The Primal-Dual Greedy Algorithm for Covering Problems

Britta Peis[1], José Verschae[2], Andreas Wierz[1]

[1] RWTH Aachen University, [2]Pontificia Universidad Católica de Chile

January 2017, Aussois

Covering integer program:

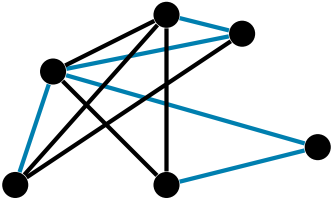
$$\min\{c^T x \mid Ax \geq r, 0 \leq x \leq d, x \text{ integral}\},$$

where $A \in \mathbb{Z}_+^{\mathcal{L} \times E}, r \in \mathbb{Z}^{\mathcal{L}}, c \in \mathbb{Z}_+^E, d \in \mathbb{Z}_+^E$

Example: Spanning trees

$$\min\left\{c^T x \mid \sum_{e \notin S} x_e \geq r(S)\ \forall S \subseteq E, x \in \{0,1\}\right\}$$

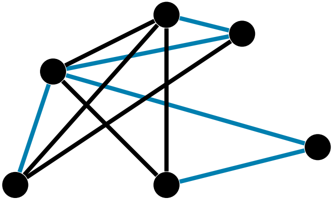
- $r(S) = n - 1 - \max\{|F| : F \subseteq S, F \text{ is cycle-free}\}$
- $A_{S,e} = \begin{cases} 1 & e \notin S \\ 0 & e \in S \end{cases}$



Example: Spanning trees

$$\min\left\{c^T x \mid \sum_{e \notin S} x_e \geq r(S)\ \forall S \subseteq E, x \in \{0,1\}\right\}$$

- $r(S) = n - 1 - \max\{|F| : F \subseteq S, F \text{ is cycle-free}\}$
- $A_{S,e} = \begin{cases} 1 & e \notin S \\ 0 & e \in S \end{cases}$



1. Sort edges $c_1 \leq c_2 \leq \cdots \leq c_{|E|}$, set $i = 1, S = \emptyset$
2. If $S + e_i$ is cycle-free, set $S = S + e_i$. Iterate with $i = i + 1$

Example: Knapsack cover

$$\min\left\{c^T x \mid \sum_{e\in E} u_e x_e \geq D, x \in \{0,1\}\right\},$$

where $u_e \in \mathbb{Z}_+, D \in \mathbb{Z}$. Unbounded integrality gap, but...

Example: Knapsack cover

$$\min\left\{c^T x \mid \sum_{e\in E} u_e x_e \geq D, x \in \{0,1\}\right\},$$

where $u_e \in \mathbb{Z}_+, D \in \mathbb{Z}$. Unbounded integrality gap, but...

$$\min\left\{c^T x \mid \sum_{e\notin S} A_{S,e} x_e \geq r(S)\ \forall S \subseteq E, x \geq 0\right\},$$

$$r(S) = D - \sum_{e\in S} u_e \qquad A_{S,e} = \begin{cases} \min\{u_e, r(S)\} & e \notin S \\ 0 & e \in S \end{cases}$$

has integrality gap 2.

Example: Knapsack cover

$$\min\left\{c^T x \mid \sum_{e\notin S} A_{S,e}x_e \geq r(S)\ \forall S \subseteq E, x \geq 0\right\},$$

$$r(S) = D - \sum_{e\in S} u_e \qquad A_{S,e} = \begin{cases}\min\{u_e, r(S)\} & e \notin S\\ 0 & e \in S\end{cases}$$

has integrality gap 2. Consider the dual to the linear relaxation

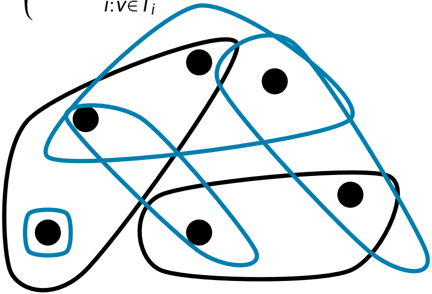
$$\max\left\{y^T r \mid \sum_{S:e\notin S} A_{S,e}y_S \leq c_e, y \geq 0\right\},$$

1. $S = \emptyset$.
2. Increase y_S until some element $e \notin S$ gets tight
3. Add e to S and iterate until $r(S) \leq 0$

