LP-based Approximation Algorithms for SetCover

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$
\begin{array}{lrcl}
\textbf{minimize} & c^{\top} x & & \\
\textbf{subject to} & Ax & \geq & b \\
 & x & \geq & 0
\end{array}
$$

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$
\begin{array}{lrcl}
\textbf{minimize} & c^\top x & & \\
\textbf{subject to} & Ax & \geq & b \\
& x & \geq & 0
\end{array}
$$

- Linear Program (LP) can be solved in polynomial time.

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

- Linear Program (LP) can be solved in polynomial time.
- Integer Linear Program (ILP): $x_i \in \mathbb{Z}$ (instead of $\mathbb{R}$). Solving ILPs is NP-hard.

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

- Linear Program (LP) can be solved in polynomial time.
- Integer Linear Program (ILP): $x_i \in \mathbb{Z}$ (instead of $\mathbb{R}$). Solving ILPs is NP-hard.

Assume we want to approximate a problem modeled as ILP

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

- Linear Program (LP) can be solved in polynomial time.
- Integer Linear Program (ILP): $x_i \in \mathbb{Z}$ (instead of $\mathbb{R}$). Solving ILPs is NP-hard.

Assume we want to approximate a problem modeled as ILP

- Important ingredient for approximation: lower bound

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

- Linear Program (LP) can be solved in polynomial time.
- Integer Linear Program (ILP): $x_i \in \mathbb{Z}$ (instead of $\mathbb{R}$). Solving ILPs is NP-hard.

Assume we want to approximate a problem modeled as ILP

- Important ingredient for approximation: lower bound

Which lower bounds do we know for the optimum of an ILP?

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

lower bound 1 (LP relaxation): For ILP with $x_i \in \{0, 1\}$ the optimum of relaxed LP with $0 \leq x_i \leq 1$ provides lower bound on optimum of ILP (or upper bound if maximizing)

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\intercal x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

lower bound 1 (LP relaxation): For ILP with $x_i \in \{0, 1\}$ the optimum of relaxed LP with $0 \leq x_i \leq 1$ provides lower bound on optimum of ILP (or upper bound if maximizing)

lower bound 2 (Duality): the value of a feasible dual solution, more later . . .

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

lower bound 1 (LP relaxation): For ILP with $x_i \in \{0, 1\}$ the optimum of relaxed LP with $0 \leq x_i \leq 1$ provides lower bound on optimum of ILP (or upper bound if maximizing)

lower bound 2 (Duality): the value of a feasible dual solution, more later . . .

LP-based approximation techniques

1. LP Rounding
2. Primal-Dual Method
3. Dual Fitting

Reminder: (Integer) Linear Programs

Optimize (i.e., minimize or maximize) a linear (objective) function subject to linear inequalities (constraints).

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

lower bound 1 (LP relaxation): For ILP with $x_i \in \{0,1\}$ the optimum of relaxed LP with $0 \leq x_i \leq 1$ provides lower bound on optimum of ILP (or upper bound if maximizing)

lower bound 2 (Duality): the value of a feasible dual solution, more later . . .

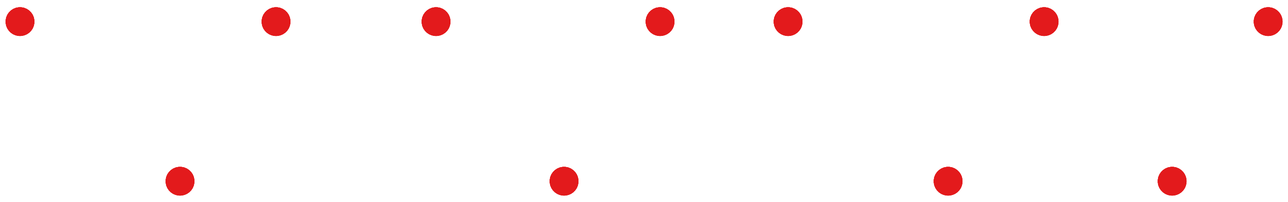
LP-based approximation techniques

1. LP Rounding
2. Primal-Dual Method
3. Dual Fitting

Example: Set Cover

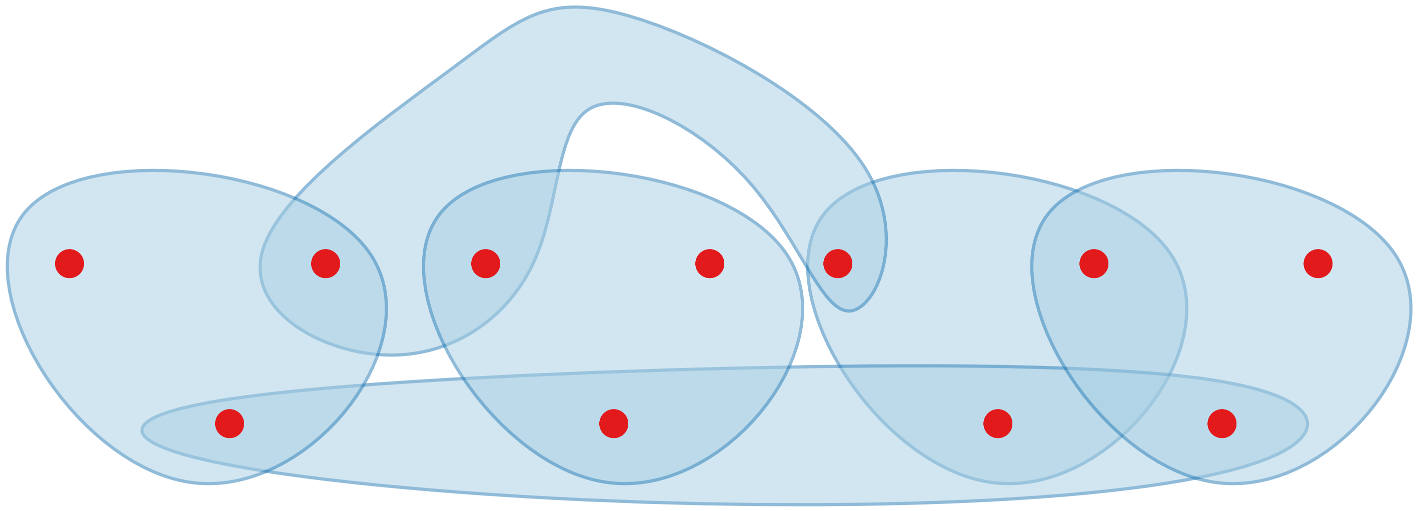
SetCover as an ILP

Ground set U



SetCover as an ILP

Ground set U
Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

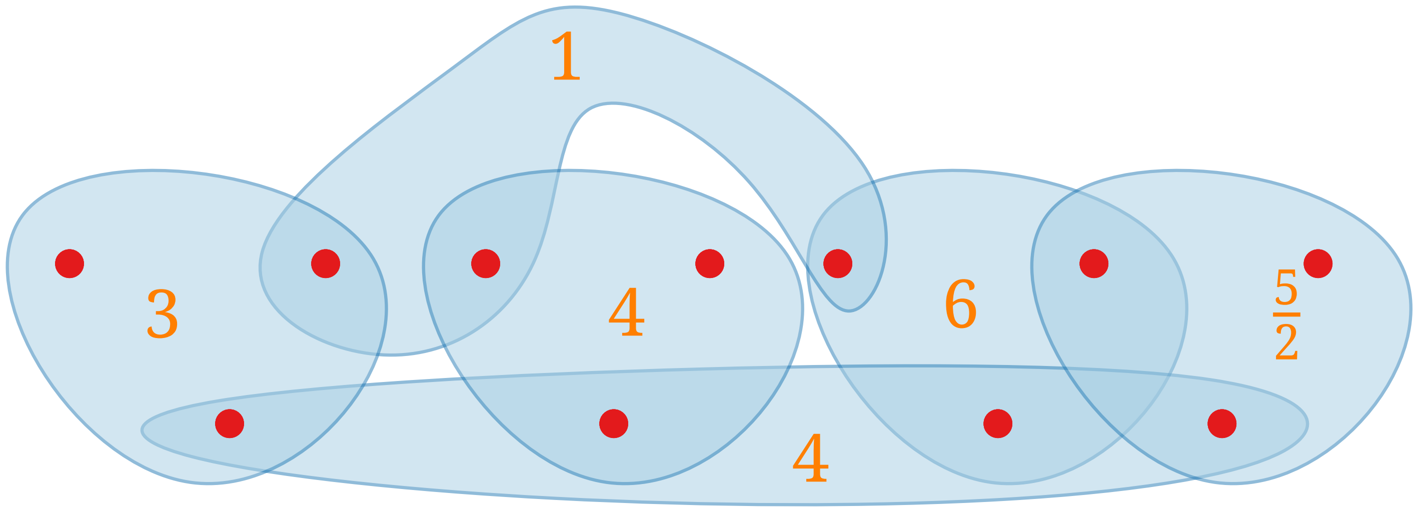


SetCover as an ILP

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c\colon \mathcal{S} \to \mathbb{Q}^+$

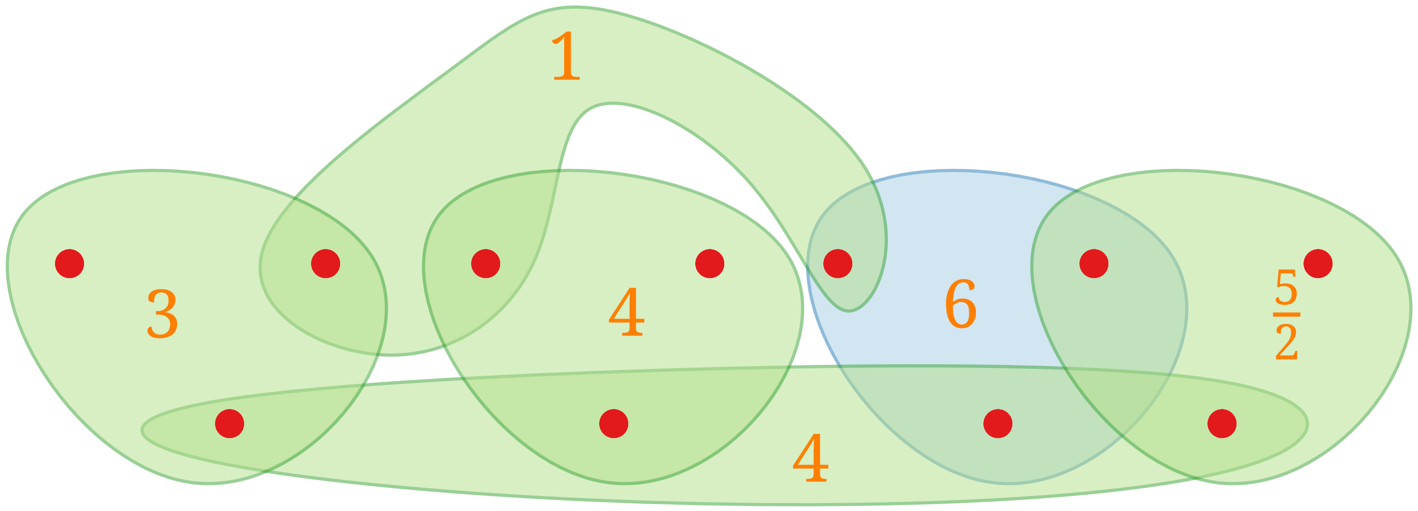


SetCover as an ILP

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

SetCover as an ILP

minimize

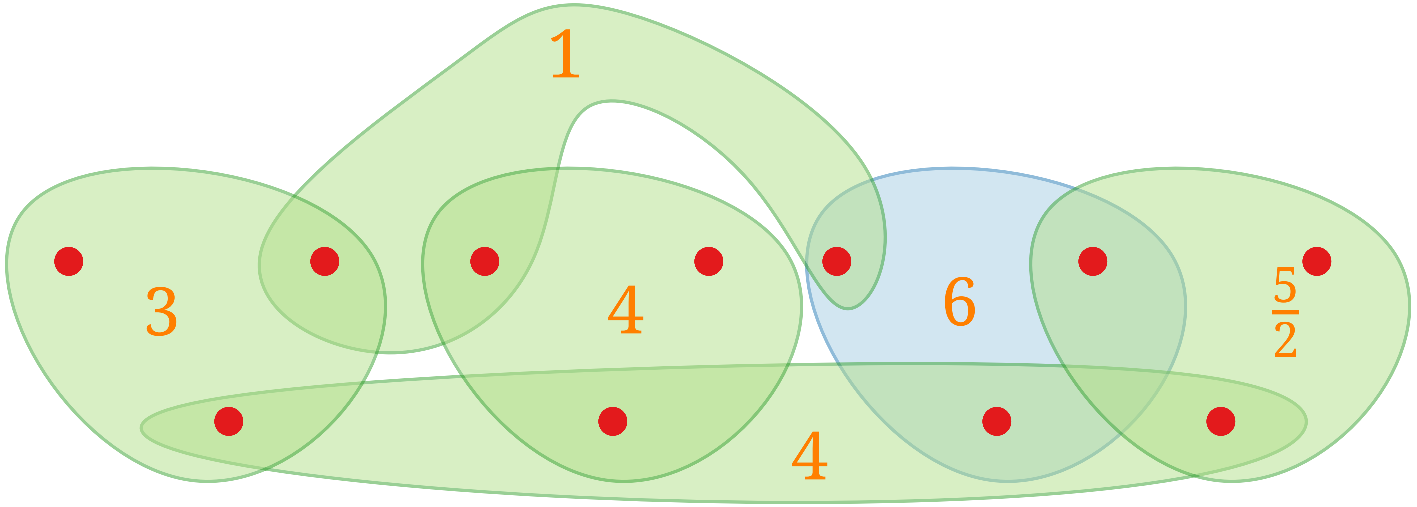
?

