Column generation for the cut packing problem

Martin Bergner Marco Lübbecke

Operations Research
RWTH Aachen University

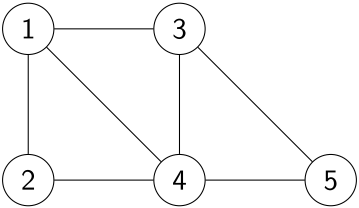
June 11, 2012
Column generation

Intro

- Goal: find a cut packing of maximal size
- What is cut packing?

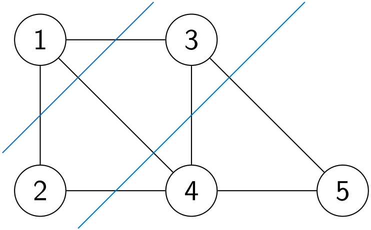
Intro

- Goal: find a cut packing of maximal size
- What is cut packing?



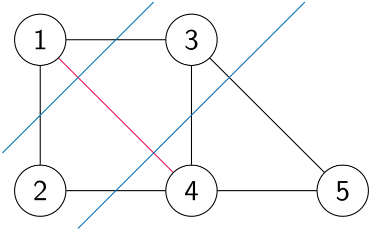
Intro

- Goal: find a cut packing of maximal size
- What is cut packing?



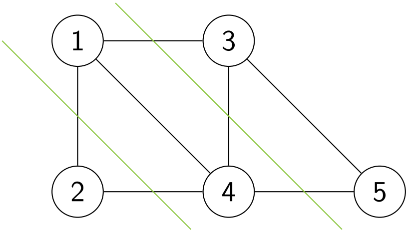
Intro

- Goal: find a cut packing of maximal size
- What is cut packing?



Intro

- Goal: find a cut packing of maximal size
- What is cut packing?



Hardness

- Cut packing is hard in general
  - Colbourn showed that cut packing is NP-hard in general
  - Caprara et.al. showed several more hardness results
  - Cut packing is as approximable as independent set (losing a factor of 2)
- Easy for chordal and directed graphs
- Applications in matrix structure detection and also in bioinformatics

Column generation model

$$\max \sum_{c \in C} x_c \tag{1}$$

$$s.t. \sum_{c \in C} \delta_c^e x_c \leq 1, \qquad \forall e \in E \tag{2}$$

$$x_c \in \{0,1\}, \quad \forall c \in C \tag{3}$$

- Variable $x_c \in \{0,1\}$ indicates whether cut chosen or not
- Constraints for every edge enforcing maximally one cut per edge

RWTH AACHEN UNIVERSITY

Column generation model

$$\max \sum_{c \in C} x_c \tag{1}$$

$$s.t. \sum_{c \in C} \delta_c^e x_c + \bar{x}_e = 1, \qquad \forall e \in E \tag{2}$$

$$\bar{x}_e, x_c \in \{0, 1\}, \quad \forall c \in C, \forall e \in E \tag{3}$$

- Variable $x_c \in \{0, 1\}$ indicates whether cut chosen or not
- Constraints for every edge enforcing maximally one cut per edge

Column generation model

$$\max \sum_{c \in C} x_c \tag{1}$$

$$s.t. \sum_{c \in C} \delta_c^e x_c + \bar{x}_e = 1, \qquad \forall e \in E \tag{2}$$

$$\bar{x}_e, x_c \in \{0, 1\}, \quad \forall c \in C, \forall e \in E \tag{3}$$

- Variable $x_c \in \{0, 1\}$ indicates whether cut chosen or not
- Constraints for every edge enforcing maximally one cut per edge

- What is the pricing problem?

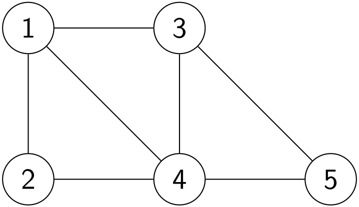
Pricing problem

- Pricing is a minimal cut problem
  - Combinatorial (e.g. Stoer-Wagner) or as a MIP
  - Possible integration of cutting planes
  - Difficult to integrate branching decisions in combinatorial algorithms
- We prefer MIP as it easy to include branching and cutting plane information