Example: Subset cover

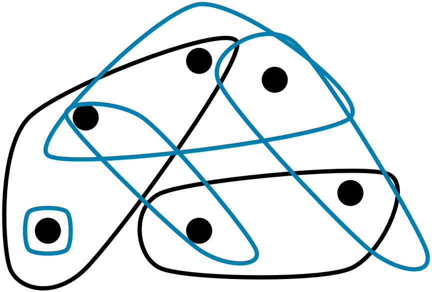
- Groundset G
- Subsets $T_i \subseteq G, 1 \leq i \leq n$
- Task: Find $S \subseteq [n]$ of minimum cost with $\cup_{i \in S} T_i = G$

$$\min \left\{ c^T x \mid \sum_{i : v \in T_i} x_i \geq 1 \; \forall v \in G, x \in \{0,1\} \right\}$$



Example: Subset cover

$$\min\left\{c^T x \mid \sum_{i:v\in T_i} x_i \geq 1 \ \forall v \in G, x \in \{0,1\}\right\}$$



1. $S = \emptyset$
2. Increase y_v for $v \notin \cup_{i\in S} T_i$ until some set T_i becomes tight
3. Add i to S and iterate

- Yields an f-approximation, where f is the max number of sets, any element is contained in

Greedy algorithm

- Let $\mathcal{L} = (2^E, \subseteq, \cup, \cap)$ be the boolean lattice

$$\min\left\{c^T x \mid \sum_{e \notin S} A_{S,e} x_e \geq r(S)\ \forall S \in \mathcal{L}, x \geq 0, x \text{ integral}\right\}$$

$$\max\left\{y^T b \mid \sum_{S : e \notin S} A_{S,e} y_S \leq c_e\ \forall e \in E, y \geq 0\right\}$$

Greedy algorithm

- Let $\mathcal{L} = (2^E, \subseteq, \cup, \cap)$ be the boolean lattice

$$\min\left\{c^T x \mid \sum_{e \notin S} A_{S,e} x_e \geq r(S)\ \forall S \in \mathcal{L}, x \geq 0, x \text{ integral}\right\}$$

$$\max\left\{y^T b \mid \sum_{S: e \notin S} A_{S,e} y_S \leq c_e\ \forall e \in E, y \geq 0\right\}$$

1. $S = \emptyset$. If $S = E$ or $r(S) \leq 0$, STOP
2. Increase y_S until some element $e \notin S$ becomes tight
3. Set $x_e = \left\lceil \frac{r(S) - r(S+e)}{A_{S,e}} \right\rceil$
4. Set $S = S + e$ and iterate

Goal: Properties of (A, r) such that:

1. the greedy algorithm obtains a feasible solution
2. with solution cost bounded by $\alpha \cdot OPT$

Goal: Properties of (A, r) such that:

1. **the greedy algorithm obtains a feasible solution**
2. with solution cost bounded by $\alpha \cdot OPT$

Feasibility is not always clear:

1. $S = \emptyset$. If $S = E$ or $r(S) \leq 0$, STOP
2. Increase y_S until some element $e \notin S$ becomes tight
3. Set $x_e = \left\lceil \frac{r(S) - r(S+e)}{A_{S,e}} \right\rceil$
4. Set $S = S + e$ and iterate

Feasibility is not always clear:

1. $S = \emptyset$. If $S = E$ or $r(S) \leq 0$, STOP
2. Increase y_S until some element $e \notin S$ becomes tight
3. Set $x_e = \left\lceil \frac{r(S) - r(S+e)}{A_{S,e}} \right\rceil$
4. Set $S = S + e$ and iterate

$$\begin{aligned} \min x_1 + 2x_2 \\ 2x_1 + x_2 \geq 3 \\ x_2 \geq 1 \\ x_1 \geq 2 \\ 0 \geq 0 \end{aligned}$$

Feasibility is not always clear:

1. $S = \emptyset$. If $S = E$ or $r(S) \leq 0$, STOP
2. Increase y_S until some element $e \notin S$ becomes tight
3. Set $x_e = \left\lceil \frac{r(S) - r(S+e)}{A_{S,e}} \right\rceil$
4. Set $S = S + e$ and iterate

$$\min x_1 + 2x_2$$

$$2x_1 + x_2 \geq 3 \leftarrow \mathbf{y}_\emptyset = \frac{1}{2}$$

$$x_2 \geq 1$$

$$x_1 \geq 2$$

$$0 \geq 0$$

Feasibility is not always clear:

1. $S = \emptyset$. If $S = E$ or $r(S) \leq 0$, STOP
2. Increase y_S until some element $e \notin S$ becomes tight
3. Set $x_e = \left\lceil \frac{r(S) - r(S+e)}{A_{S,e}} \right\rceil$
4. Set $S = S + e$ and iterate

$$\begin{aligned} &\min x_1 + 2x_2 \\ 2x_1 + x_2 &\geq 3 \leftarrow \mathbf{y}_\emptyset = \frac{\mathbf{1}}{\mathbf{2}} \\ x_2 &\geq 1 \leftarrow \mathbf{y_1} = \frac{\mathbf{3}}{\mathbf{2}} \\ x_1 &\geq 2 \\ 0 &\geq 0 \end{aligned}$$

- obtains $x = (1, 1)$, which is infeasible for $S = \{2\}$

Recall: Matroids (Spanning trees)

- $\mathcal{L}$ is the boolean lattice
- $A_{S,e} = \begin{cases} 1 & e \notin S \\ 0 & e \in S \end{cases}$
- r is supermodular:

$$r(S) + r(T) \leq r(S \cup T) + r(S \cap T) \ \forall S, T \in \mathcal{L}$$

- $r(S) \geq r(S + e) \geq r(S) - 1 \ \forall S \in \mathcal{L}, e \notin S$

yields optimality of the greedy algorithm for

$$\min \left\{ c^T x \mid Ax \geq r, x \in \{0, 1\} \right\}$$

Recall: Matroids (Spanning trees)

- $\mathcal{L}$ is the boolean lattice
- $A_{S,e} = \begin{cases} 1 & e \notin S \\ 0 & e \in S \end{cases}$
- r is supermodular:

$$r(S) + r(T) \leq r(S \cup T) + r(S \cap T) \; \forall S, T \in \mathcal{L}$$

- $r(S) \geq r(S + e) \geq r(S) - 1 \; \forall S \in \mathcal{L}, e \notin S$

yields optimality of the greedy algorithm for

$$\min \left\{ c^T x \mid Ax \geq r, x \in \{0, 1\} \right\}$$

...but we want: $A \in \mathbb{Z}_+^{\mathcal{L} \times E}$

Definition: A system (A, r) with $A \in \mathbb{Z}_+^{\mathcal{L} \times E}$, $r \in \mathbb{Z}^{\mathcal{L}}$ and $\mathcal{L}$ is the boolean lattice on E is said to be **nice**, if

- r is supermodular

Definition: A system (A, r) with $A \in \mathbb{Z}_+^{\mathcal{L} \times E}$, $r \in \mathbb{Z}^{\mathcal{L}}$ and $\mathcal{L}$ is the boolean lattice on E is said to be **nice**, if

- r is supermodular
- $0 \leq r(S) - r(S + e) \leq A_{S,e}$ for all $S \in \mathcal{L}, e \notin S$

Definition: A system (A, r) with $A \in \mathbb{Z}_+^{\mathcal{L} \times E}, r \in \mathbb{Z}^{\mathcal{L}}$ and $\mathcal{L}$ is the boolean lattice on E is said to be **nice**, if

- r is supermodular
- $0 \leq r(S) - r(S + e) \leq A_{S,e}$ for all $S \in \mathcal{L}, e \notin S$
- A is submodular for all $e \in E$:

$$A_{S,e} + A_{T,e} \geq A_{S \cup T,e} + A_{S \cap T,e} \; \forall S, T \in \mathcal{L}$$

Definition: A system (A, r) with $A \in \mathbb{Z}_+^{\mathcal{L}\times E}$, $r \in \mathbb{Z}^{\mathcal{L}}$ and $\mathcal{L}$ is the boolean lattice on E is said to be **nice**, if

- r is supermodular
- $0 \leq r(S) - r(S+e) \leq A_{S,e}$ for all $S \in \mathcal{L}, e \notin S$
- A is submodular for all $e \in E$:

$$A_{S,e} + A_{T,e} \geq A_{S\cup T,e} + A_{S\cap T,e} \;\forall S, T \in \mathcal{L}$$