subject to

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

SETCOVER as an ILP

minimize

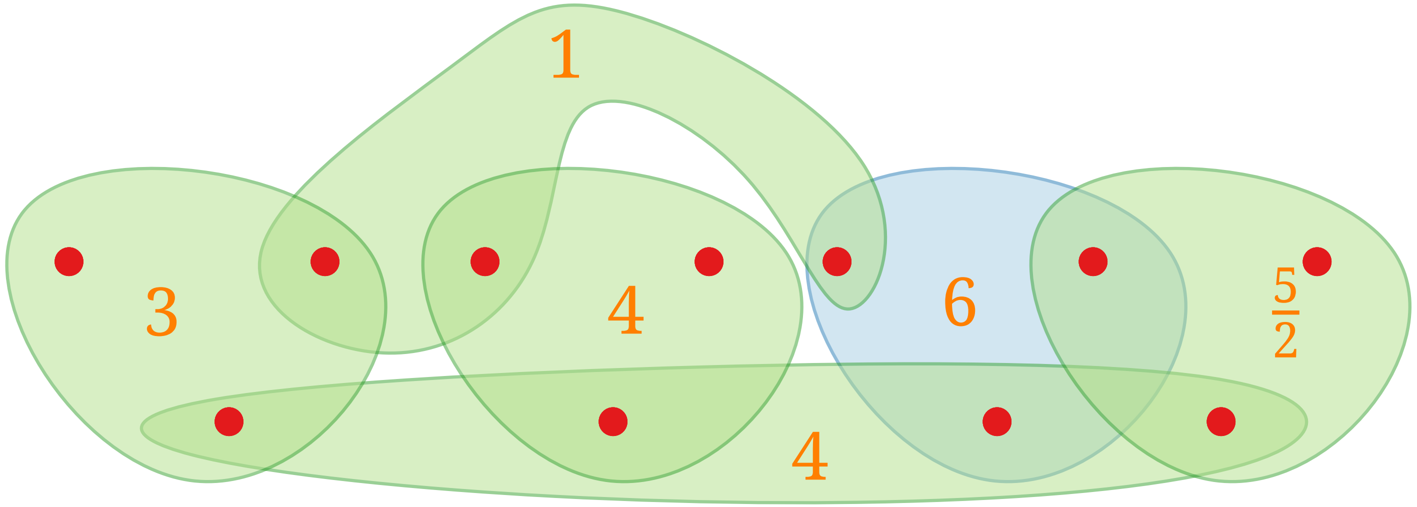
subject to

x_S $\quad$ $S \in \mathcal{S}$

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

SetCover as an ILP

minimize

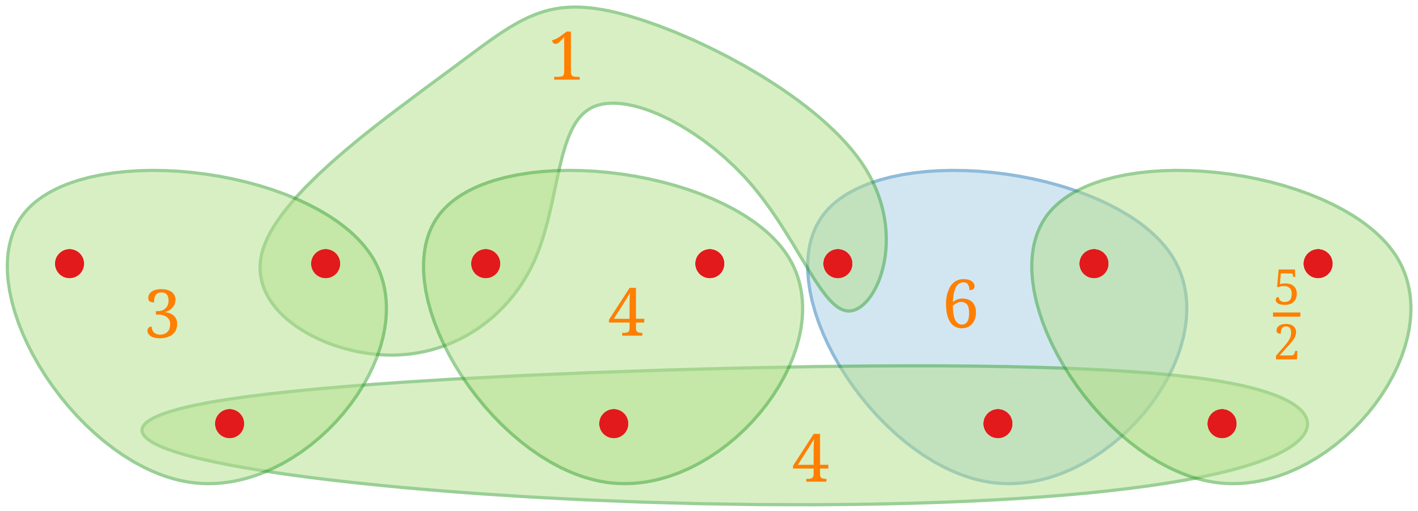
subject to

$$x_S \in \{0,1\} \quad S \in \mathcal{S}$$

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

SetCover as an ILP

$$\text{minimize} \quad \sum_{S \in \mathcal{S}} c_S x_S$$

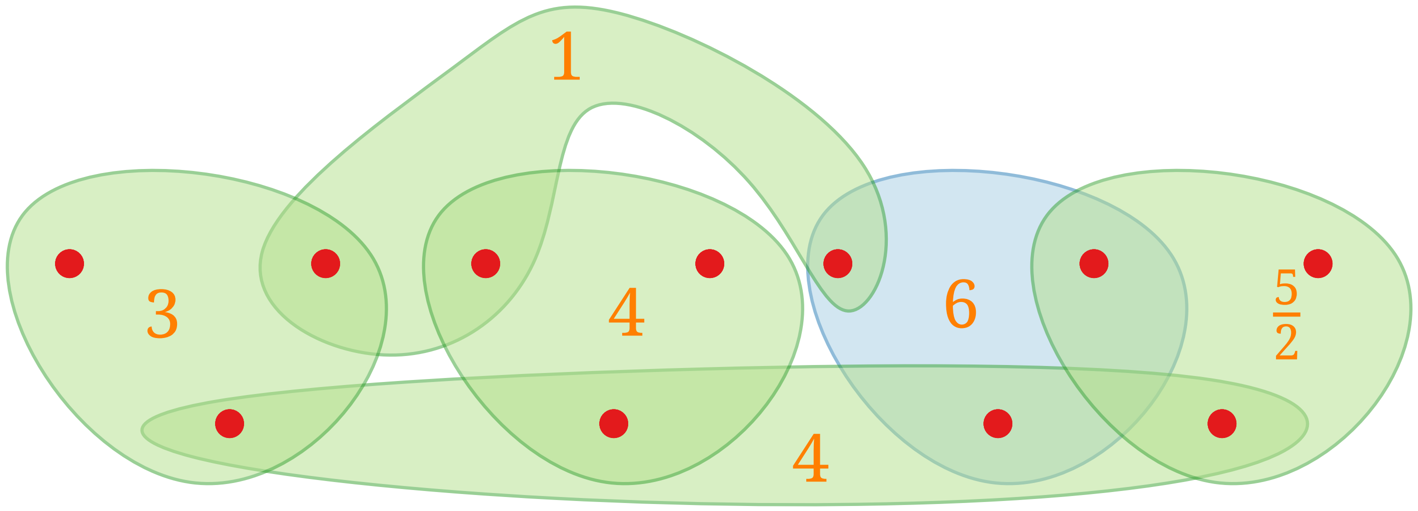
$$\text{subject to}$$

$$x_S \in \{0,1\} \quad S \in \mathcal{S}$$

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

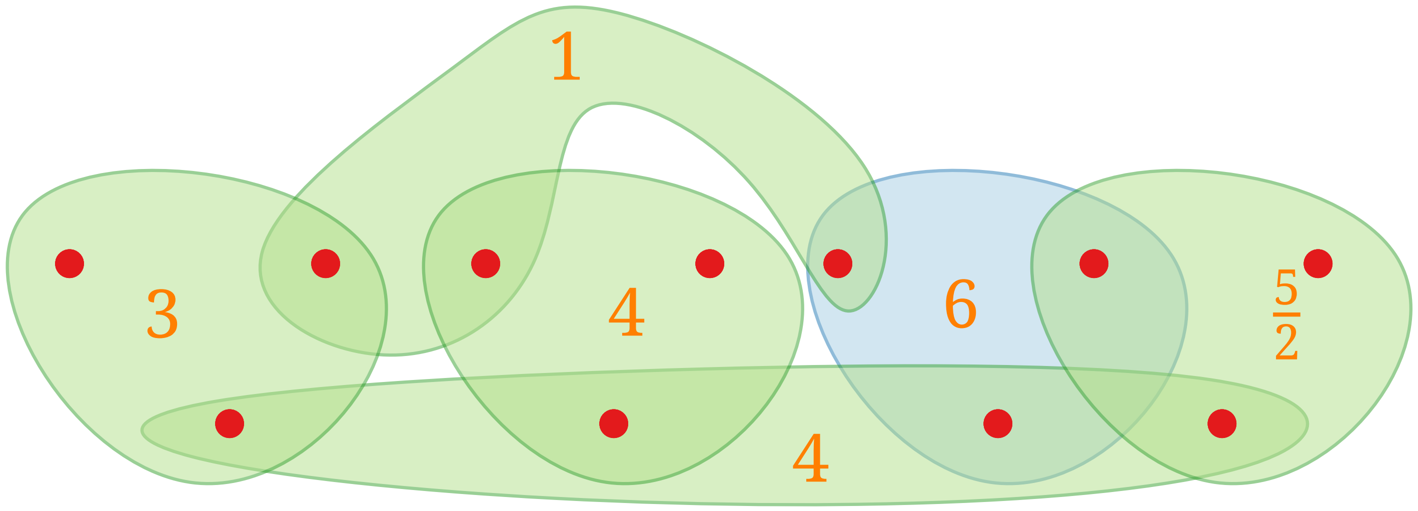
SetCover as an ILP

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to		$u \in U$
	$x_S \in \{0,1\}$	$S \in \mathcal{S}$

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

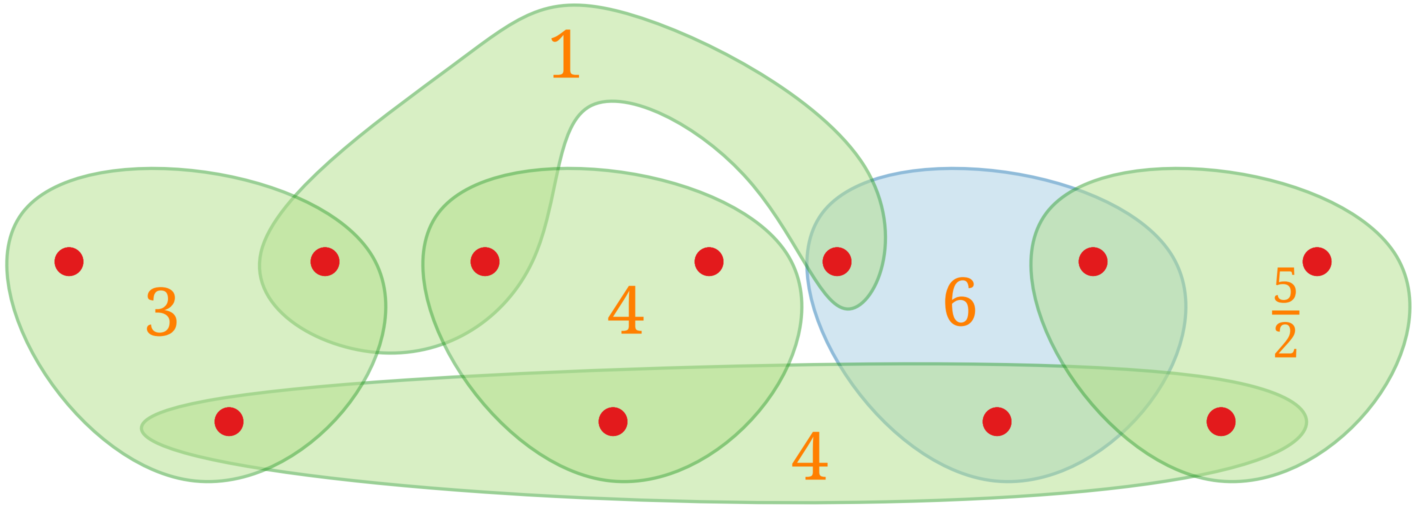
SETCOVER as an ILP

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \in \{0,1\} & S \in \mathcal{S} \end{array}$$

Ground set U

Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$

Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



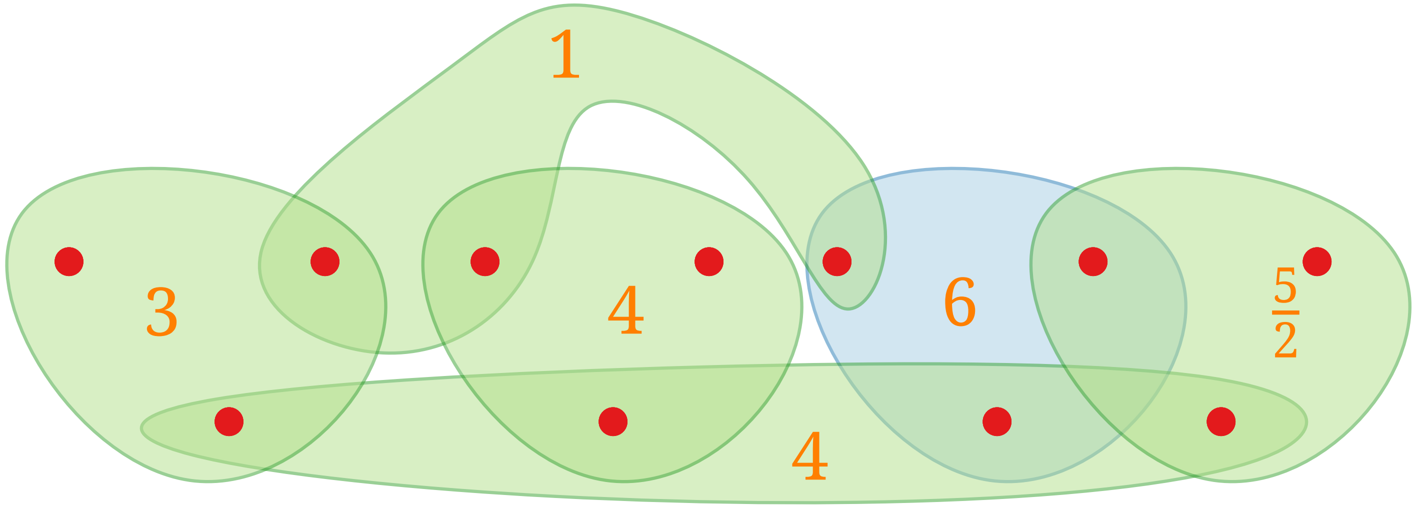
Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

SetCover as an ILP

$$\begin{array}{lll} \textbf{minimize} & \sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \in \{0, 1\} & S \in \mathcal{S} \end{array}$$

Relaxed LP: $0 \leq x_S (\leq 1)$ instead
but how does it help?

