [Wak86]
- Trivial cases:
  - $w \geq \mathbf{0}$

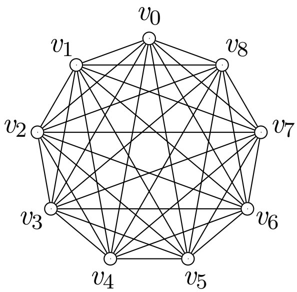


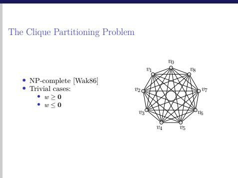


1. blue drawn are edges with negative weights, all others non-negative
2. interesting instances need to have ‘conflicts’

The Clique Partitioning Problem

- NP-complete [Wak86]
- Trivial cases:
  - $w \geq \mathbf{0}$
  - $w \leq \mathbf{0}$

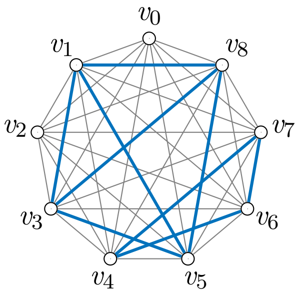


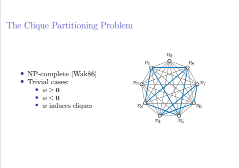


1. blue drawn are edges with negative weights, all others non-negative
2. interesting instances need to have ‘conflicts’

The Clique Partitioning Problem

- NP-complete [Wak86]
- Trivial cases:
  - $w \geq \mathbf{0}$
  - $w \leq \mathbf{0}$
  - w induces cliques





1. blue drawn are edges with negative weights, all others non-negative
2. interesting instances need to have ‘conflicts’

An ILP-formulation of Clique Partitioning

characteristic vectors x

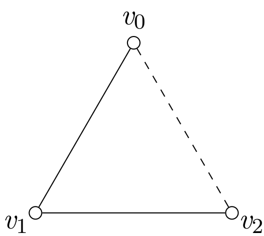
An ILP-formulation of Clique Partitioning
characteristic vectors x

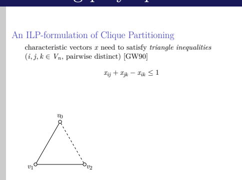
1. we want to make use of (I)LP techniques: linear inequalities
2. explain support graph: left hand side can be drawn as graph, dotted negative, solid positive
3. triangle inequalities ensure transitivity
4. for the same i, j, k there are actually three triangles (show)

An ILP-formulation of Clique Partitioning

characteristic vectors x need to satisfy *triangle inequalities* ($i, j, k \in V_n$, pairwise distinct) [GW90]

$$x_{ij} + x_{jk} - x_{ik} \leq 1$$



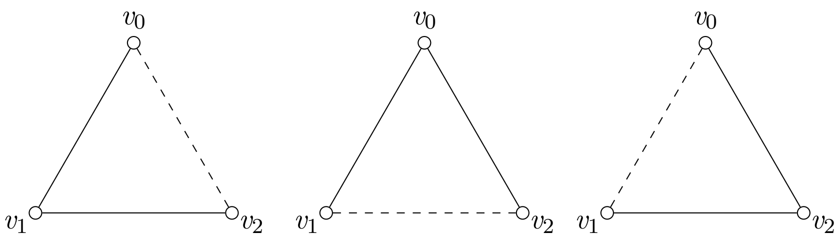

An ILP-formulation of Clique Partitioning
characteristic vectors x need to satisfy *triangle inequalities* ($i, j, k \in V_n$, pairwise distinct) [GW90]
$$x_{ij} + x_{jk} - x_{ik} \leq 1$$

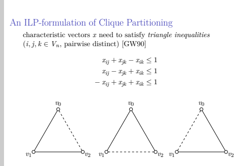
1. we want to make use of (I)LP techniques: linear inequalities
2. explain support graph: left hand side can be drawn as graph, dotted negative, solid positive
3. triangle inequalities ensure transitivity
4. for the same i, j, k there are actually three triangles (show)

An ILP-formulation of Clique Partitioning

characteristic vectors x need to satisfy *triangle inequalities* ($i, j, k \in V_n$, pairwise distinct) [GW90]

$$\begin{aligned} x_{ij} + x_{jk} - x_{ik} &\leq 1 \\ x_{ij} - x_{jk} + x_{ik} &\leq 1 \\ -x_{ij} + x_{jk} + x_{ik} &\leq 1 \end{aligned}$$





1. we want to make use of (I)LP techniques: linear inequalities
2. explain support graph: left hand side can be drawn as graph, dotted negative, solid positive
3. triangle inequalities ensure transitivity
4. for the same i, j, k there are actually three triangles (show)

The Task

The Task

1. original task: branch-and-cut algorithm, *i.e.*, solving the ILP
2. consists of multiple parts: **cutting planes for bounds**, finding good solutions and branch management
3. explain overall iteration procedure
4. even if not integral, a few percent off might be good for practical use
5. however, no guarantee to arrive at integral or within a constant factor

The Task

focus on cutting plane procedure:

- solve LP relaxation of the problem

The Task

focus on cutting plane procedure:
- solve LP relaxation of the problem

1. original task: branch-and-cut algorithm, *i.e.*, solving the ILP
2. consists of multiple parts: **cutting planes for bounds**, finding good solutions and branch management
3. explain overall iteration procedure
4. even if not integral, a few percent off might be good for practical use
5. however, no guarantee to arrive at integral or within a constant factor

The Task

focus on cutting plane procedure:

- solve LP relaxation of the problem
- tighten the solution iteratively

The Task

focus on cutting plane procedure:
- solve LP relaxation of the problem
- tighten the solution iteratively

1. original task: branch-and-cut algorithm, *i.e.*, solving the ILP
2. consists of multiple parts: **cutting planes for bounds**, finding good solutions and branch management
3. explain overall iteration procedure
4. even if not integral, a few percent off might be good for practical use
5. however, no guarantee to arrive at integral or within a constant factor

The Task

focus on cutting plane procedure:

- solve LP relaxation of the problem
- tighten the solution iteratively
- hopefully arrive at integral solution

The Task
focus on cutting plane procedure:
- solve LP relaxation of the problem
- tighten the solution iteratively
- hopefully arrive at integral solution

1. original task: branch-and-cut algorithm, *i.e.*, solving the ILP
2. consists of multiple parts: **cutting planes for bounds**, finding good solutions and branch management
3. explain overall iteration procedure
4. even if not integral, a few percent off might be good for practical use
5. however, no guarantee to arrive at integral or within a constant factor

The Task

focus on cutting plane procedure:

- solve LP relaxation of the problem
- tighten the solution iteratively
- hopefully arrive at integral solution
- ???
- profit

The Task

focus on cutting plane procedure:
- solve LP relaxation of the problem
- tighten the solution iteratively
- hopefully arrive at integral solution
- ???
- profit