- A is monotone decreasing:

$$A_{S,e} \geq A_{T,e} \;\forall S \subset T$$

Definition: A system (A, r) with $A \in \mathbb{Z}_+^{\mathcal{L}\times E}$, $r \in \mathbb{Z}^{\mathcal{L}}$ and $\mathcal{L}$ is the boolean lattice on E is said to be **nice**, if

- r is supermodular
- $0 \leq r(S) - r(S+e) \leq A_{S,e}$ for all $S \in \mathcal{L}, e \notin S$
- A is submodular for all $e \in E$:

$$A_{S,e} + A_{T,e} \geq A_{S\cup T,e} + A_{S\cap T,e} \; \forall S, T \in \mathcal{L}$$

- A is monotone decreasing:

$$A_{S,e} \geq A_{T,e} \; \forall S \subset T$$

- $e \in S$ implies $A_{S,e} = 0$

Definition: A system (A, r) with $A \in \mathbb{Z}_+^{\mathcal{L} \times E}$, $r \in \mathbb{Z}^{\mathcal{L}}$ and $\mathcal{L}$ is the boolean lattice on E is said to be **nice**, if

- r is supermodular
- $0 \leq r(S) - r(S + e) \leq A_{S,e}$ for all $S \in \mathcal{L}, e \notin S$
- A is submodular for all $e \in E$:

$$A_{S,e} + A_{T,e} \geq A_{S \cup T,e} + A_{S \cap T,e} \ \forall S, T \in \mathcal{L}$$

- A is monotone decreasing:

$$A_{S,e} \geq A_{T,e} \ \forall S \subset T$$

- $e \in S$ implies $A_{S,e} = 0$

Proposition 1: Let (A, r) be a nice system and let $x \in \{0, 1\}^E$ with $E' = \{e \in E : x_e = 1\}$. If $r(E') \leq 0$, then x is feasible for

$$\left\{Ax \geq r, x \geq 0, x \in \mathbb{Z}_+^E\right\}.$$

Proof: (1) Show that x is feasible on all chains
$\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$.

Proof: (1) Show that x is feasible on all chains $\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$.

- For S_ℓ : $A_{S_\ell} x = \sum_{e \notin S_\ell} A_{S_\ell, e} x_e = 0 \geq r(E')$

Proof: (1) Show that x is feasible on all chains
$\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$.

- For S_ℓ : $A_{S_\ell} x = \sum_{e \notin S_\ell} A_{S_\ell, e} x_e = 0 \geq r(E')$
- Suppose it is true for i, then:

$$A_{S_{i-1}} x = \sum_{j=i}^{\ell} A_{S_{i-1},j} = \sum_{j=i+1}^{\ell} A_{S_{i-1},j} + A_{S_{i-1},i}$$

$$\overset{\text{A monotone}}{\geq} \sum_{j=i+1}^{\ell} A_{S_i,j} + A_{S_{i-1},i}$$

$$\overset{\text{induction}}{\geq} r(S_i) + A_{S_{i-1},i}$$

$$\overset{\text{A-r-coupling}}{\geq} r(S_i) + r(S_{i-1}) - r(S_i) = r(S_{i-1})$$

Proof (cont.): (2) Define function $h(S) = r(S) - A_S x$. Then for every chain $\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$, $h(S)$ is monotone increasing

Proof (cont.): (2) Define function $h(S) = r(S) - A_S x$. Then for every chain $\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$, $h(S)$ is monotone increasing

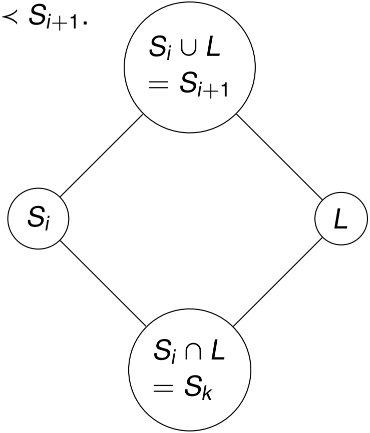
$$\begin{aligned} & h(S_{i+1}) \geq h(S_i) \\ \Leftrightarrow & r(S_{i+1}) - A_{S_{i+1}} x \geq r(S_i) - A_{S_i} x \\ \Leftrightarrow & r(S_i) - r(S_{i+1}) \leq \sum_{j=i+2}^{\ell} \underbrace{(A_{S_i} - A_{S_{i+1}})_j}_{\geq 0} x_j + A_{S_i, i+1} x_{i+1}. \end{aligned}$$