Ground set U
Family $\mathcal{S} \subseteq 2^U$ with $\bigcup \mathcal{S} = U$
Costs $c \colon \mathcal{S} \to \mathbb{Q}^+$



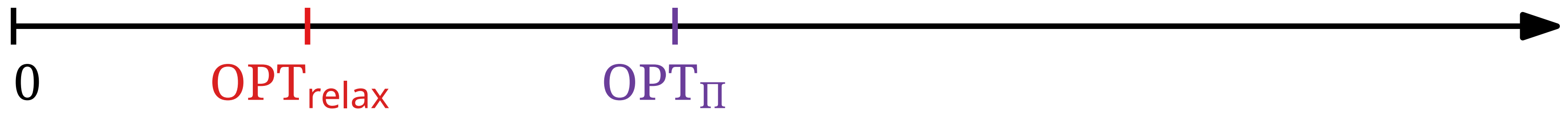
Find cover $\mathcal{S}' \subseteq \mathcal{S}$ of U with minimum cost.

LP Rounding

Technique I) LP-Rounding

Consider a minimization problem Π in ILP form.

Technique I) LP-Rounding



Consider a minimization problem Π in ILP form.

Compute a solution for the LP-relaxation.

Technique I) LP-Rounding

Consider a minimization problem Π in ILP form.

Compute a solution for the LP-relaxation.

Round to obtain an integer solution for Π.

Technique I) LP-Rounding

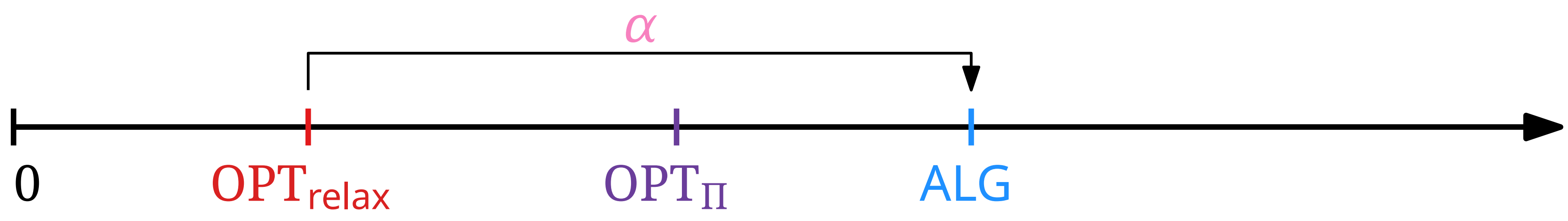
Consider a minimization problem Π in ILP form.

Compute a solution for the LP-relaxation.

Round to obtain an integer solution for Π.

Difficulty: Ensure the **feasiblity** of the solution.

Technique I) LP-Rounding



Consider a minimization problem Π in ILP form.

Compute a solution for the LP-relaxation.

Round to obtain an integer solution for Π.

Difficulty: Ensure the **feasiblity** of the solution.

Approximation factor: $\mathrm{ALG}/\mathrm{OPT}_{\Pi} \leq \mathrm{ALG}/\mathrm{OPT}_{\text{relax}}$.

SetCover – LP-Relaxation

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

SetCover – LP-Relaxation

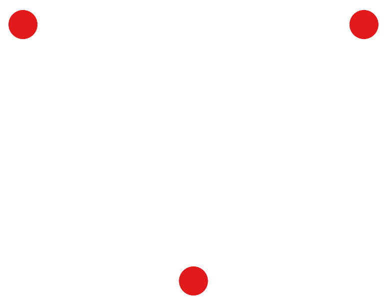
$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

Minimum Set Cover vs. relaxed optimum?

SetCover – LP-Relaxation

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
 & x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

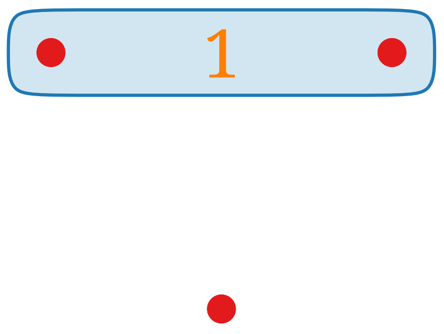
Minimum Set Cover vs. relaxed optimum?



SetCover – LP-Relaxation

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
& x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

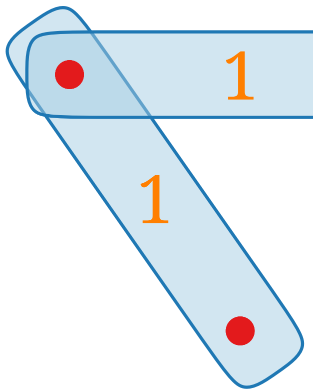
Minimum Set Cover vs. relaxed optimum?



SetCover – LP-Relaxation

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

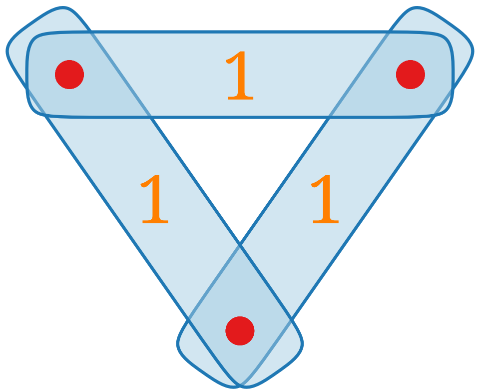
Minimum Set Cover vs. relaxed optimum?



SetCover – LP-Relaxation

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

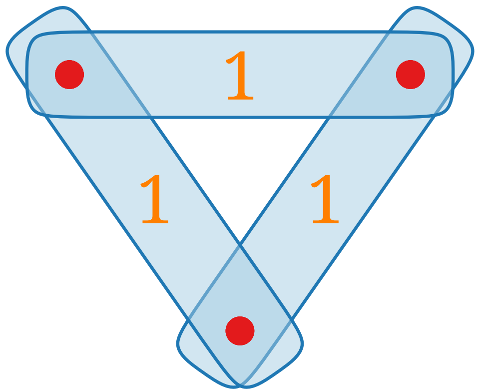
Minimum Set Cover vs. relaxed optimum?

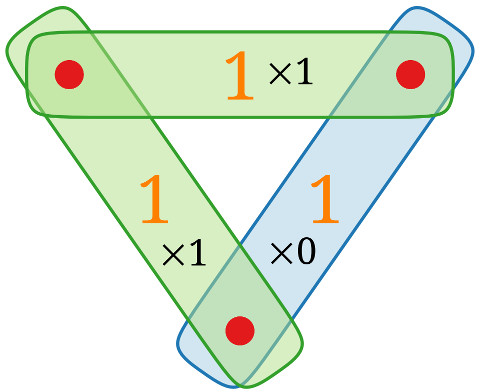


SetCover – LP-Relaxation

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
 & x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

Minimum Set Cover vs. relaxed optimum?



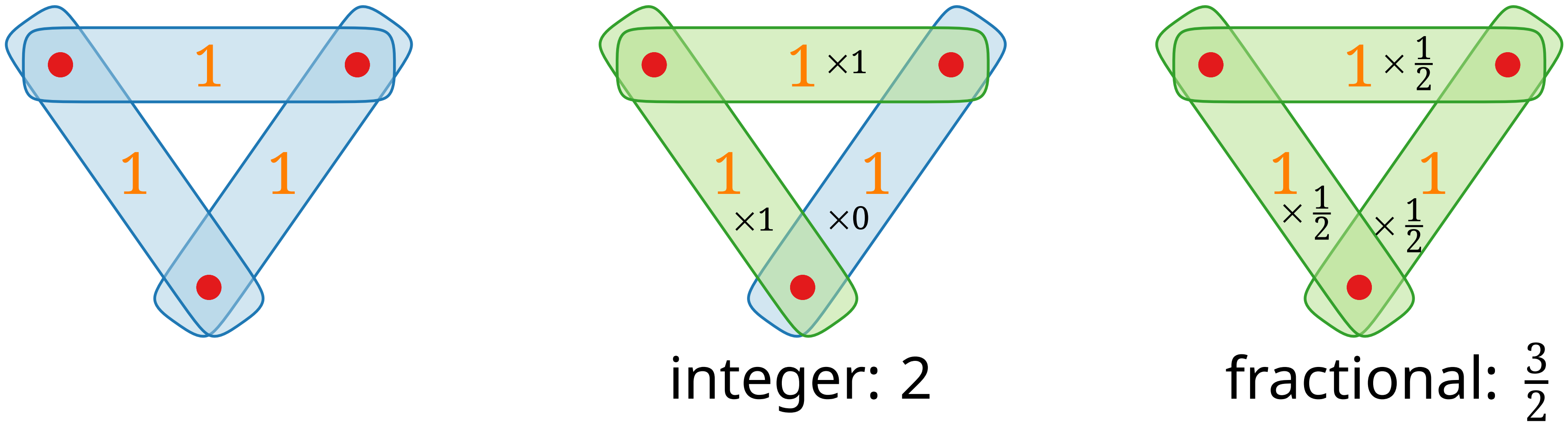


integer: 2

SetCover – LP-Relaxation

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

Minimum Set Cover vs. relaxed optimum?



LP-Rounding: First Attempt

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding-Attempt$(U, \mathcal{S}, c)$

LP-Rounding: First Attempt

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 && u \in U \\ & x_S \geq 0 && S \in \mathcal{S} \end{aligned}$$

LP-Rounding-Attempt($U, \mathcal{S}, c$)

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

LP-Rounding: First Attempt

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
 & x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

LP-Rounding: First Attempt

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.

LP-Rounding: First Attempt

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
 & x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

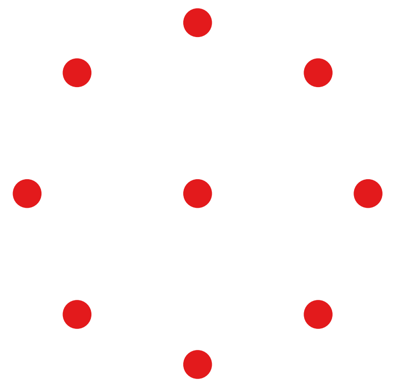
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

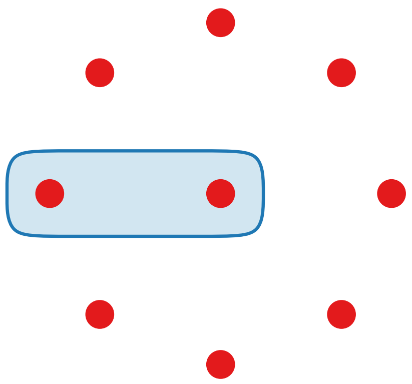
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

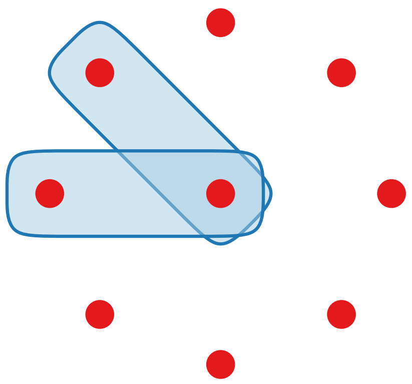
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 \qquad u \in U \\ & x_S \geq 0 \qquad\qquad S \in \mathcal{S} \end{aligned}$$