1. original task: branch-and-cut algorithm, *i.e.*, solving the ILP
2. consists of multiple parts: **cutting planes for bounds**, finding good solutions and branch management
3. explain overall iteration procedure
4. even if not integral, a few percent off might be good for practical use
5. however, no guarantee to arrive at integral or within a constant factor

Separating Inequalities

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities

Separating Inequalities

Ingredients

- classes of facet-defining linear inequalities

Ingredients

- classes of facet-defining linear inequalities

1. explain: what is separation (check (and produce))
2. explain: facet-defining

Ingredients

- classes of facet-defining linear inequalities
- algorithms for separating them

Ingredients

- classes of facet-defining linear inequalities
- algorithms for separating them

1. explain: what is separation (check (and produce))
2. explain: facet-defining

Ingredients

- classes of facet-defining linear inequalities
- algorithms for separating them
- data to test on (next section)

Ingredients

- classes of facet-defining linear inequalities
- algorithms for separating them
- data to test on (next section)

1. explain: what is separation (check (and produce))
2. explain: facet-defining

Triangle Inequalities (Δ)

Using complete enumeration $\mathcal{O}(n^3)$

Four variations

Triangle Inequalities (Δ)
Using complete enumeration $\mathcal{O}(n^3)$
Four variations

1. Separation algorithms: separators
2. separators get abbreviations
3. `maxcut` is for limiting lp size
4. `var_once` is for solving the graph evenly

Triangle Inequalities (Δ)

Using complete enumeration $\mathcal{O}(n^3)$

Four variations

abbreviation	`maxcut`	`var_once`
Δ	400	–
Δ_∞	∞	–
$\Delta^{\leq 1}$	400	✓
$\Delta^{\leq 1}_\infty$	∞	✓

Triangle Inequalities (Δ)
Using complete enumeration $\mathcal{O}(n^3)$
Four variations

abbreviation	`maxcut`	`var_once`
Δ	400	–
Δ_∞	∞	–
$\Delta^{\leq 1}$	400	✓
$\Delta^{\leq 1}_\infty$	∞	✓

1. Separation algorithms: separators
2. separators get abbreviations
3. `maxcut` is for limiting lp size
4. `var_once` is for solving the graph evenly

2-Partition Inequalities

Let $S, T \subseteq V_n$ disjoint.

2-*partition inequality* ([S, T]-inequality) [GW90]

$$\sum_{s \in S} \sum_{t \in T} x_{st} - \sum_{\substack{s,s' \in S \\ s \neq s'}} x_{ss'} - \sum_{\substack{t,t' \in T \\ t \neq t'}} x_{tt'} \leq \min(|S|, |T|)$$

2-Partition Inequalities
Let $S, T \subseteq V_n$ disjoint.
2-*partition inequality* ([S, T]-inequality) [GW90]
$$\sum_{s \in S} \sum_{t \in T} x_{st} - \sum_{\substack{s,s' \in S \\ s \neq s'}} x_{ss'} - \sum_{\substack{t,t' \in T \\ t \neq t'}} x_{tt'} \leq \min(|S|, |T|)$$

1. Generalization of triangle inequalities

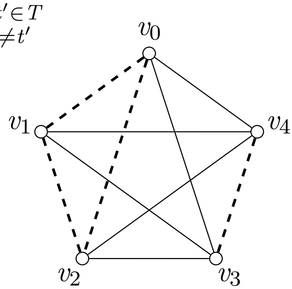
2-Partition Inequalities

Let $S, T \subseteq V_n$ disjoint.

2-*partition inequality* ([S, T]-inequality) [GW90]

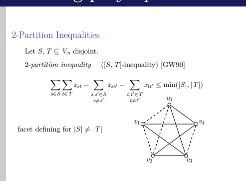
$$\sum_{s\in S}\sum_{t\in T} x_{st} - \sum_{\substack{s,s'\in S\\ s\neq s'}} x_{ss'} - \sum_{\substack{t,t'\in T\\ t\neq t'}} x_{tt'} \leq \min(|S|, |T|)$$

facet defining for $|S| \neq |T|$



2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─2-Partition Inequalities



1. Generalization of triangle inequalities

Separation of 2-Partition Inequalities (st)

- already NP-hard for any fixed $|S|$ [ORS01]

Separation of 2-Partition Inequalities (st)

- already NP-hard for any fixed $|S|$ [ORS01]

1. no use in using approximation if it is np-hard again
2. not explaining details, involve random choosing of an order of some nodes. Thus re-run three times.

Separation of 2-Partition Inequalities (st)

- already NP-hard for any fixed $|S|$ [ORS01]
- using two heuristics: $\texttt{st}^1$ and $\texttt{st}^2$ from [GW89]

Separation of 2-Partition Inequalities (st)

- already NP-hard for any fixed $|S|$ [ORS01]
- using two heuristics: $\texttt{st}^1$ and $\texttt{st}^2$ from [GW89]

1. no use in using approximation if it is np-hard again
2. not explaining details, involve random choosing of an order of some nodes. Thus re-run three times.

Separation of 2-Partition Inequalities (st)

- already NP-hard for any fixed $|S|$ [ORS01]
- using two heuristics: $\texttt{st}^1$ and $\texttt{st}^2$ from [GW89]
- only search for constraints with $|S| = 1$

Separation of 2-Partition Inequalities (st)
- already NP-hard for any fixed $|S|$ [ORS01]
- using two heuristics: $\texttt{st}^1$ and $\texttt{st}^2$ from [GW89]
- only search for constraints with $|S| = 1$

1. no use in using approximation if it is np-hard again
2. not explaining details, involve random choosing of an order of some nodes. Thus re-run three times.

Two-Chorded Odd Cycle Ineuqalities

Let $5 \leq k \leq n$ and $v : \mathbb{Z}_k \to \mathbb{Z}_n$ injective

two-chorded odd cycle inequality [GW90]

$$\sum_{i \in \mathbb{Z}_k} x_{v_i v_{i+1}} - x_{v_i v_{i+2}} \leq \left\lfloor \frac{1}{2} k \right\rfloor$$

Two-Chorded Odd Cycle Ineuqalities
Let $5 \leq k \leq n$ and $v : \mathbb{Z}_k \to \mathbb{Z}_n$ injective
two-chorded odd cycle inequality [GW90]
$$\sum_{i \in \mathbb{Z}_k} x_{v_i v_{i+1}} - x_{v_i v_{i+2}} \leq \left\lfloor \frac{1}{2} k \right\rfloor$$