- True, as (A, r) is nice: $r(S) - r(S+e) \leq A_{S,e}$

Proof (cont.): (3) Show that $h(S) = r(S) - A_S x \leq 0 \ \forall S \in \mathcal{L}$

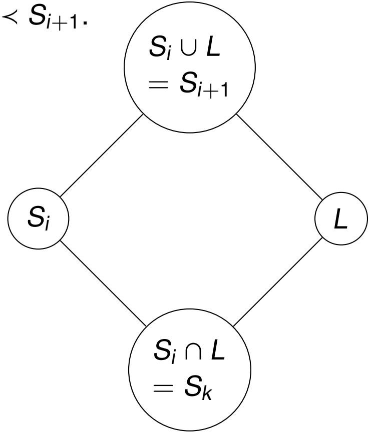
Proof (cont.): (3) Show that $h(S) = r(S) - A_S x \leq 0 \; \forall S \in \mathcal{L}$

- Suppose not, let $L \in \mathcal{L}$ be a maximal counterexample.
- Let $\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$ be a chain,
- let i be maximal with $L \not\preceq S_i, L \prec S_{i+1}$.



Proof (cont.): (3) Show that $h(S) = r(S) - A_S x \leq 0 \; \forall S \in \mathcal{L}$

- Suppose not, let $L \in \mathcal{L}$ be a maximal counterexample.
- Let $\emptyset = S_0 \subset S_1 \cdots \subset S_\ell = E', |E'| = \ell$ be a chain,
- let i be maximal with $L \not\preceq S_i, L \prec S_{i+1}$.



- h is supermodular:

$$h(L) \leq \underbrace{h(S_{i+1})}_{\leq 0} + \underbrace{h(S_k) - h(S_i)}_{\leq 0 \text{ h is monotone increasing}}$$

□

Goal: Properties of (A, r) such that:

1. the greedy algorithm obtains a feasible solution,
2. **with solution cost bounded by $\alpha \cdot$ OPT**

Still bad integrality gaps: Take a knapsack cover instance:

$$\sum_{e \in E} u_e x_e \geq D$$

and add redundant constraints

$$\sum_{e \notin S} u_e x_e \geq D - \sum_{e \in S} u_e \ \forall S \subset E.$$

It is a nice system (A, r) with $A_{S,e} = \begin{cases} u_e & e \notin S \\ 0 & e \in S. \end{cases}$

Truncation: Given a nice system (A, r), consider the **truncated sytem** (A', r) with

$$A'_{S,e} = r(S)^{+} - r(S+e)^{+} \ \forall S \in \mathcal{L}, e \notin S$$

Truncation: Given a nice system (A, r), consider the **truncated sytem** (A', r) with

$$A'_{S,e} = r(S)^{+} - r(S+e)^{+} \; \forall S \in \mathcal{L}, e \notin S$$

Observation:

- For knapsack cover, this yields the flow-cover inequalities
  $\rightarrow$ greedy is a 2-approximation

Truncation: Given a nice system (A, r), consider the **truncated sytem** (A', r) with

$$A'_{S,e} = r(S)^+ - r(S+e)^+ \ \forall S \in \mathcal{L}, e \notin S$$

Observation:

- For knapsack cover, this yields the flow-cover inequalities $\rightarrow$ greedy is a 2-approximation
- But (A', r) is no longer nice. A' is not submodular

Truncation: Given a nice system (A, r), consider the **truncated sytem** (A', r) with

$$A'_{S,e} = r(S)^+ - r(S + e)^+ \; \forall S \in \mathcal{L}, e \notin S$$

Observation:

- For knapsack cover, this yields the flow-cover inequalities $\rightarrow$ greedy is a 2-approximation
- But (A', r) is no longer nice. A' is not submodular

Proposition 2: Let (A, r) be a nice system and (A', r) its truncation. A vector $x \in \{0, 1\}^E$ satisfies $Ax \geq r$ if and only if it satisfies $A'x \geq r$.