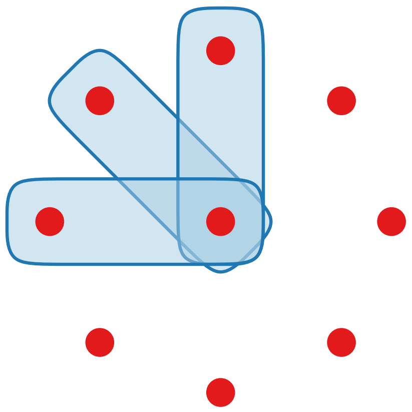
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 \qquad u \in U \\ & x_S \geq 0 \qquad\quad S \in \mathcal{S} \end{aligned}$$

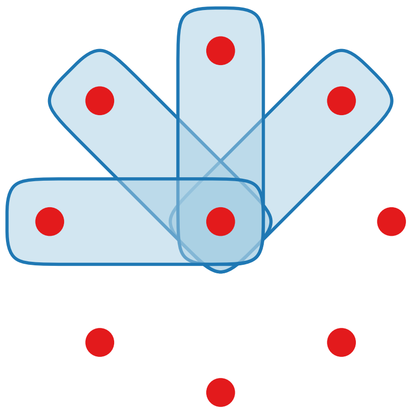
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
 & x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

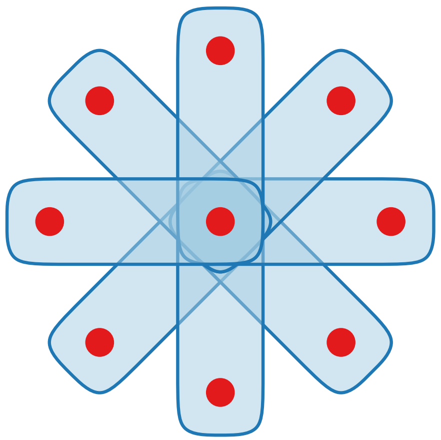
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$
\begin{array}{lll}
\textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\
& x_S \geq 0 & S \in \mathcal{S}
\end{array}
$$

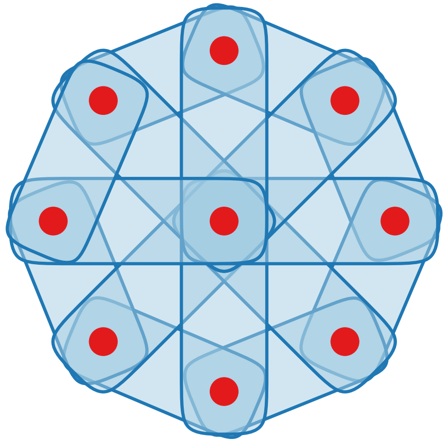
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$\begin{aligned}
\textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S & \\
\textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 & u \in U \\
& x_S \geq 0 & S \in \mathcal{S}
\end{aligned}$$

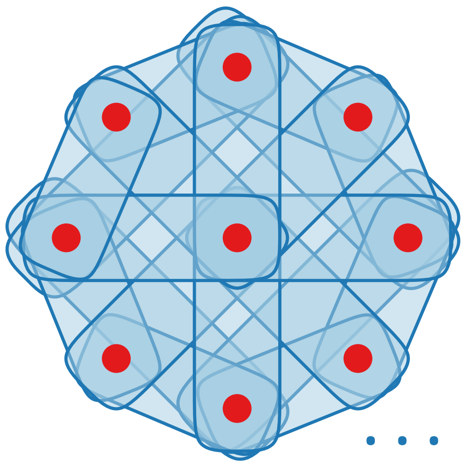
LP-Rounding-Attempt$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.



LP-Rounding: First Attempt

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding-Attempt$(U, \mathcal{S}, c)$

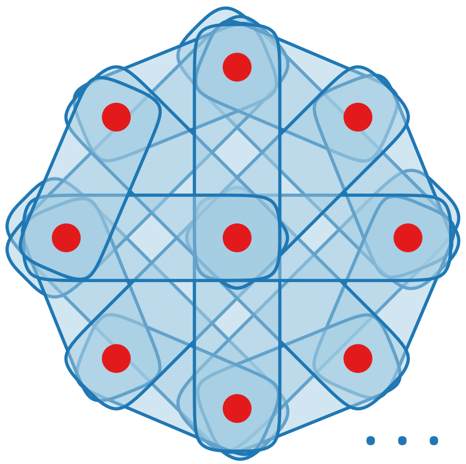
Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S > 0$ to 1.

– Generates a valid solution.

– Scaling factor arbitrarily large.

Use frequency f



LP-Rounding

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S \geq$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

LP-Rounding

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

LP-Rounding

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{aligned}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Why is this feasible?

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

LP-Rounding

$$\begin{array}{lll} \textbf{minimize} & \sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.
Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

Why is this feasible?

sum of $\leq f$ elements
$\rightarrow$ at least one $\geq 1/f$
by pigeonhole principle

LP-Rounding

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 && u \in U \\ & x_S \geq 0 && S \in \mathcal{S} \end{aligned}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.
Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

What is the approximation factor?

Why is this feasible?
sum of $\leq f$ elements
$\rightarrow$ at least one $\geq 1/f$
by pigeonhole principle

LP-Rounding

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 \qquad u \in U \\ & x_S \geq 0 \qquad\quad\; S \in \mathcal{S} \end{aligned}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

What is the approximation factor?

the rounded solution is bounded by f times the fractional solution

Why is this feasible?

sum of $\leq f$ elements

$\rightarrow$ at least one $\geq 1/f$

by pigeonhole principle

LP-Rounding

$$\begin{array}{lll} \textbf{minimize} & \sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.
Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

Theorem. LP-Rounding gives a factor-f approximation algorithm for SetCover.

LP-Rounding

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

LP-Rounding$(U, \mathcal{S}, c)$

Compute optimal solution x for LP-relaxation.

Round each x_S with $x_S \geq 1/f$ to 1; remaining to 0.

Let f be the frequency of (i.e., the number of sets containing) the most frequent element.

Theorem. LP-Rounding gives a factor-f approximation algorithm for SetCover.

Note: Factor-$O(\log n)$ can be achieved with randomized rounding (next lecture)

Reminder: LP Duality

LP-Duality

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

LP-Duality

primal

$$\begin{array}{llcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

dual

???

LP-Duality

primal

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

dual

$$\begin{array}{lrcl} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & A^\top y & \leq & c \\ & y & \geq & 0 \end{array}$$

LP-Duality

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

minimize	$c^\intercal x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\intercal y$			*dual*
subject to	$A^\intercal y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

LP-Duality

minimize	$c^\intercal x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\intercal y$			*dual*
subject to	$A^\intercal y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} c_j x_j \geq \qquad\qquad \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \qquad \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

				primal
minimize	$c^\top x$			
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

				dual
maximize	$b^\top y$			
subject to	$\boxed{A^\top y \leq c}$			
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \qquad\qquad \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \qquad\qquad \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

minimize	$c^\top x$		*primal*
subject to	Ax	$\geq$	b
	$\boxed{x \geq 0}$		

maximize	$b^\top y$		*dual*
subject to	$\boxed{A^\top y \leq c}$		
	y	$\geq$	0

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i \right)} x_j = \qquad \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i \right)} x_j = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i x_j \right) \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

primal

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	$\boxed{x \geq 0}$		

dual

maximize	$b^\top y$		
subject to	$\boxed{A^\top y \leq c}$		
	y	$\geq$	0

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i\right)} x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} y_i x_j\right) \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

primal

$$\begin{array}{lrcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & \boxed{x \geq 0} & & \end{array}$$

dual

$$\begin{array}{lrcl} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & \boxed{A^\top y \leq c} & & \\ & y & \geq & 0 \end{array}$$

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i .$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i \right)} x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \geq \sum_{i=1}^{m} b_i y_i .$$

LP-Duality

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	$\boxed{x \geq 0}$			

maximize	$b^\top y$			*dual*
subject to	$\boxed{A^\top y \leq c}$			
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i\right)} x_j = \sum_{i=1}^{m} \left(\boxed{\sum_{j=1}^{n} a_{ij} x_j}\right) y_i \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

		primal
minimize	$c^\top x$	
subject to	$Ax \geq b$	
	$x \geq 0$	

		dual
maximize	$b^\top y$	
subject to	$A^\top y \leq c$	
	$y \geq 0$	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i\right)} x_j = \sum_{i=1}^{m} \left(\boxed{\sum_{j=1}^{n} a_{ij} x_j}\right) y_i \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

minimize	$c^\top x$	*primal*
subject to	$\boxed{Ax \geq b}$	
	$\boxed{x \geq 0}$	

maximize	$b^\top y$	*dual*
subject to	$\boxed{A^\top y \leq c}$	
	$\boxed{y \geq 0}$	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i \right)} x_j = \sum_{i=1}^{m} \left(\boxed{\sum_{j=1}^{n} a_{ij} x_j} \right) y_i \geq \sum_{i=1}^{m} b_i y_i \,.$$

LP-Duality

		primal
minimize	$c^\top x$	
subject to	$\boxed{Ax \geq b}$	
	$\boxed{x \geq 0}$	

		dual
maximize	$b^\top y$	
subject to	$\boxed{A^\top y \leq c}$	
	$\boxed{y \geq 0}$	

Theorem. (weak duality) If $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Proof.

$$\sum_{j=1}^{n} \boxed{c_j} x_j \overset{\geq}{\geq} \sum_{j=1}^{n} \boxed{\left(\sum_{i=1}^{m} a_{ij} y_i\right)} x_j = \sum_{i=1}^{m} \left(\boxed{\sum_{j=1}^{n} a_{ij} x_j}\right) y_i \overset{\geq}{\geq} \sum_{i=1}^{m} \boxed{b_i} y_i \,.$$

LP-Duality

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. (weak duality) If $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ are valid solutions for the primal and dual program, resp., then

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{i=1}^{m} b_i y_i \,.$$

Theorem. (strong duality) primal has a finite optimum $\Leftrightarrow$ dual has a finite optimum. If $x^* = (x_1^*, \ldots, x_n^*)$ and $y^* = (y_1^*, \ldots, y_m^*)$ are optimal solutions for the primal and dual program, respectively, then

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^* \,.$$

Complementary Slackness

primal

$$\begin{array}{llcl} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

dual

$$\begin{array}{llcl} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & A^\top y & \leq & c \\ & y & \geq & 0 \end{array}$$

Complementary Slackness

primal

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	x	$\geq$	0

dual

maximize	$b^\top y$		
subject to	$A^\top y$	$\leq$	c
	y	$\geq$	0

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then

Complementary Slackness

primal

$$\begin{array}{llll} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

dual

$$\begin{array}{llll} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & A^\top y & \leq & c \\ & y & \geq & 0 \end{array}$$

Theorem. Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Complementary Slackness

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^{m} a_{ij} y_i = c_j$

Complementary Slackness

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^{m} a_{ij} y_i = c_j$

Dual CS:
For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$

Complementary Slackness

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^{m} a_{ij} y_i = c_j$

Dual CS:
For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$

Proof.

Complementary Slackness

minimize	$c^\top x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\top y$			*dual*
subject to	$A^\top y$	$\leq$	c	
	y	$\geq$	0	

Theorem. Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \dots, n$: $\quad x_j = 0 \quad$ or $\quad \sum_{i=1}^{m} a_{ij} y_i = c_j$

Dual CS:
For each $i = 1, \dots, m$: $\quad y_i = 0 \quad$ or $\quad \sum_{j=1}^{n} a_{ij} x_j = b_i$

Proof. Follows from LP-duality:

Complementary Slackness

minimize	$c^\intercal x$			*primal*
subject to	Ax	$\geq$	b	
	x	$\geq$	0	

maximize	$b^\intercal y$			*dual*
subject to	$A^\intercal y$	$\leq$	c	
	y	$\geq$	0	

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^{m} a_{ij} y_i = c_j$

Dual CS:
For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$

Proof. Follows from LP-duality:

$$\sum_{j=1}^{n} c_j x_j \geq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \geq \sum_{i=1}^{m} b_i y_i \,.$$

Complementary Slackness

primal

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	$\boxed{x}$	$\boxed{\geq}$	$\boxed{0}$

dual

maximize	$b^\top y$		
subject to	$\boxed{A^\top y}$	$\boxed{\leq}$	$\boxed{c}$
	y	$\geq$	0

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^m a_{ij} y_i = c_j$

Dual CS:
For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^n a_{ij} x_j = b_i$

Proof. Follows from LP-duality:

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \sum_{j=1}^{n} \left(\boxed{\sum_{i=1}^{m} a_{ij} y_i} \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \geq \sum_{i=1}^{m} b_i y_i \,.$$

Complementary Slackness

		primal
minimize	$c^\top x$	
subject to	$\boxed{Ax \geq b}$	
	$\boxed{x \geq 0}$	

		dual
maximize	$b^\top y$	
subject to	$\boxed{A^\top y \leq c}$	
	$\boxed{y \geq 0}$	

Theorem. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ be valid solutions for the primal and dual program, respectively. Then x and y are optimal if and only if the following conditions are met:

Primal CS:
For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^m a_{ij} y_i = c_j$

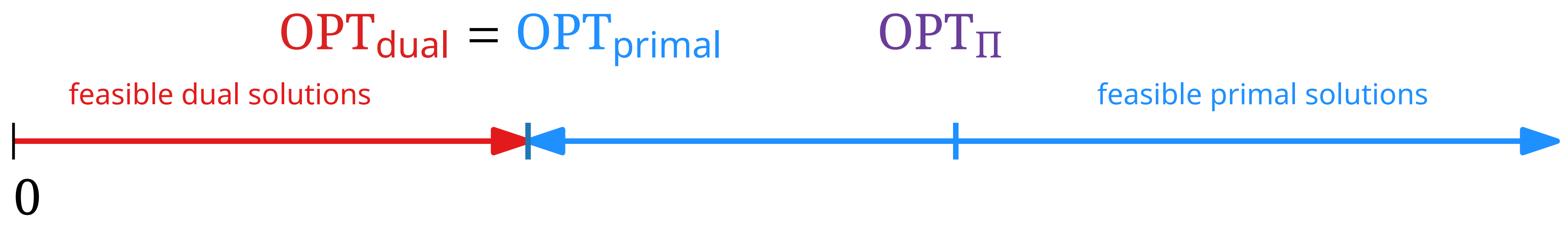
Dual CS:
For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^n a_{ij} x_j = b_i$