1. separable in polytime shown by müller 1996
2. technique hard to motivate

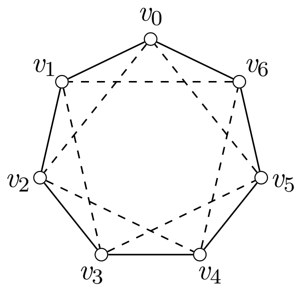
Two-Chorded Odd Cycle Ineuqalities

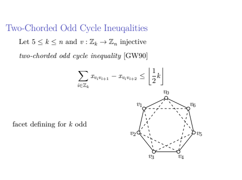
Let $5 \leq k \leq n$ and $v : \mathbb{Z}_k \to \mathbb{Z}_n$ injective

two-chorded odd cycle inequality [GW90]

$$\sum_{i \in \mathbb{Z}_k} x_{v_i v_{i+1}} - x_{v_i v_{i+2}} \leq \left\lfloor \frac{1}{2} k \right\rfloor$$

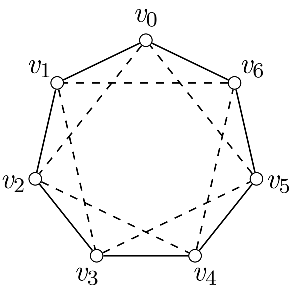
facet defining for k odd





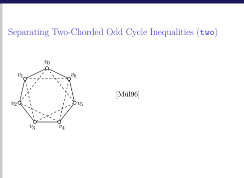
1. separable in polytime shown by müller 1996
2. technique hard to motivate

Separating Two-Chorded Odd Cycle Inequalities (`two`)



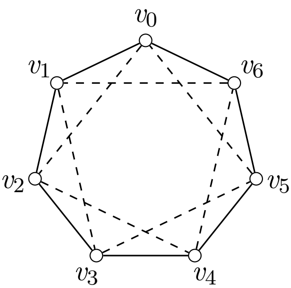
[Mül96]

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─Separating Two-Chorded Odd Cycle Inequalities (two)



1. following an odd cycle and summing edges by +- 1 gives left hand side
2. we do not need to know the k to know that we get a lhs
3. maybe we can also rewrite the rhs independent of k
4. new problem distinguish even from odd cycles -> hints at auxiliary graph

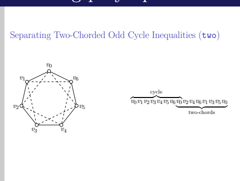
Separating Two-Chorded Odd Cycle Inequalities (`two`)



$$\overbrace{v_0\, v_1\, v_2\, v_3\, v_4\, v_5\, v_6\, v_0}^{\text{cycle}}\underbrace{v_2\, v_4\, v_6\, v_1\, v_3\, v_5\, v_0}_{\text{two-chords}}$$

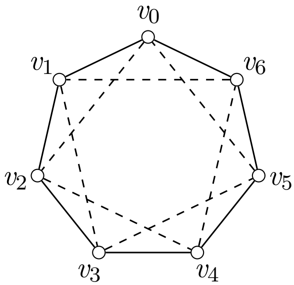
2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─Separating Two-Chorded Odd Cycle Inequalities (two)



1. following an odd cycle and summing edges by +- 1 gives left hand side
2. we do not need to know the k to know that we get a lhs
3. maybe we can also rewrite the rhs independent of k
4. new problem distinguish even from odd cycles -> hints at auxiliary graph

Separating Two-Chorded Odd Cycle Inequalities (`two`)



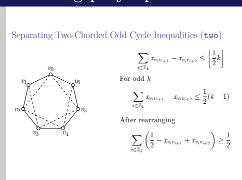
$$\sum_{i\in\mathbb{Z}_k} x_{v_iv_{i+1}} - x_{v_iv_{i+2}} \leq \left\lfloor \frac{1}{2}k \right\rfloor$$

For odd k

$$\sum_{i\in\mathbb{Z}_k} x_{v_iv_{i+1}} - x_{v_iv_{i+2}} \leq \frac{1}{2}(k-1)$$

After rearranging

$$\sum_{i\in\mathbb{Z}_k} \left(\frac{1}{2} - x_{v_iv_{i+1}} + x_{v_iv_{i+2}}\right) \geq \frac{1}{2}$$



1. following an odd cycle and summing edges by +- 1 gives left hand side
2. we do not need to know the k to know that we get a lhs
3. maybe we can also rewrite the rhs independent of k
4. new problem distinguish even from odd cycles -> hints at auxiliary graph

Auxiliary Graph $H = (V_H, E_H)$

For every pair $(i, j) \in V_n^2$ with
$i \neq j$:

2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─Auxiliary Graph $H = (V_H, E_H)$

Auxiliary Graph $H = (V_H, E_H)$

For every pair $(i, j) \in V_n^2$ with
$i \neq j$:

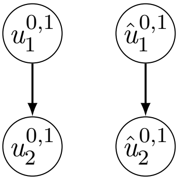
1. bipartide graph can distinguish between odd and even walks
2.

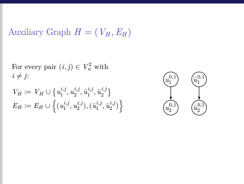
Auxiliary Graph $H = (V_H, E_H)$

For every pair $(i, j) \in V_n^2$ with $i \neq j$:

$$V_H := V_H \cup \left\{ u_1^{i,j}, u_2^{i,j}, \hat{u}_1^{i,j}, \hat{u}_2^{i,j} \right\}$$

$$E_H := E_H \cup \left\{ (u_1^{i,j}, u_2^{i,j}), (\hat{u}_1^{i,j}, \hat{u}_2^{i,j}) \right\}$$





1. bipartide graph can distinguish between odd and even walks
2.

Auxiliary Graph $H = (V_H, E_H)$

For every triple $(i, j, k) \in V_n^3$
with i, j, k pairwise distinct:

Auxiliary Graph $H = (V_H, E_H)$

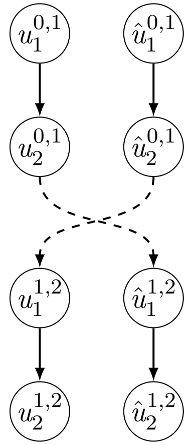
For every triple $(i, j, k) \in V_n^3$
with i, j, k pairwise distinct:

1. bipartide graph can distinguish between odd and even walks
2.

Auxiliary Graph $H = (V_H, E_H)$

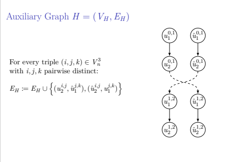
For every triple $(i, j, k) \in V_n^3$
with i, j, k pairwise distinct:

$$E_H := E_H \cup \left\{ (u_2^{i,j}, \hat{u}_1^{j,k}), (\hat{u}_2^{i,j}, u_1^{j,k}) \right\}$$



2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─Auxiliary Graph $H = (V_H, E_H)$



1. bipartide graph can distinguish between odd and even walks
2.