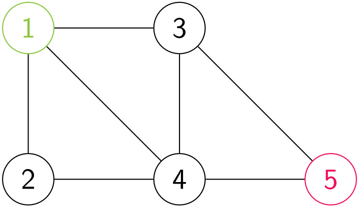
Pricing problem

- Pricing is a minimal cut problem
  - Combinatorial (e.g. Stoer-Wagner) or as a MIP
  - Possible integration of cutting planes
  - Difficult to integrate branching decisions in combinatorial algorithms
- We prefer MIP as it easy to include branching and cutting plane information
- Potential based
  - $y_{ij} = 1$ iff edge (i, j) is in the cut
  - $u_1 = 0$ and $u_i = 1$ for some $i \in \{2, \ldots, n\}$
  - $y_{ij} = 0$ iff $u_i = u_j$, $\forall (i, j) \in A'$
  - $y_{ij} = 1$ iff $u_i \neq u_i$, $\forall (i, j) \in A'$

Pricing problem – Visualization

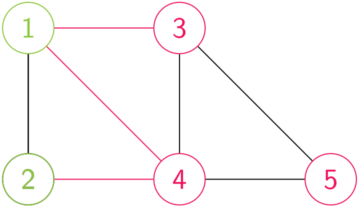


Pricing problem – Visualization



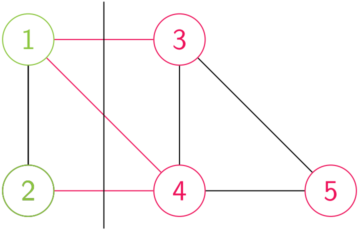
- Green nodes: $u_i = 0$; red nodes: $u_i = 1$

Pricing problem – Visualization



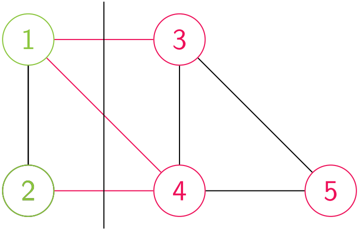
- Green nodes: $u_i = 0$; red nodes: $u_i = 1$
- For red edges it holds that $y_{ij} = 1$

Pricing problem – Visualization



- Green nodes: $u_i = 0$; red nodes: $u_i = 1$
- For red edges it holds that $y_{ij} = 1$

Pricing problem – Visualization

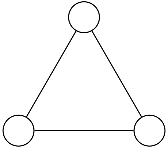


- Green nodes: $u_i = 0$; red nodes: $u_i = 1$
- For red edges it holds that $y_{ij} = 1$
- Needs to work with arbitrary edge weights

Pricing problem – MIP model

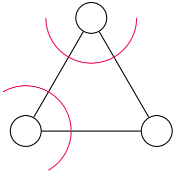
$$
\begin{array}{rll}
\min & \displaystyle\sum_{(i,j)\in=A'} c_{ij} y_{ij} & \\
\textit{subject to} & u_i - u_j + y_{ij} \geq 0 & \forall (i,j) \in A' \\
& u_j - u_i + y_{ij} \geq 0 & \forall (i,j) \in A' \\
& y_{ij} - u_i - u_j \leq 0 & \forall (i,j) \in A' \\
& u_i + u_j + y_{ij} \leq 2 & \forall (i,j) \in A' \\
& & \\
& \displaystyle\sum_{i\in V} u_i \geq 1 & \\
& & \\
& u_1 = 0 & \\
& y \geq 0, u \geq 0 &
\end{array}
$$

Branching – Visualization



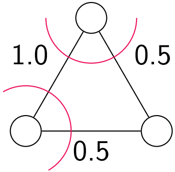
- Two (different) cases:

Branching – Visualization



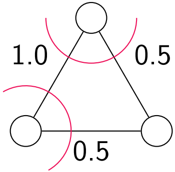
- Two (different) cases:

Branching – Visualization



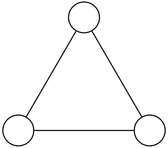
- Two (different) cases:

Branching – Visualization



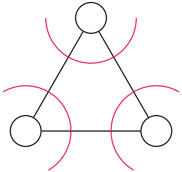
- Two (different) cases:
    1. Branching on original variables

Branching – Visualization



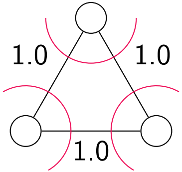
- Two (different) cases:
  1. Branching on original variables

Branching – Visualization



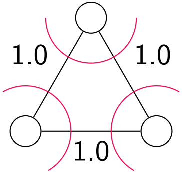
- Two (different) cases:
  1. Branching on original variables