Proof. Follows from LP-duality:

$$\sum_{j=1}^{n} \boxed{c_j} x_j \geq \sum_{j=1}^{n} \left(\boxed{\sum_{i=1}^{m} a_{ij} y_i} \right) x_j = \sum_{i=1}^{m} \left(\boxed{\sum_{j=1}^{n} a_{ij} x_j} \right) y_i \geq \sum_{i=1}^{m} \boxed{b_i} y_i \,.$$

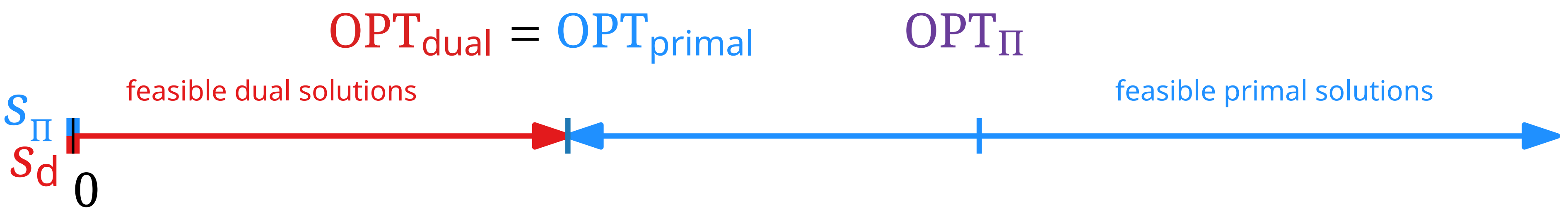
primal-dual method

Technique II) Primal–Dual Method



Consider a minimization problem Π in ILP form.

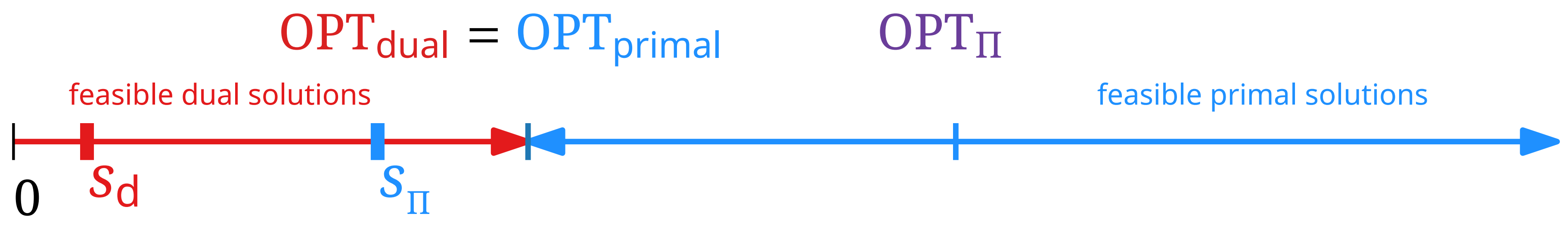
Technique II) Primal–Dual Method



Consider a minimization problem Π in ILP form.

- Start with (trivial) feasible dual solution and infeasible primal solution (e.g., all variables = 0).

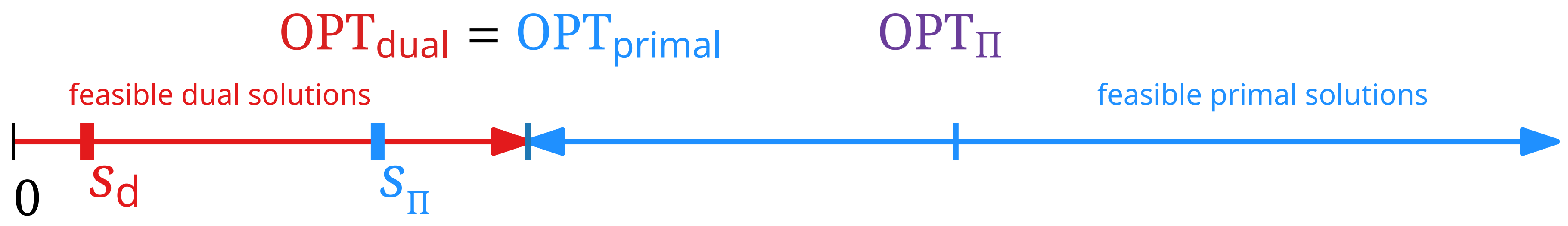
Technique II) Primal–Dual Method



Consider a minimization problem Π in ILP form.

- Start with (trivial) feasible dual solution and infeasible primal solution (e.g., all variables $= 0$).

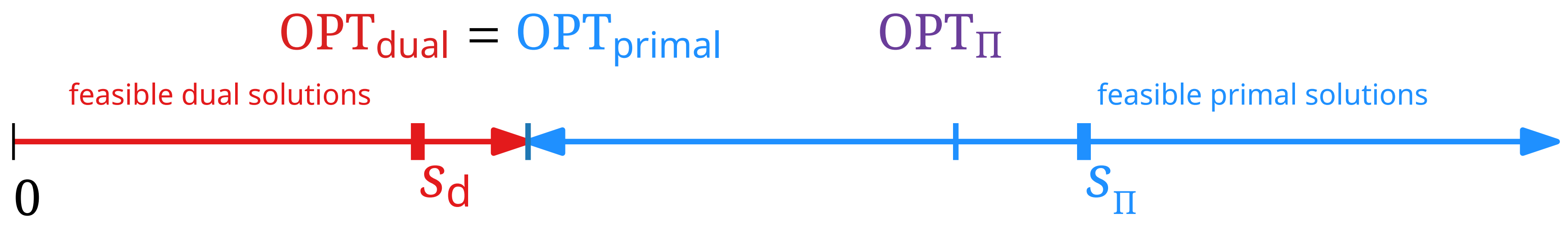
Technique II) Primal–Dual Method



Consider a minimization problem Π in ILP form.

- Start with (trivial) feasible dual solution and infeasible primal solution (e.g., all variables = 0).
- Compute dual solution s_{d} and integral primal solution s_{Π} for Π iteratively: Increase s_{d} according to CS and make s_{Π} “more feasible”.

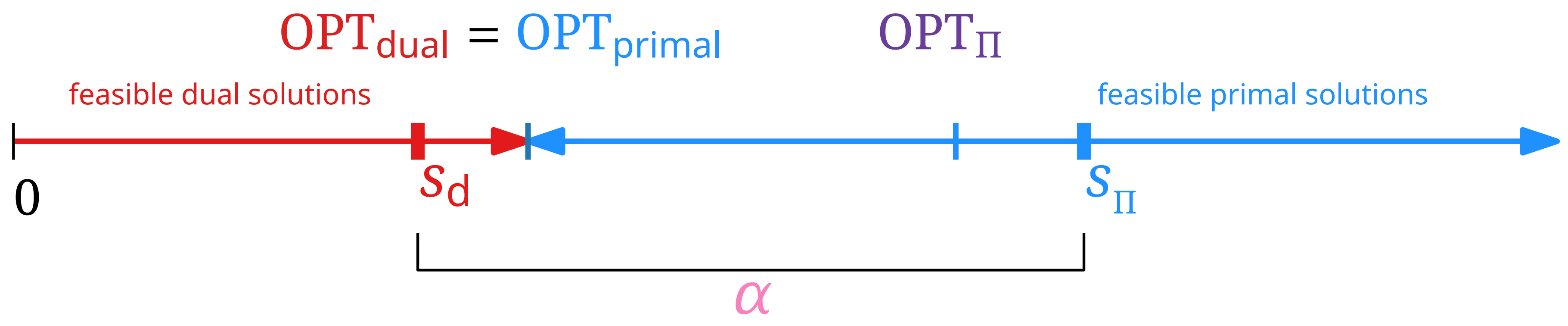
Technique II) Primal–Dual Method



Consider a minimization problem Π in ILP form.

- Start with (trivial) feasible dual solution and infeasible primal solution (e.g., all variables = 0).
- Compute dual solution s_{d} and integral primal solution s_{Π} for Π iteratively: Increase s_{d} according to CS and make s_{Π} "more feasible".

Technique II) Primal–Dual Method

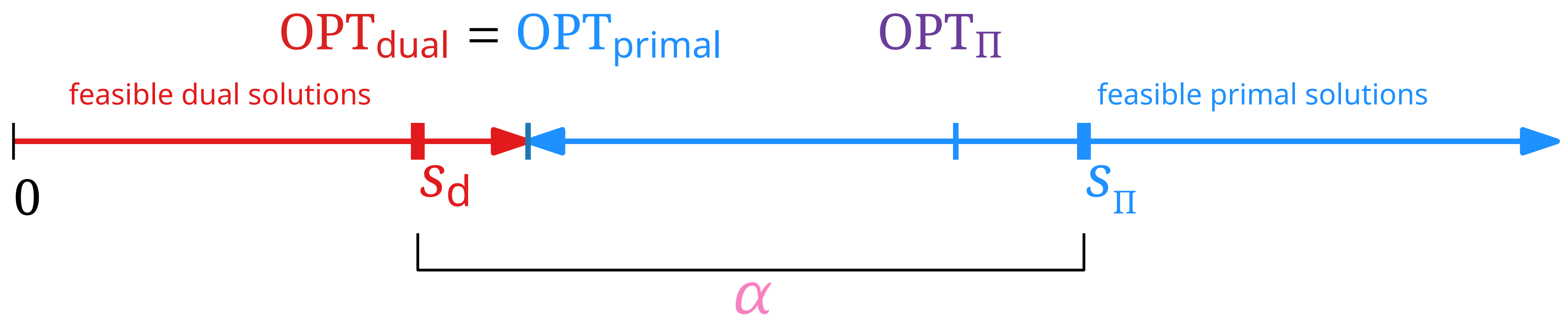


Consider a minimization problem Π in ILP form.

- Start with (trivial) feasible dual solution and infeasible primal solution (e.g., all variables $= 0$).
- Compute dual solution s_{d} and integral primal solution s_{Π} for Π iteratively:
  Increase s_{d} according to CS and make s_{Π} "more feasible".

Approximation factor $\leq \mathrm{obj}(s_{\Pi}) / \mathrm{obj}(s_{\text{d}})$

Technique II) Primal–Dual Method



Consider a minimization problem Π in ILP form.

- Start with (trivial) feasible dual solution and infeasible primal solution (e.g., all variables = 0).
- Compute dual solution s_{d} and integral primal solution s_{Π} for Π iteratively: Increase s_{d} according to CS and make s_{Π} "more feasible".

Approximation factor $\leq \mathrm{obj}(s_{\Pi})/\mathrm{obj}(s_{\mathrm{d}})$

Advantage: Don't need LP-"machinery"; possibly faster, more flexible.

SetCover – Dual LP

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

?

SetCover – Dual LP

minimize $\sum_{S \in \mathcal{S}} c_S x_S$

subject to $\sum_{S \ni u} x_S \geq 1 \qquad u \in U$

$x_S \geq 0 \qquad S \in \mathcal{S}$

maximize

subject to

SetCover – Dual LP

minimize $\displaystyle\sum_{S \in \mathcal{S}} c_S x_S$

subject to $\displaystyle\sum_{S \ni u} x_S \geq 1 \qquad u \in U$

$x_S \geq 0 \qquad S \in \mathcal{S}$

maximize

subject to

$y_u \geq 0 \qquad u \in U$

SetCover – Dual LP

$$\text{minimize} \quad \sum_{S \in \mathcal{S}} c_S x_S$$

$$\text{subject to} \quad \sum_{S \ni u} x_S \geq 1 \qquad u \in U$$

$$x_S \geq 0 \qquad S \in \mathcal{S}$$

$$\text{maximize} \quad \sum_{u \in U} y_u$$

subject to

$$y_u \geq 0 \qquad u \in U$$

SetCover – Dual LP

$$\begin{aligned} \textbf{minimize} \quad & \sum_{S \in \mathcal{S}} c_S x_S \\ \textbf{subject to} \quad & \sum_{S \ni u} x_S \geq 1 && u \in U \\ & x_S \geq 0 && S \in \mathcal{S} \end{aligned}$$

$$\begin{aligned} \textbf{maximize} \quad & \sum_{u \in U} y_u \\ \textbf{subject to} \quad & \sum_{u \in S} y_u \leq c_S && S \in \mathcal{S} \\ & y_u \geq 0 && u \in U \end{aligned}$$

SetCover – Dual LP

Covering LP

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \text{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

Packing LP

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \text{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

SetCover – Dual LP

Covering LP

$$\text{minimize} \quad \sum_{S \in \mathcal{S}} c_S x_S$$

$$\text{subject to} \quad \sum_{S \ni u} x_S \geq 1 \qquad u \in U$$

$$x_S \geq 0 \qquad S \in \mathcal{S}$$

Packing LP

$$\text{maximize} \quad \sum_{u \in U} y_u$$

$$\text{subject to} \quad \sum_{u \in S} y_u \leq c_S \qquad S \in \mathcal{S}$$ “no overpacking”

$$y_u \geq 0 \qquad u \in U$$

Relaxing Complementary Slackness

$$\begin{array}{llll} \textbf{minimize} & c^\top x & & \\ \textbf{subject to} & Ax & \geq & b \\ & x & \geq & 0 \end{array}$$

$$\begin{array}{llll} \textbf{maximize} & b^\top y & & \\ \textbf{subject to} & A^\top y & \leq & c \\ & y & \geq & 0 \end{array}$$