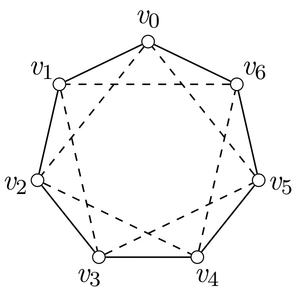
Weights for the auxiliary graph

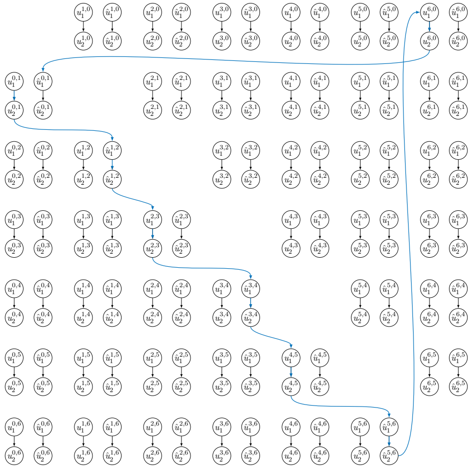
$$\sum_{i \in \mathbb{Z}_k} \Bigg(\underbrace{\frac{1}{2} - x_{v_i v_{i+1}}}_{\text{inner-gadget weights}} + \underbrace{x_{v_i v_{i+2}}}_{\text{inter-gadget weights}} \Bigg) \geq \frac{1}{2}$$

Weights for the auxiliary graph

$$\sum_{i \in \mathbb{Z}_k} \Bigg(\underbrace{\frac{1}{2} - x_{v_i v_{i+1}}}_{\text{inner-gadget weights}} + \underbrace{x_{v_i v_{i+2}}}_{\text{inter-gadget weights}} \Bigg) \geq \frac{1}{2}$$

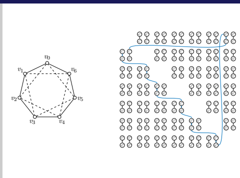
1. no negative cycles as long as triangle inequalities are satisfied! (müller)





2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities



1. each violated odd-cycle inequality corresponds to a u,uhat walk (same gadget)

Dimensions of H

$$|V_H| = 4n(n-1) = 4n^2 - 4n$$

$$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$$

Dimensions of H

$|V_H| = 4n(n-1) = 4n^2 - 4n$
$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$

1. size has effect on choice of shortest path
2. floyd-warshall takes 80GB of memory on largest instances
3. we can terminate belman-ford early -> after 15 minutes

Dimensions of H

$$|V_H| = 4n(n-1) = 4n^2 - 4n$$
$$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$$

Shortest path-algorithms

- Floyd-Warshall $\mathcal{O}(|V|^3) = \mathcal{O}(n^6)$

Dimensions of H
$|V_H| = 4n(n-1) = 4n^2 - 4n$
$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$
Shortest path-algorithms
- Floyd-Warshall $\mathcal{O}(|V|^3) = \mathcal{O}(n^6)$

1. size has effect on choice of shortest path
2. floyd-warshall takes 80GB of memory on largest instances
3. we can terminate belman-ford early -> after 15 minutes

Dimensions of H

$$|V_H| = 4n(n-1) = 4n^2 - 4n$$
$$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$$

Shortest path-algorithms

- Floyd-Warshall $\mathcal{O}(|V|^3) = \mathcal{O}(n^6)$
- Belman-Ford: $\mathcal{O}(|V_H| \cdot |E_H|) = \mathcal{O}(n^5)$, $\mathcal{O}(n^2)$ many times

Dimensions of H

$|V_H| = 4n(n-1) = 4n^2 - 4n$
$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$

Shortest path-algorithms
- Floyd-Warshall $\mathcal{O}(|V|^3) = \mathcal{O}(n^6)$
- Belman-Ford: $\mathcal{O}(|V_H| \cdot |E_H|) = \mathcal{O}(n^5)$, $\mathcal{O}(n^2)$ many times

1. size has effect on choice of shortest path
2. floyd-warshall takes 80GB of memory on largest instances
3. we can terminate belman-ford early -> after 15 minutes

Dimensions of H

$$|V_H| = 4n(n-1) = 4n^2 - 4n$$

$$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$$

Shortest path-algorithms

- Floyd-Warshall $\mathcal{O}(|V|^3) = \mathcal{O}(n^6)$, $\mathcal{O}(|V|^2) = \mathcal{O}(n^4)$ space!
- Belman-Ford: $\mathcal{O}(|V_H| \cdot |E_H|) = \mathcal{O}(n^5)$, $\mathcal{O}(n^2)$ many times, $\mathcal{O}(|V|) = \mathcal{O}(n^2)$ space

Dimensions of H

$|V_H| = 4n(n-1) = 4n^2 - 4n$
$|E_H| = \underbrace{2n(n-1)}_{\text{inner-gadget edges}} + \underbrace{2n(n-1)(n-2)}_{\text{inter-gadget edges}} = 2n^3 - 4n^2 + 2n$

Shortest path-algorithms
- Floyd-Warshall $\mathcal{O}(|V|^3) = \mathcal{O}(n^6)$, $\mathcal{O}(|V|^2) = \mathcal{O}(n^4)$ space!
- Belman-Ford: $\mathcal{O}(|V_H| \cdot |E_H|) = \mathcal{O}(n^5)$, $\mathcal{O}(n^2)$ many times, $\mathcal{O}(|V|) = \mathcal{O}(n^2)$ space

1. size has effect on choice of shortest path
2. floyd-warshall takes 80GB of memory on largest instances
3. we can terminate belman-ford early -> after 15 minutes

Half-Chorded Odd Cycle Ineuqalities

Let $5 \leq k \leq n$, let $d = \frac{k-1}{2}$ and $v : \mathbb{Z}_k \to \mathbb{Z}_n$ injective

half-chorded odd cycle inequality [And+22]

$$\sum_{i \in \mathbb{Z}_k} (x_{v_i v_{i+1}} - x_{v_i v_{i+d}}) \leq k - 3$$

Half-Chorded Odd Cycle Ineuqalities
Let $5 \leq k \leq n$, let $d = \frac{k-1}{2}$ and $v : \mathbb{Z}_k \to \mathbb{Z}_n$ injective
half-chorded odd cycle inequality [And+22]
$$\sum_{i \in \mathbb{Z}_k} (x_{v_i v_{i+1}} - x_{v_i v_{i+d}}) \leq k - 3$$

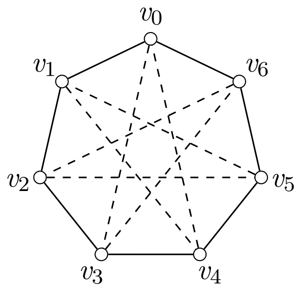
Half-Chorded Odd Cycle Ineuqalities

Let $5 \leq k \leq n$, let $d = \frac{k-1}{2}$ and $v : \mathbb{Z}_k \to \mathbb{Z}_n$ injective

half-chorded odd cycle inequality [And+22]