Branching – Visualization



- Two (different) cases:
  1. Branching on original variables

Branching – Visualization



- Two (different) cases:
  1. Branching on original variables
  2. Ryan-Foster branching

Branching

- Ryan-Foster branching on pairs of edges
  - Can be respected in the pricing problem as simple as a constraints $u_i + u_j \leq 1$ resp. $u_i - u_j = 0$
- Minimal cut for the DIFF-branch is NP-hard in general

- Branching on slack variables $\bar{x}_e$
  - Can be respected in the pricing problem as bound changes $y_{ij} = 1 - \bar{x}_e$ with $e = (v_i, v_j)$

Cutting planes

- LP bound is bad
- There is a feasible solution with value $\frac{n}{2}$ for K_n, optimal solution has value 1

Cutting planes

- LP bound is bad
- There is a feasible solution with value $\frac{n}{2}$ for K_n, optimal solution has value 1

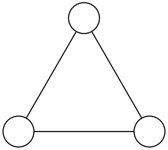
- Add cutting planes
  - Odd hole cuts: $\sum_{c \in C} \delta_c^O x_c \leq |O| - 1, \; \forall O \in \mathcal{O}$
  - Clique cuts: $\sum_{c \in C} \delta_c^K x_c \leq 1, \; \forall K \in \mathcal{C}$

Cutting planes

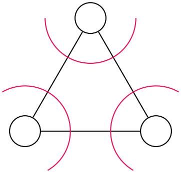
- LP bound is bad
- There is a feasible solution with value $\frac{n}{2}$ for K_n, optimal solution has value 1

- Add cutting planes
  - Odd hole cuts: $\sum_{c \in C} \delta_c^O x_c \leq |O| - 1, \forall O \in \mathcal{O}$
  - Clique cuts: $\sum_{c \in C} \delta_c^K x_c \leq 1, \forall K \in \mathcal{C}$

Clique cuts

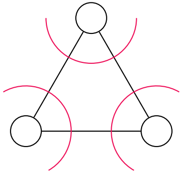


Clique cuts



- $$\sum_{c \in C} \delta_c^K x_c \leq 1, \forall K \in \mathcal{C}$$

Clique cuts



- Integration in Pricing is not difficult
  - $\sum_{c \in C} \delta_c^K x_c \implies y_{ij} \leq z_K, \quad \forall (i,j) \in K$
  - add $\zeta_K z_K$ to objective function of pricing problem

Results

- What is the influence of cuts on the solution process
- Random graphs
  - 10 nodes
  - 25 nodes
  - 50 nodes

Random graphs

- 10 nodes
  - Turning cuts off is 16% slower (geom. mean)
  - It needs 4x more nodes
- 25 nodes
  - With cuts: Solving time in geom. mean $\approx 1min$
  - Without cuts: no instance solved withing 1h time limit

Random graphs – 50 nodes

	default		no cuts	
	Nodes	Gap	Nodes	Gap
rand_50_0	3	1612.50%	162	300.00%
rand_50_1	8	360.70%	724	300.00%
rand_50_2	10	205.00%	305	300.00%
rand_50_3	1	511.10%	256	500.00%
rand_50_4	19	191.70%	225	370.00%
rand_50_5	14	775.00%	232	700.00%
rand_50_6	1	149.50%	136	300.00%
rand_50_7	3	193.80%	198	500.00%
rand_50_8	5	337.50%	157	300.00%
rand_50_9	7	120.00%	239	300.00%

Random graphs – 50 nodes

	default		no cuts	
	Nodes	Dual	Nodes	Dual
rand_50_0	3	17.1	162	24
rand_50_1	8	18.4	724	24
rand_50_2	10	18.3	305	24
rand_50_3	1	18.3	256	24
rand_50_4	19	17.5	225	23.5
rand_50_5	14	17.5	232	24
rand_50_6	1	9.9	136	24
rand_50_7	3	11.8	198	24
rand_50_8	5	17.5	157	24
rand_50_9	7	15.4	239	24

Conclusions

- We are able to solve cutpacking instances with up to 40 edges in 1 hour time limit
- The clique formulation of the problem is much stronger
- Initialization takes long

Outlook

- Use combinatorial algorithm when possible
  - nonnegative edge weights
  - no cuts added
- Add multiple solutions per pricing iteration
- Looking for more cuts
  - Separate odd hole cuts
  - Look for cuts in slack variables

The end

Thank you for your attention.
Questions?