Primal CS:

For each $j = 1, \ldots, n$: $x_j = 0$ or $\sum_{i=1}^m a_{ij} y_i = c_j$

Dual CS:

For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^n a_{ij} x_j = b_i$

$$\Leftrightarrow \sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i$$

Relaxing Complementary Slackness

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	x	$\geq$	0

maximize	$b^\top y$		
subject to	$A^\top y$	$\leq$	c
	y	$\geq$	0

~~**Primal CS**~~: Relaxed Primal CS

For each $j = 1, \ldots, n$: $x_j = 0$ or $\cancel{\sum_{i=1}^m a_{ij} y_i = c_j}$

$$c_j/\alpha \leq \sum_{i=1}^m a_{ij} y_i \leq c_j$$

Dual CS:

For each $i = 1, \ldots, m$: $y_i = 0$ or $\sum_{j=1}^n a_{ij} x_j = b_i$

$$\Leftrightarrow \sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i$$

Relaxing Complementary Slackness

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	x	$\geq$	0

maximize	$b^\top y$		
subject to	$A^\top y$	$\leq$	c
	y	$\geq$	0

~~**Primal CS**~~: Relaxed Primal CS

For each $j = 1, \ldots, n$: $x_j = 0$ or $\cancel{\sum_{i=1}^{m} a_{ij} y_i = c_j}$

$$c_j/\alpha \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$$

~~**Dual CS**~~: Relaxed Dual CS

For each $i = 1, \ldots, m$: $y_i = 0$ or $\cancel{\sum_{j=1}^{n} a_{ij} x_j = b_i}$

$$b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta \cdot b_i$$

$$\Leftrightarrow \sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i$$

Relaxing Complementary Slackness

minimize	$c^\top x$		
subject to	Ax	$\geq$	b
	x	$\geq$	0

maximize	$b^\top y$		
subject to	$A^\top y$	$\leq$	c
	y	$\geq$	0

~~Primal CS~~: Relaxed Primal CS

For each $j = 1, \ldots, n$: $x_j = 0$ or $\cancel{\sum_{i=1}^m a_{ij} y_i = c_j}$

$$c_j/\alpha \leq \sum_{i=1}^m a_{ij} y_i \leq c_j$$

~~Dual CS~~: Relaxed Dual CS

For each $i = 1, \ldots, m$: $y_i = 0$ or $\cancel{\sum_{j=1}^n a_{ij} x_j = b_i}$

$$b_i \leq \sum_{j=1}^n a_{ij} x_j \leq \beta \cdot b_i$$

$$\cancel{\Leftrightarrow \sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i} \quad \Rightarrow \sum_{j=1}^n c_j x_j \leq \alpha\beta \sum_{i=1}^m b_i y_i \leq \alpha\beta \cdot \mathrm{OPT}_{\mathrm{LP}}$$

Primal–Dual Method

Start with a feasible dual and infeasible primal solution (often trivial).

Primal–Dual Method

Start with a feasible dual and infeasible primal solution (often trivial).

“Improve” the feasibility of the primal solution...

Primal–Dual Method

Start with a feasible dual and infeasible primal solution (often trivial).

“Improve” the feasibility of the primal solution...

. . . and simultaneously the objective value of the dual solution.

Primal–Dual Method

Start with a feasible dual and infeasible primal solution (often trivial).

“Improve” the feasibility of the primal solution...

. . . and simultaneously the objective value of the dual solution.

Do so until the relaxed CS conditions are met.

Primal–Dual Method

Start with a feasible dual and infeasible primal solution (often trivial).

“Improve” the feasibility of the primal solution...

. . . and simultaneously the objective value of the dual solution.

Do so until the relaxed CS conditions are met.

Maintain that the primal solution is integer-valued.

Primal–Dual Method

Start with a feasible dual and infeasible primal solution (often trivial).

"Improve" the feasibility of the primal solution...

. . . and simultaneously the objective value of the dual solution.

Do so until the relaxed CS conditions are met.

Maintain that the primal solution is integer-valued.

The feasibility of the primal solution and the relaxed CS conditions provide an approximation ratio.

Relaxed CS for SetCover

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

Relaxed CS for SetCover

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

(Unrelaxed) primal CS:

Relaxed CS for SetCover

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to	$\sum_{S \ni u} x_S \geq 1$	$u \in U$
	$x_S \geq 0$	$S \in \mathcal{S}$

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow$

Relaxed CS for SetCover

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to	$\sum_{S \ni u} x_S \geq 1$	$u \in U$
	$x_S \geq 0$	$S \in \mathcal{S}$

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

Relaxed CS for SETCOVER

minimize $\sum_{S \in \mathcal{S}} c_S x_S$	maximize $\sum_{u \in U} y_u$
subject to $\sum_{S \ni u} x_S \geq 1 \quad u \in U$	subject to $\sum_{u \in S} y_u \leq c_S \quad S \in \mathcal{S}$
$x_S \geq 0 \quad S \in \mathcal{S}$	$y_u \geq 0 \quad u \in U$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

Relaxed CS for SetCover

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to	$\sum_{S \ni u} x_S \geq 1$	$u \in U$
	$x_S \geq 0$	$S \in \mathcal{S}$

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

only chooses critical sets

Relaxed CS for SETCOVER

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

only chooses critical sets

Relaxed dual CS:

Relaxed CS for SetCover

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to	$\sum_{S \ni u} x_S \geq 1$	$u \in U$
	$x_S \geq 0$	$S \in \mathcal{S}$

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

only chooses critical sets

Relaxed dual CS: $y_u \neq 0 \Rightarrow$

Relaxed CS for SetCover

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to	$\sum_{S \ni u} x_S \geq 1$	$u \in U$
	$x_S \geq 0$	$S \in \mathcal{S}$

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

only chooses critical sets

Relaxed dual CS: $y_u \neq 0 \Rightarrow 1 \leq \sum_{S \ni u} x_S \leq \overset{?}{\beta} \cdot 1$

Relaxed CS for SetCover

minimize	$\sum_{S \in \mathcal{S}} c_S x_S$	
subject to	$\sum_{S \ni u} x_S \geq 1$	$u \in U$
	$x_S \geq 0$	$S \in \mathcal{S}$

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

only chooses critical sets

Relaxed dual CS: $y_u \neq 0 \Rightarrow 1 \leq \sum_{S \ni u} x_S \leq f \cdot 1$

Relaxed CS for SetCover

$$\begin{array}{lll} \textbf{minimize} & \displaystyle\sum_{S \in \mathcal{S}} c_S x_S & \\ \textbf{subject to} & \displaystyle\sum_{S \ni u} x_S \geq 1 & u \in U \\ & x_S \geq 0 & S \in \mathcal{S} \end{array}$$

$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

critical set

(Unrelaxed) primal CS: $x_S \neq 0 \Rightarrow \sum_{u \in S} y_u = c_S$

only chooses critical sets

trivial for binary x

Relaxed dual CS: $y_u \neq 0 \Rightarrow 1 \leq \displaystyle\sum_{S \ni u} x_S \leq f \cdot 1$

Primal–Dual Schema for SETCOVER

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0,\ y \leftarrow 0$

repeat

until all elements are covered.

return x

Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0, y \leftarrow 0$

repeat

Select an uncovered element u.

until all elements are covered.

return x

Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$

repeat

Select an uncovered element u.

Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

until all elements are covered.

return x

Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$

repeat

Select an uncovered element u.

Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

until all elements are covered.

return x

Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$

repeat

  Select an uncovered element u.

  Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

  Select all critical sets and update x.

  Mark all elements in these sets as covered.

until all elements are covered.

return x

Primal–Dual Schema for SETCOVER

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0, y \leftarrow 0$

repeat

Select an uncovered element u.

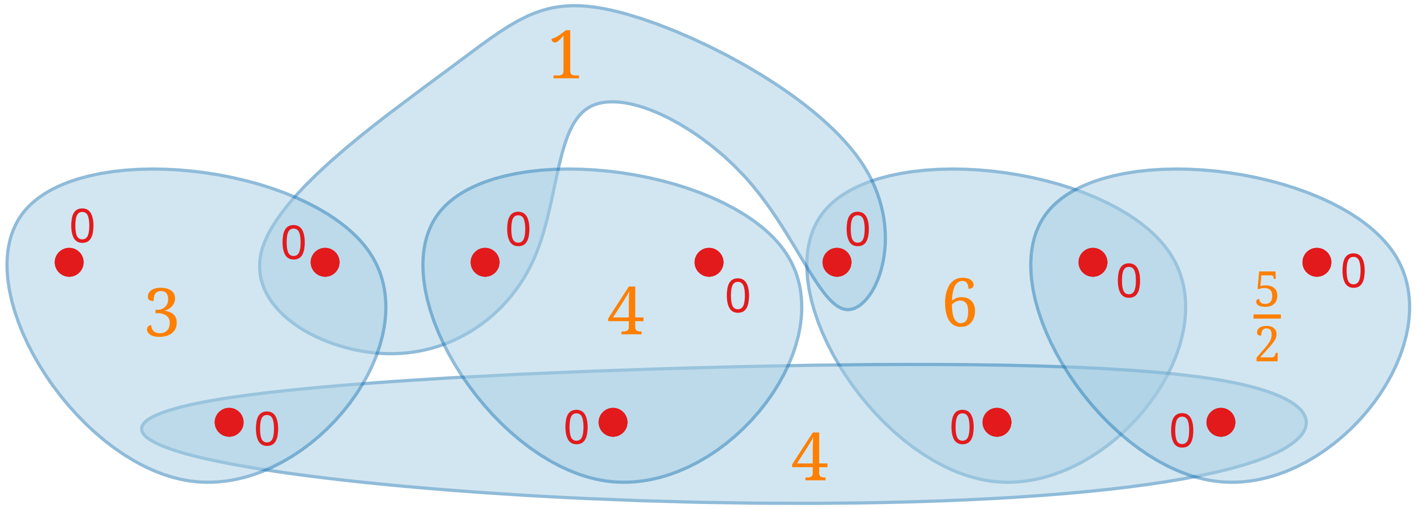
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0$, $y \leftarrow 0$

repeat

Select an uncovered element u.

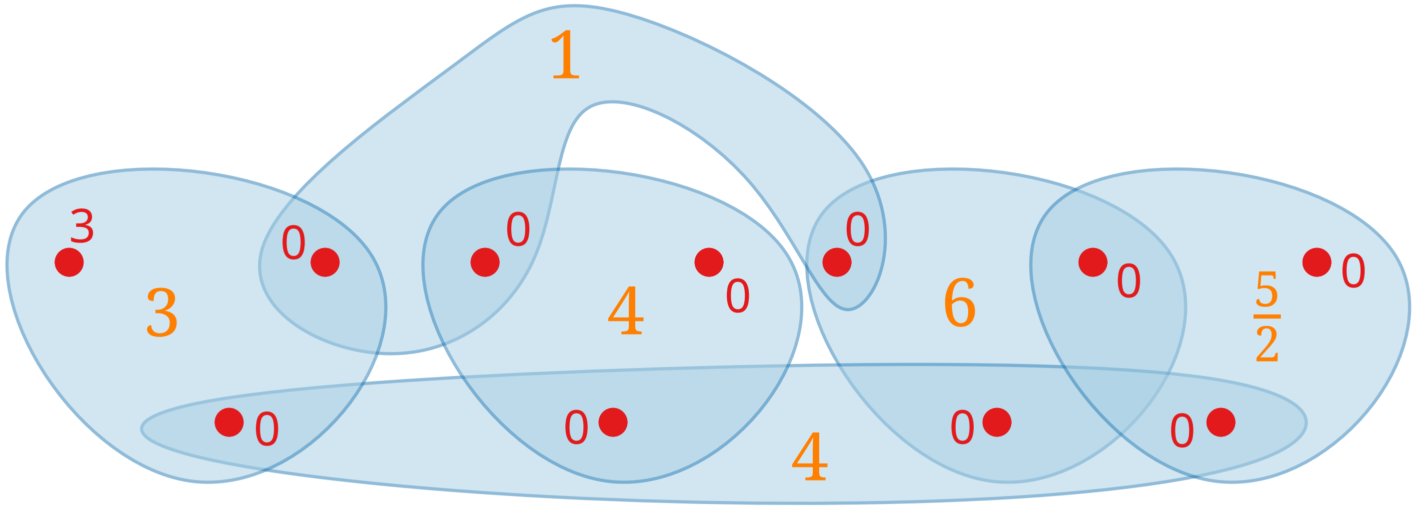
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

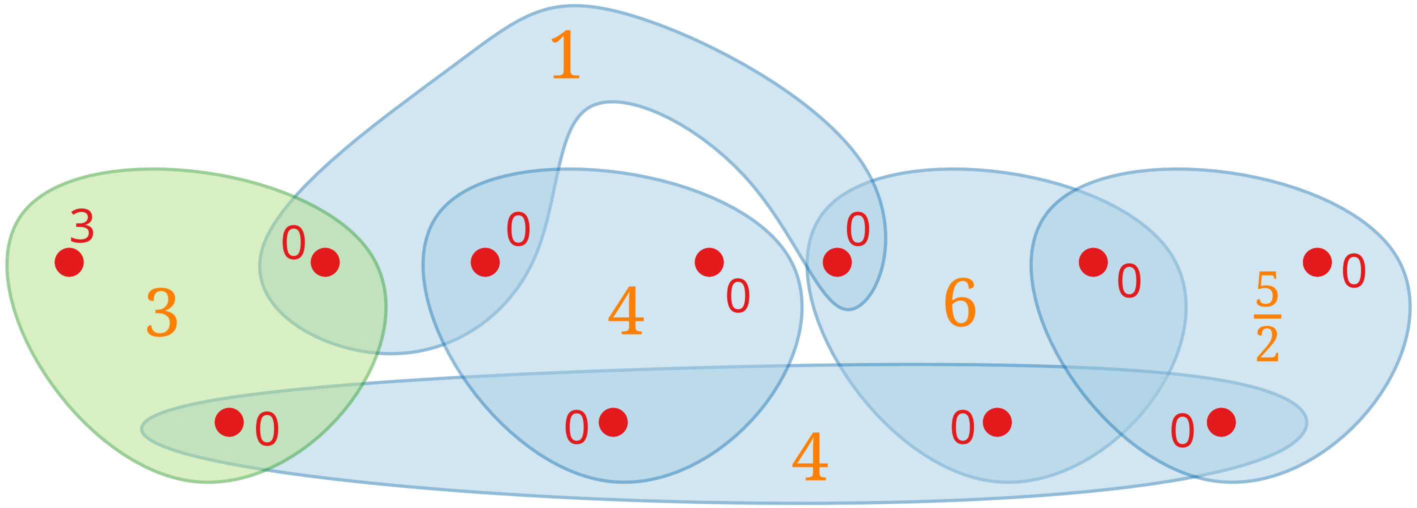
return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0, y \leftarrow 0$
repeat
  Select an uncovered element u.
  Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).
  Select all critical sets and update x.
  Mark all elements in these sets as covered.
until all elements are covered.
return x



Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$

repeat

Select an uncovered element u.