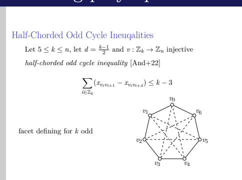
$$\sum_{i \in \mathbb{Z}_k} (x_{v_i v_{i+1}} - x_{v_i v_{i+d}}) \leq k - 3$$

facet defining for k odd

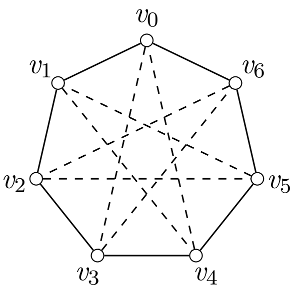


2023-11-21

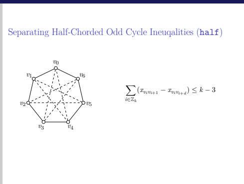
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─Half-Chorded Odd Cycle Ineuqalities



Separating Half-Chorded Odd Cycle Ineuqalities (**half**)

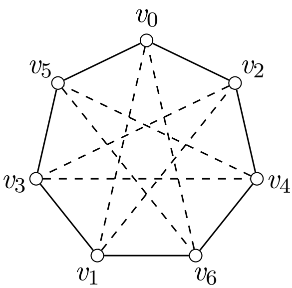


$$\sum_{i \in \mathbb{Z}_k} (x_{v_i v_{i+1}} - x_{v_i v_{i+d}}) \leq k - 3$$



1. half-chords also form a cycle, solid edges are two-chords to that cycle -> relabeling
2. technique with auxiliary graph is exactly the same, except all weights are non-negative
3. dijkstra, can terminate early if current weight is already ≥ 3

Separating Half-Chorded Odd Cycle Ineuqalities (`half`)



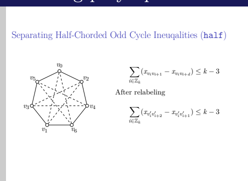
$$\sum_{i \in \mathbb{Z}_k} (x_{v_i v_{i+1}} - x_{v_i v_{i+d}}) \leq k - 3$$

After relabeling

$$\sum_{i \in \mathbb{Z}_k} (x_{v'_i v'_{i+2}} - x_{v'_i v'_{i+1}}) \leq k - 3$$

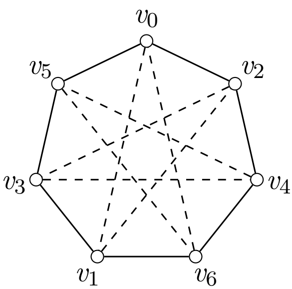
2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Separating Inequalities
└─Separating Half-Chorded Odd Cycle Ineqalities (half)



1. half-chords also form a cycle, solid edges are two-chords to that cycle -> relabeling
2. technique with auxiliary graph is exactly the same, except all weights are non-negative
3. dijkstra, can terminate early if current weight is already ≥ 3

Separating Half-Chorded Odd Cycle Ineuqalities (**half**)

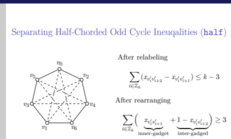


After relabeling

$$\sum_{i\in\mathbb{Z}_k} (x_{v'_i v'_{i+2}} - x_{v'_i v'_{i+1}}) \leq k-3$$

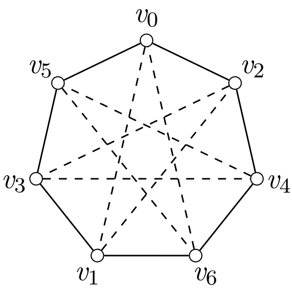
After rearranging

$$\sum_{i\in\mathbb{Z}_k} \Bigg(\underbrace{x_{v'_i v'_{i+1}}}_{\text{inner-gadget}} + \underbrace{1 - x_{v'_i v'_{i+2}}}_{\text{inter-gadged}} \Bigg) \geq 3$$



1. half-chords also form a cycle, solid edges are two-chords to that cycle -> relabeling
2. technique with auxiliary graph is exactly the same, except all weights are non-negative
3. dijkstra, can terminate early if current weight is already ≥ 3

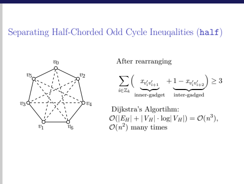
Separating Half-Chorded Odd Cycle Ineuqalities (`half`)



After rearranging

$$\sum_{i \in \mathbb{Z}_k} \Big(\underbrace{x_{v_i' v_{i+1}'}}_{\text{inner-gadget}} + \underbrace{1 - x_{v_i' v_{i+2}'}}_{\text{inter-gadged}} \Big) \geq 3$$

Dijkstra's Algortihm:
$\mathcal{O}(|E_H| + |V_H| \cdot \log |V_H|) = \mathcal{O}(n^3)$,
$\mathcal{O}(n^2)$ many times



1. half-chords also form a cycle, solid edges are two-chords to that cycle -> relabeling
2. technique with auxiliary graph is exactly the same, except all weights are non-negative
3. dijkstra, can terminate early if current weight is already ≥ 3

Run Configurations

	Δ-st-1	Δ-st-2	Δ-st-12	Δ-½	Δ-2	Δ-c	all
1	Δ	Δ	Δ	Δ	Δ	Δ	Δ
2	st[1]	st[2]	st[1]	half	two	half	st[1]
3	—	—	st[2]	—	—	two	st[2]
4	—	—	—	—	—	—	half
5	—	—	—	—	—	—	two

2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└Separating Inequalities
└Run Configurations

Run Configurations

	Δ-st-1	Δ-st-2	Δ-st-12	Δ-½	Δ-2	Δ-c	all
1	Δ	Δ	Δ	Δ	Δ	Δ	Δ
2	st[1]	st[2]	st[1]	half	two	half	st[1]
3	—	—	st[2]	—	—	two	st[2]
4	—	—	—	—	—	—	half
5	—	—	—	—	—	—	two

1. all use Δ first, because they implicitly rely on it
2. ordered them by (expected) run time

Data

2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Data

Data

Natural instances

- `cetacea` (30) [GW89]
- `cats` (36)

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└Data
└Natural instances

2023-11-21

Natural instances

- cetacea (30) [GW89]
- cats (36)

Natural instances

- `cetacea` (30) [GW89]
- `cats` (36)
- `football` (115) [Kap+15]
- `adjnoun` (112)
- `polbooks` (105)
- `lesmis` (77)
- `dolphins` (62)
- `karate` (34)

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Data
└─Natural instances

Natural instances
- cetacea (30) [GW89]
- cats (36)
- football (115) [Kap+15]
- adjnoun (112)
- polbooks (105)
- lesmis (77)
- dolphins (62)
- karate (34)

Natural instances

- `cetacea` (30) [GW89]
- `cats` (36)
- `football` (115) [Kap+15]
- `adjnoun` (112)
- `polbooks` (105)
- `lesmis` (77)
- `dolphins` (62)
- `karate` (34)
- `organoid_[size]_[difficulty]` (160, 100, 80, 40) [Pre23]