Theorem: Let (A, r) be a nice system. Then the greedy algorithm applied to its truncation (A', r) obtains a feasible solution to

$$\min\{c^T x \mid A' x \geq r, x \geq 0, x \text{ integral}\}$$

Theorem: Let (A, r) be a nice system. Then the greedy algorithm applied to its truncation (A', r) obtains a feasible solution to

$$\min\{c^T x \mid A' x \geq r, x \geq 0, x \text{ integral}\}$$

Proof:

- Greedy algorithm computes $E' \subseteq E$ with $r(E') \leq 0$
- Proposition 1 implies that x is feasible for $Ax \geq r$
- Proposition 2 implies that x is feasible for $A' x \geq r$

□

Theorem: Let (A, r) be a nice system. Then the greedy algorithm applied to its truncation (A', r) obtains a feasible solution to

$$\min\{c^T x \mid A'x \geq r, x \geq 0, x \text{ integral}\}$$

Proof:

- Greedy algorithm computes $E' \subseteq E$ with $r(E') \leq 0$
- Proposition 1 implies that x is feasible for $Ax \geq r$
- Proposition 2 implies that x is feasible for $A'x \geq r$

□

Question: Can we get good bounds?

Bad news: Subset cover can be formulated as a nice system

- Groundset G
- Subsets $T_i \subseteq G, 1 \leq i \leq n$
- Task: Find $S \subseteq [n]$ of minimum cost with $\cup_{i \in S} T_i = G$

Bad news: Subset cover can be formulated as a nice system

- Groundset G
- Subsets $T_i \subseteq G, 1 \leq i \leq n$
- Task: Find $S \subseteq [n]$ of minimum cost with $\cup_{i \in S} T_i = G$

$$A_{S,i} = \begin{cases} |T_i| & i \notin S \\ 0 & i \in S \end{cases}$$

$$r(S) = n - |\cup_{i \in S} T_i|, \text{ for } S \subseteq [n]$$

Bad news: Subset cover can be formulated as a nice system

- Groundset G
- Subsets $T_i \subseteq G, 1 \leq i \leq n$
- Task: Find $S \subseteq [n]$ of minimum cost with $\cup_{i \in S} T_i = G$

$$A_{S,i} = \begin{cases} |T_i| & i \notin S \\ 0 & i \in S \end{cases}$$

$$r(S) = n - |\cup_{i \in S} T_i|, \text{ for } S \subseteq [n]$$

Corollary: There is no $(1 - o(1)) \log n$ approximation algorithm for the truncated system (A', r) of a nice system, unless $P = NP$.

Whats wrong?

- Seemed to work for matroids and knapsack,

Whats wrong?

- Seemed to work for matroids and knapsack,
- but... coefficients in A have a large range
  $\rightarrow$ What happens if the contribution is either large or zero?

Measurements δ and β: For all $S \in \mathcal{L}, e \notin S$ with $r(S) - r(S+e) > 0$ compute

$$\frac{A_{S,e}}{r(S) - r(S+e)}$$

Measurements δ and β: For all $S \in \mathcal{L}, e \notin S$ with $r(S) - r(S+e) > 0$ compute

$$\frac{A_{S,e}}{r(S) - r(S+e)}$$

Let δ be the maximum of these values.

Measurements δ and β: For all $S \in \mathcal{L}, e \notin S$ with $r(S) - r(S+e) > 0$ compute

$$\frac{A_{S,e}}{r(S) - r(S+e)}$$

Let δ be the maximum of these values.
Analogously compute

$$\frac{A_{\emptyset,e}}{A_{S,e}}$$

for S and $e \notin S$ with $A_{S,e} > 0$ and let β be the maximum.

Theorem: The greedy algorithm applied to a truncated sytem (A', r) of a nice system (A, r) yields a $(\beta\delta + 1)$-approximation.

Theorem: The greedy algorithm applied to a truncated sytem (A', r) of a nice system (A, r) yields a $(\beta\delta + 1)$-approximation. If r is non-negative, it yields a $\beta\delta$-approximation.

Theorem: The greedy algorithm applied to a truncated sytem (A', r) of a nice system (A, r) yields a $(\beta\delta + 1)$-approximation. If r is non-negative, it yields a $\beta\delta$-approximation.

Values for the examples:

- Matroids: $\beta = \delta = 1$, r non-negative
- Knapsack cover: $\beta = \delta = 1$
- Subset cover: $\beta = 1, \delta = \max_i |T_i|$

Outlook:

- Replacement of $\mathcal{L}$ with different lattices... e.g. ringfamilies
- Close the gap between $\log \delta$ inapproximability and δ-approximation for $\beta = 1$