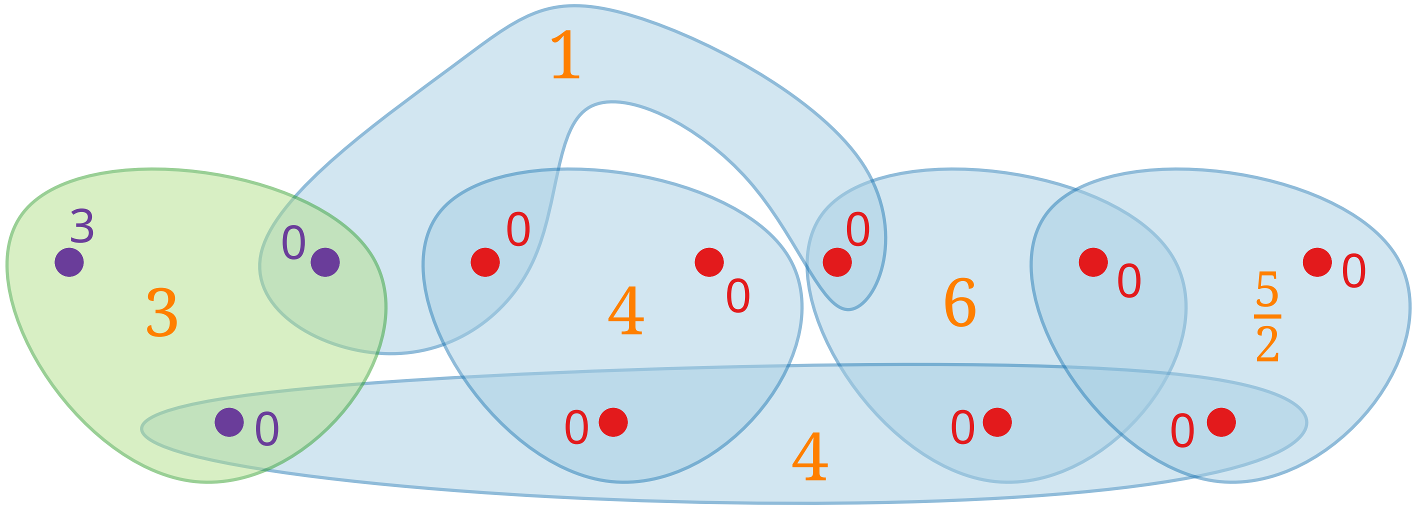
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0, y \leftarrow 0$

repeat

Select an uncovered element u.

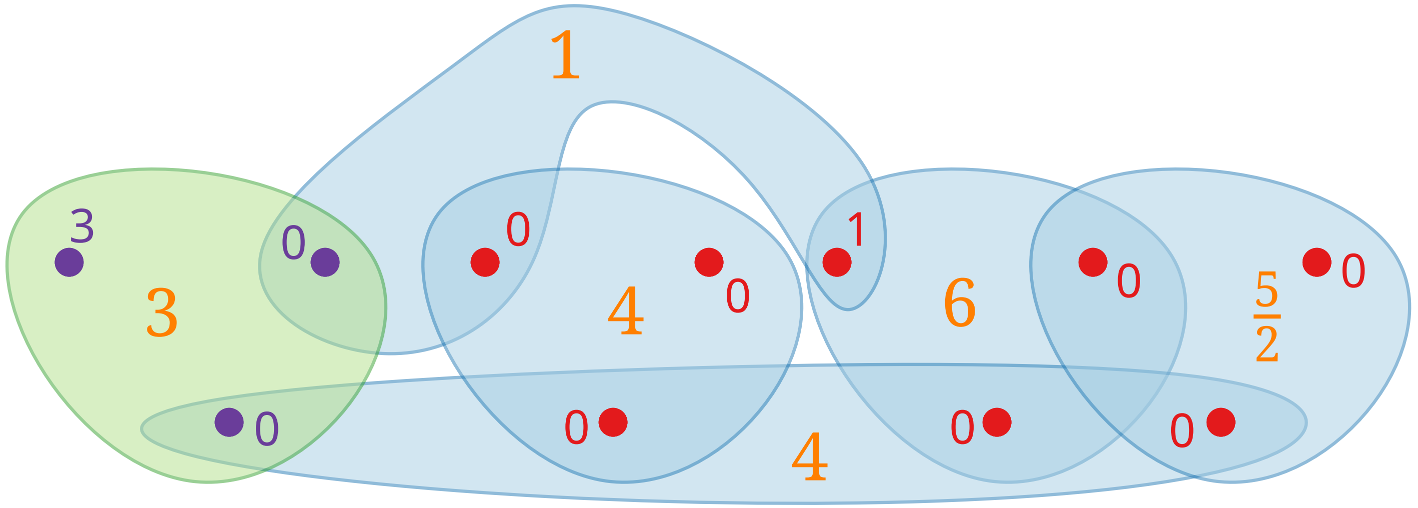
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0, y \leftarrow 0$

repeat

Select an uncovered element u.

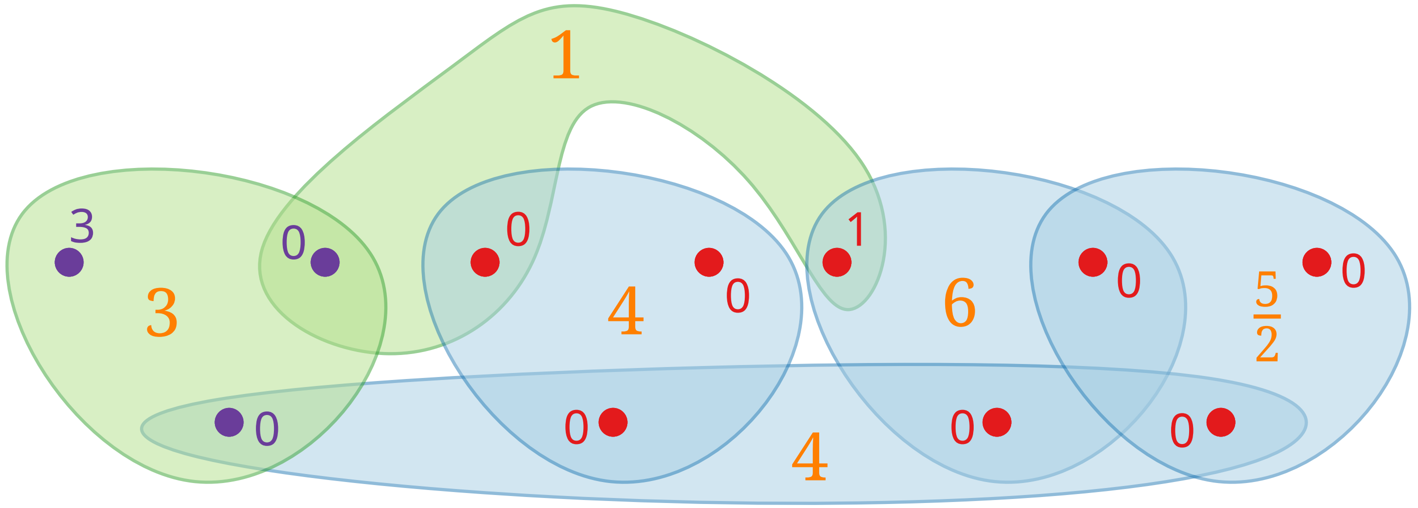
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

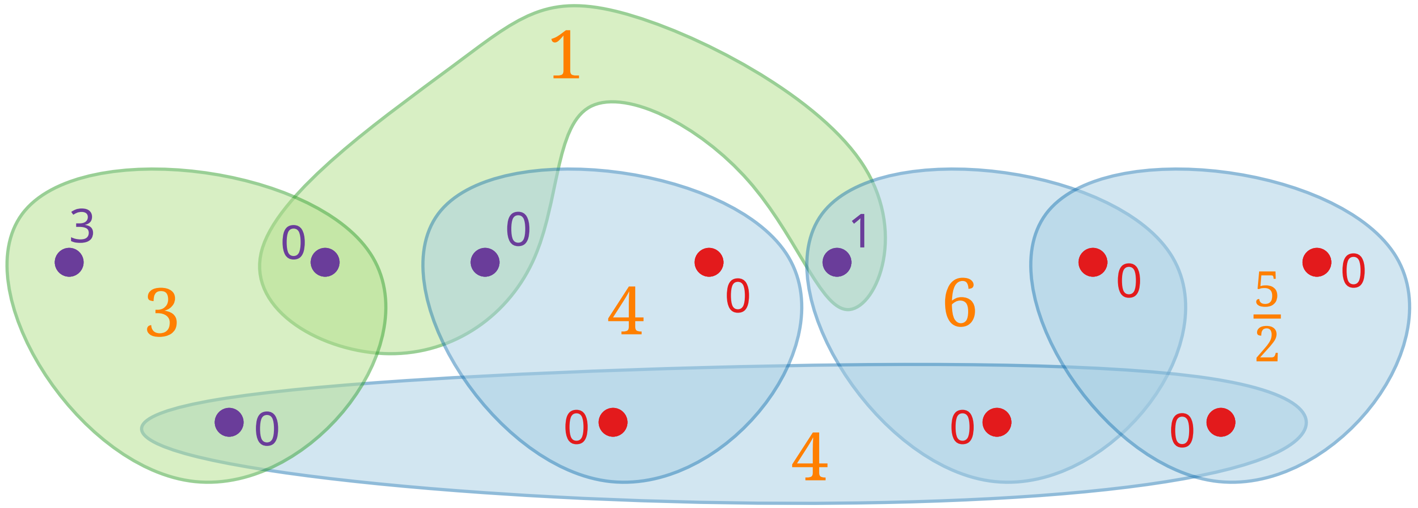
return x



Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$
repeat
  Select an uncovered element u.
  Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).
  Select all critical sets and update x.
  Mark all elements in these sets as covered.
until all elements are covered.
return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

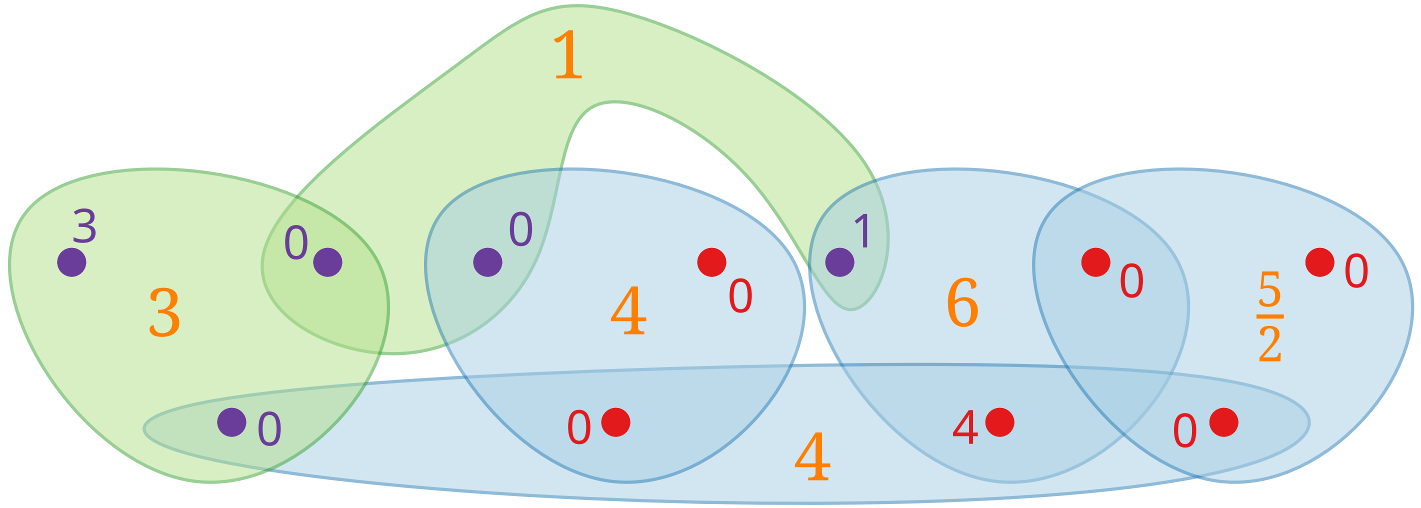
$x \leftarrow 0$, $y \leftarrow 0$

repeat

- Select an uncovered element u.
- Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).
- Select all critical sets and update x.
- Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$

repeat

Select an uncovered element u.