Natural instances
- cetacea (30) [GW89]
- cats (36)
- football (115) [Kap+15]
- adjnoun (112)
- polbooks (105)
- lesmis (77)
- dolphins (62)
- karate (34)
- organoid_[size]_[difficulty] (160, 100, 80, 40) [Pre23]

Random instances

- Binary `r_binary_[size]`
  `size` $\in \{25, 50\}$

Random instances
- Binary r_binary_[size]
size $\in \{25, 50\}$

Random instances

- Binary `r_binary_[size]`
  $\texttt{size} \in \{25, 50\}$
- Uniform `r_uniform_[size]_[lb]_[ub]`
  $\texttt{size} \in \{25, 50\}$
  $(\texttt{lb}, \texttt{ub}) \in \{(-10, 10), (-100, 100), (-10, 100)\}$

Random instances

- Binary `r_binary_[size]`
  $\texttt{size} \in \{25, 50\}$
- Uniform `r_uniform_[size]_[lb]_[ub]`
  $\texttt{size} \in \{25, 50\}$
  $(\texttt{lb}, \texttt{ub}) \in \{(-10, 10), (-100, 100), (-10, 100)\}$

Random instances

- Binary `r_binary_[size]`
  $\texttt{size} \in \{25, 50\}$
- Uniform `r_uniform_[size]_[lb]_[ub]`
  $\texttt{size} \in \{25, 50\}$
  $(\texttt{lb}, \texttt{ub}) \in \{(-10, 10), (-100, 100), (-10, 100)\}$
- Normal `r_normal_[size]_[mu]_[sigma]`
  $\texttt{size} \in \{25, 50\}$
  $(\texttt{mu}, \texttt{sigma}) \in \{0, 0.5, 2\} \times \{0.5, 1, 2\}$

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└Data
└Random instances

Random instances

- Binary `r_binary_[size]`
  $\texttt{size} \in \{25, 50\}$
- Uniform `r_uniform_[size]_[lb]_[ub]`
  $\texttt{size} \in \{25, 50\}$
  $(\texttt{lb}, \texttt{ub}) \in \{(-10, 10), (-100, 100), (-10, 100)\}$
- Normal `r_normal_[size]_[mu]_[sigma]`
  $\texttt{size} \in \{25, 50\}$
  $(\texttt{mu}, \texttt{sigma}) \in \{0, 0.5, 2\} \times \{0.5, 1, 2\}$

Empirical Results

2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Empirical Results

Empirical Results

Relative Gap

Relative gap of solution z' to the optimal solution z

$$\mathrm{gap}(z, z') := \frac{|z - z'|}{|z|}$$

Relative Gap

Relative gap of solution z' to the optimal solution z

$$\mathrm{gap}(z, z') := \frac{|z - z'|}{|z|}$$

1. case $z = 0$ can be ignored
2. only possible to compute when optimal value is known

Example: `adjnoun`

Table 6: Computational results for `adjnoun`

						Δ-st-1			Δ-st-2			Δ-st-12						all		
1		Δ	Δ_∞	$\Delta^{\leq 1}$	$\Delta^{\leq 1}_\infty$	1	2	3	1	2	3	1	2	3	Δ-½	Δ-2	Δ-c	1	2	3
2	# it.	122	**6**	204	45	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	1,038	1,320	1,200	10,000	10,000	10,000
3	# $\not\Delta$-calls	0	0	0	0	1067	1094	1095	622	613	614	1111	1098	1095	81	97	110	1117	1131	1083
4	cuts																			
5	# total	47,262	12,963	26,027	**8,003**	390,825	392,629	387,427	994,177	994,987	987,904	382,283	391,531	387,849	140,769	171,755	158,097	373,139	372,381	391,373
6	# max	400	8,362	186	500	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400
7	# min	2	16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	# $\not\Delta$-cuts	0	0	0	0	22,493	22,443	23,675	136,047	134,127	134,175	22,847	23,823	22,871	30,158	38,660	39,872	23,366	22,366	23,421
9	# removed	43,535	7,190	23,452	5,379	386,392	388,288	383,042	988,377	988,865	982,200	377,858	387,156	383,454	134,093	165,476	150,739	368,669	367,987	386,963
10	objective																			
11	bound	0.4275	0.4275	0.4275	0.4275	0.391	0.3904	0.3895	0.3865	**0.3863**	0.3866	0.3899	0.389	0.3903	0.398	0.3978	0.3978	0.3891	0.39	0.3893
12	gap	36.1%	36.1%	36.1%	36.1%	24.4%	24.3%	24.0%	23.0%	22.9%	23.0%	24.1%	23.8%	24.2%	26.7%	26.6%	26.6%	23.8%	24.1%	23.9%
13	time																			
14	total 1/s	4.846	**0.794**	5.822	4.210	652.918	657.831	640.986	1234.069	1249.390	1211.961	654.060	668.754	666.687	2322.078	3922.257	4303.249	650.074	623.255	657.059
15	normalized	1.00	**0.16**	1.20	0.87	134.73	135.75	132.27	254.66	257.82	250.10	134.97	138.00	137.57	479.17	809.38	888.00	134.15	128.61	135.59
16	lp time 1/s	4.701	**0.764**	5.566	4.042	618.769	623.890	606.817	1198.666	1213.964	1176.594	619.855	634.372	632.492	67.300	100.684	79.603	615.768	589.009	623.100

Example: adjnoun
Table 6: Computational results for adjnoun

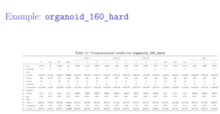
Example: `organoid_160_hard`

Table 11: Computational results for `organoid_160_hard`

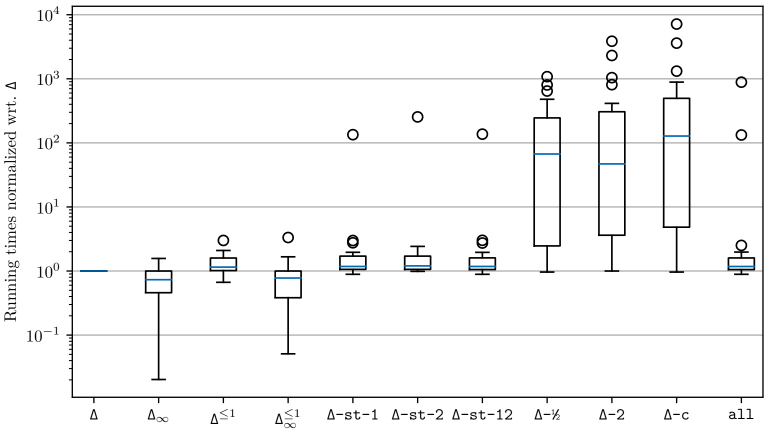
						`Δ-st-1`			`Δ-st-2`			`Δ-st-12`						`all`		
1		Δ	Δ_∞	$\Delta^{\leq 1}$	$\Delta^{\leq 1}_\infty$	1	2	3	1	2	3	1	2	3	`Δ-½`	`Δ-2`	`Δ-c`	1	2	3
2	# it.	3,712	**6**	10,000	20	4,060	4,152	3,929	4,000	4,186	4,030	4,037	3,990	3,917	3,712	3,712	3,712	4,081	3,928	3,922
3	# $\not\Delta$-calls	0	0	0	0	1	1	1	2	1	1	2	1	2	1	1	2	4	1	2
4	cuts																			
5	# total	1,482,621	131,841	1,729,012	**15,901**	1,621,281	1,657,361	1,568,785	1,592,542	1,669,574	1,608,612	1,609,391	1,591,066	1,562,088	1,482,621	1,482,621	1,482,621	1,626,670	1,568,107	1,563,353
6	# max	400	65,771	237	1,450	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400
7	# min	16	4	53	21	6	6	6	1	3	6	2	5	3	16	16	16	2	6	5
8	# $\not\Delta$-cuts	0	0	0	0	6	6	6	24	6	6	30	5	9	0	0	0	17	6	15
9	# removed	1,476,839	55,589	1,723,930	10,233	1,615,450	1,652,117	1,563,129	1,586,880	1,663,428	1,602,578	1,602,759	1,584,994	1,556,817	1,476,839	1,476,839	1,476,839	1,619,226	1,561,818	1,558,062
10	objective																			
11	bound	135.5	135.5	136.3	135.5	**135.3**	**135.3**	**135.3**	**135.3**	**135.3**	**135.3**	**135.3**	**135.3**	**135.3**	135.5	135.5	135.5	**135.3**	**135.3**	**135.3**
12	gap	0.1%	0.1%	0.7%	0.1%	0.0%*	0.0%*	0.0%*	0.0%*	0.0%*	0.0%*	0.0%*	0.0%*	0.0%*	0.1%	0.1%	0.1%	0.0%*	0.0%*	0.0%*
13	time																			
14	total 1/s	286.655	140.807	599.619	**14.621**	305.721	309.898	300.815	302.782	311.934	305.005	305.710	302.744	300.734	846.531	1192.236	1752.368	307.655	301.564	300.637
15	normalized	1.00	0.49	2.09	**0.05**	1.07	1.08	1.05	1.06	1.09	1.06	1.07	1.06	1.05	2.95	4.16	6.11	1.07	1.05	1.05
16	lp time 1/s	280.912	140.651	586.847	**14.398**	299.461	303.465	294.748	296.512	305.446	298.737	299.366	296.552	294.604	281.039	281.239	281.777	301.282	295.407	294.493

2023-11-21

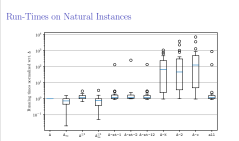
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Empirical Results
└─Example: `organoid_160_hard`



Run-Times on Natural Instances

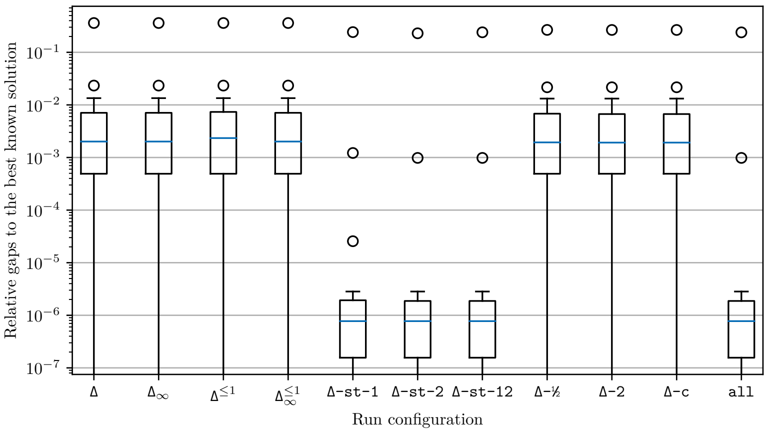


2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Empirical Results
└─Run-Times on Natural Instances

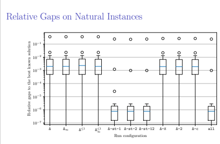


1. Running time of run configurations on non-random instances.
2. For run configurations containing a `st`-separator the results were averaged per instance among the three runs.
3. All results are normalized with respect to the `Δ` run configuration (first column).
4. The ordinate is logarthmically scaled. Box plots are drawn in the standard way, *i.e.*, boxes are from Q_1 to Q_3, the blue mark highlights the median, and whiskers extend by $1.5 * \mathrm{IQR}$.
5. are cycles worth it?

Relative Gaps on Natural Instances

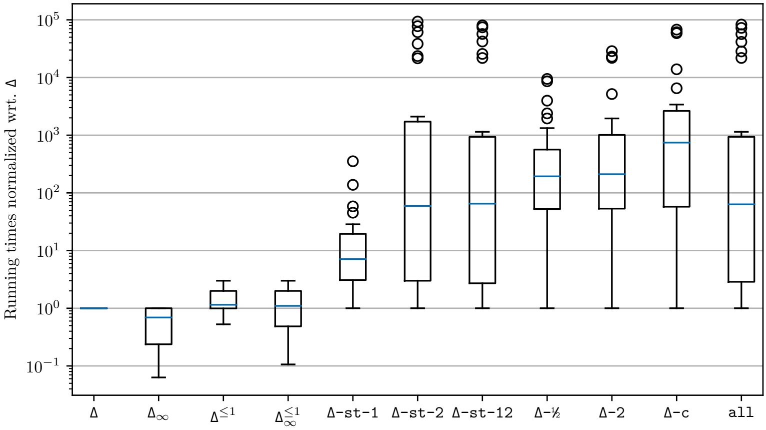


2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Empirical Results
└─Relative Gaps on Natural Instances



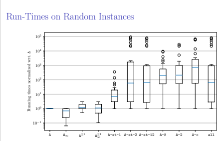
1. st-separators artifact of measuring integrality with tolerance parameter

Run-Times on Random Instances

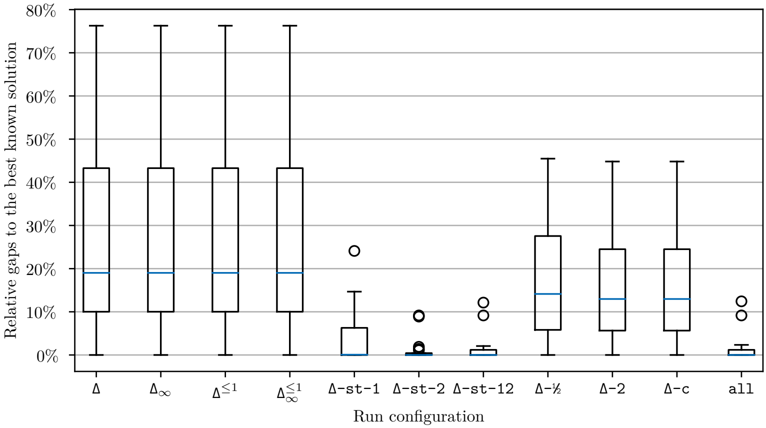


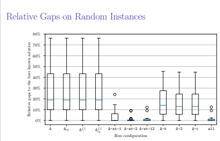
2023-11-21

Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└Empirical Results
└Run-Times on Random Instances



Relative Gaps on Random Instances





1. axis is now linear!
2. explain underlying structure in natural instances to show why they are easier

Conclusion

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Conclusion

Conclusion

Conclusion

- Δ and `st` often sufficient for integral solutions
- `two` and `half` run longer and do only improve bounds marginally

Conclusion

- Δ and st often sufficient for integral solutions
- two and half run longer and do only improve bounds marginally

1. Everything with a note: on the instances i tested
2. Personal: great fun, learned much
3. would not have started it if i knew how little i knew ((i)lp, optimization, c++), 10/10

Reference I

[And+22] Bjoern Andres et al. "A Polyhedral Study of Lifted Multicuts". In: *Discrete Optimization* 47 (Feb. 2022), p. 100757. issn: 15725286. doi: 10.1016/j.disopt.2022.100757. url: https://linkinghub.elsevier.com/retrieve/pii/S1572528622000627 (visited on 10/16/2023).

[GW89] M. Grötschel and Y. Wakabayashi. "A Cutting Plane Algorithm for a Clustering Problem". In: *Mathematical Programming* 45.1-3 (Aug. 1989), pp. 59–96. issn: 0025-5610, 1436-4646. doi: 10.1007/BF01589097. url: http://link.springer.com/10.1007/BF01589097 (visited on 10/16/2023).

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Conclusion
└─Reference

Reference I

[And+22] Bjoern Andres et al. "A Polyhedral Study of Lifted Multicuts". In: *Discrete Optimization* 47 (Feb. 2022), p. 100757. issn: 15725286. doi: 10.1016/j.disopt.2022.100757. url: https://linkinghub.elsevier.com/retrieve/pii/S1572528622000627 (visited on 10/16/2023).

[GW89] M. Grötschel and Y. Wakabayashi. "A Cutting Plane Algorithm for a Clustering Problem". In: *Mathematical Programming* 45.1-3 (Aug. 1989), pp. 59–96. issn: 0025-5610, 1436-4646. doi: 10.1007/BF01589097. url: http://link.springer.com/10.1007/BF01589097 (visited on 10/16/2023).

Reference II

[GW90] M. Grötschel and Y. Wakabayashi. "Facets of the Clique Partitioning Polytope". In: *Mathematical Programming* 47 (May 1990), pp. 367–387. issn: 0025-5610, 1436-4646. doi: 10.1007/BF01580870. url: http://link.springer.com/10.1007/BF01580870 (visited on 10/16/2023).

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Conclusion
└─Reference

Reference II

[GW90] M. Grötschel and Y. Wakabayashi. "Facets of the Clique Partitioning Polytope". In: *Mathematical Programming* 47 (May 1990), pp. 367–387. issn: 0025-5610, 1436-4646. doi: 10.1007/BF01580870. url: http://link.springer.com/10.1007/BF01580870 (visited on 10/16/2023).

Reference III

[Kap+15] Jörg H. Kappes et al. "A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems". In: *International Journal of Computer Vision* 115.2 (Nov. 2015), pp. 155–184. issn: 0920-5691, 1573-1405. doi: 10.1007/s11263-015-0809-x. url: http://link.springer.com/10.1007/s11263-015-0809-x (visited on 10/27/2023).

[Mül96] Rudolf Müller. "On the Partial Order Polytope of a Digraph". In: *Mathematical Programming* 73.1 (Apr. 1996), pp. 31–49. issn: 0025-5610, 1436-4646. doi: 10.1007/BF02592097. url: http://link.springer.com/10.1007/BF02592097 (visited on 10/16/2023).

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└Conclusion
└Reference

Reference III
[Kap+15] Jörg H. Kappes et al. "A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems". In: International Journal of Computer Vision 115.2 (Nov. 2015), pp. 155–184. issn: 0920-5691, 1573-1405. doi: 10.1007/s11263-015-0809-x. url: http://link.springer.com/10.1007/s11263-015-0809-x (visited on 10/27/2023).
[Mül96] Rudolf Müller. "On the Partial Order Polytope of a Digraph". In: Mathematical Programming 73.1 (Apr. 1996), pp. 31–49. issn: 0025-5610, 1436-4646. doi: 10.1007/BF02592097. url: http://link.springer.com/10.1007/BF02592097 (visited on 10/16/2023).

Reference IV

[ORS01] Maarten Oosten, Jeroen H. G. C. Rutten, and Frits C. R. Spieksma. "The Clique Partitioning Problem: Facets and Patching Facets". In: *Networks* 38.4 (Dec. 2001), pp. 209–226. issn: 0028-3045, 1097-0037. doi: 10.1002/net.10004. url: https://onlinelibrary.wiley.com/doi/10.1002/net.10004 (visited on 10/16/2023).

[Pre23] Jannik Presberger. "Segmentation and Clustering of Organoids in Light Microscopy Images". Master Thesis (unpublished). Dresden, Saxony, Germany: Technische Universität Dresden, 2023.

Reference IV
[ORS01] Maarten Oosten, Jeroen H. G. C. Rutten, and Frits C. R. Spieksma. "The Clique Partitioning Problem: Facets and Patching Facets". In: *Networks* 38.4 (Dec. 2001), pp. 209–226. issn: 0028-3045, 1097-0037. doi: 10.1002/net.10004. url: https://onlinelibrary.wiley.com/doi/10.1002/net.10004 (visited on 10/16/2023).
[Pre23] Jannik Presberger. "Segmentation and Clustering of Organoids in Light Microscopy Images". Master Thesis (unpublished). Dresden, Saxony, Germany: Technische Universität Dresden, 2023.

Reference V

[Wak86] Yoshiko Wakabayashi. “Aggregation of Binary Relations: Algorithmic and Polyhedral Investigations”. In: 1986. url: https://www.semanticscholar.org/paper/Aggregation-of-binary-relations%3A-algorithmic-and-Wakabayashi/75262f6acccb09118efb517efba46c1c7957c2a9 (visited on 10/20/2023).

2023-11-21
Evaluation of separation routines for some classes of inequalities of the Clique Partitioning polytope
└─Conclusion
└─Reference

Reference V

[Wak86] Yoshiko Wakabayashi. “Aggregation of Binary Relations: Algorithmic and Polyhedral Investigations”. In: 1986. url: https://www.semanticscholar.org/paper/Aggregation-of-binary-relations%3A-algorithmic-and-Wakabayashi/75262f6acccb09118efb517efba46c1c7957c2a9 (visited on 10/20/2023).