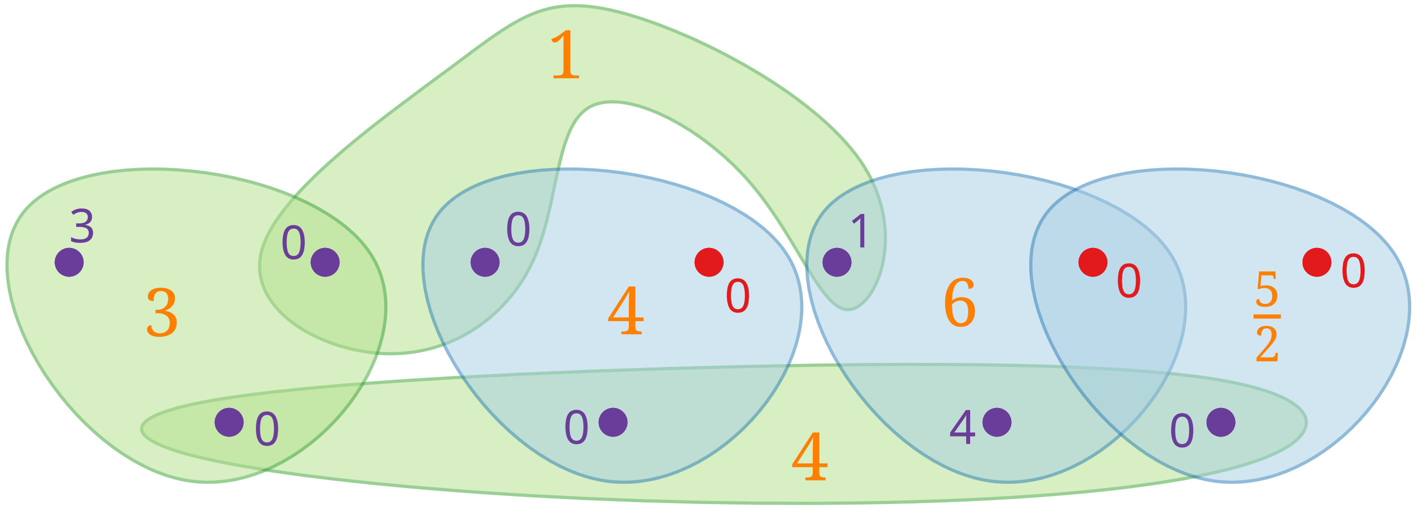
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0, y \leftarrow 0$

repeat

Select an uncovered element u.

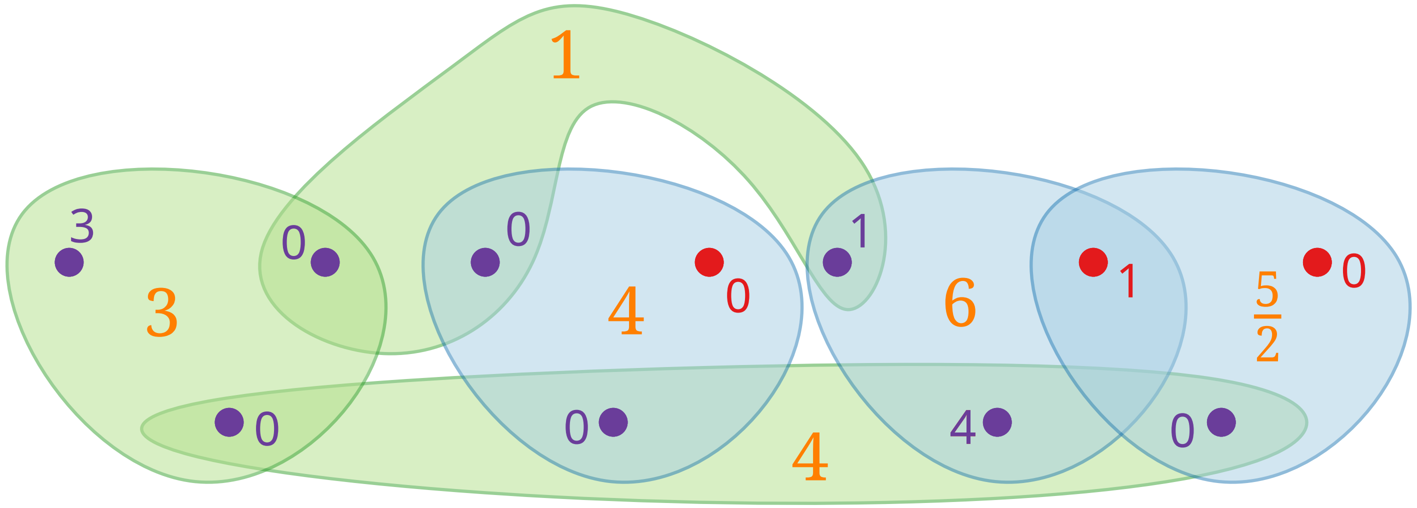
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0, y \leftarrow 0$

repeat

Select an uncovered element u.

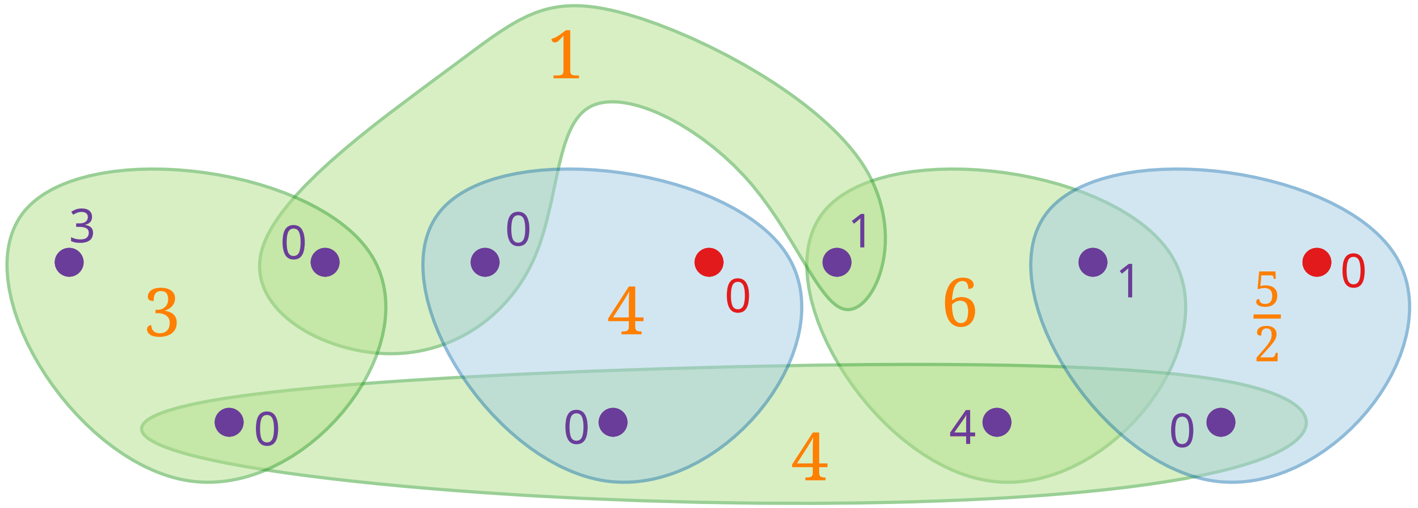
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

return x



Primal–Dual Schema for SETCOVER

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0$, $y \leftarrow 0$

repeat

Select an uncovered element u.

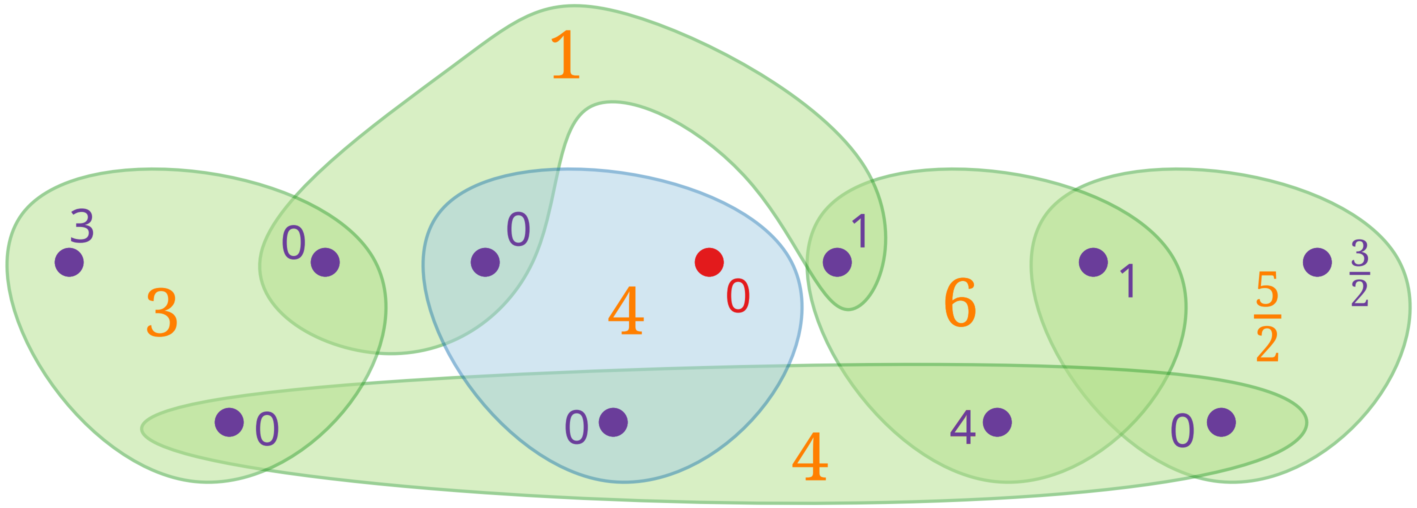
Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

Select all critical sets and update x.

Mark all elements in these sets as covered.

until all elements are covered.

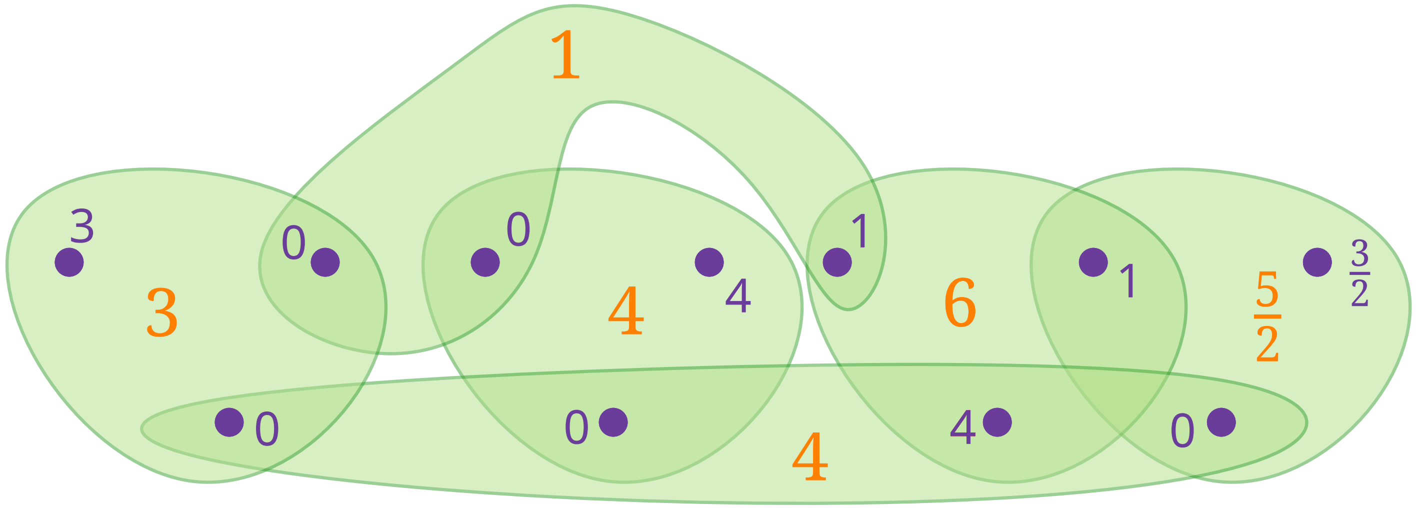
return x



Primal–Dual Schema for SetCover

PrimalDualSetCover$(U, \mathcal{S}, c)$

$x \leftarrow 0$, $y \leftarrow 0$
repeat
  Select an uncovered element u.
  Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).
  Select all critical sets and update x.
  Mark all elements in these sets as covered.
until all elements are covered.
return x



Primal–Dual Schema for SetCover

PrimalDualSetCover($U, \mathcal{S}, c$)

$x \leftarrow 0$, $y \leftarrow 0$

repeat

  Select an uncovered element u.

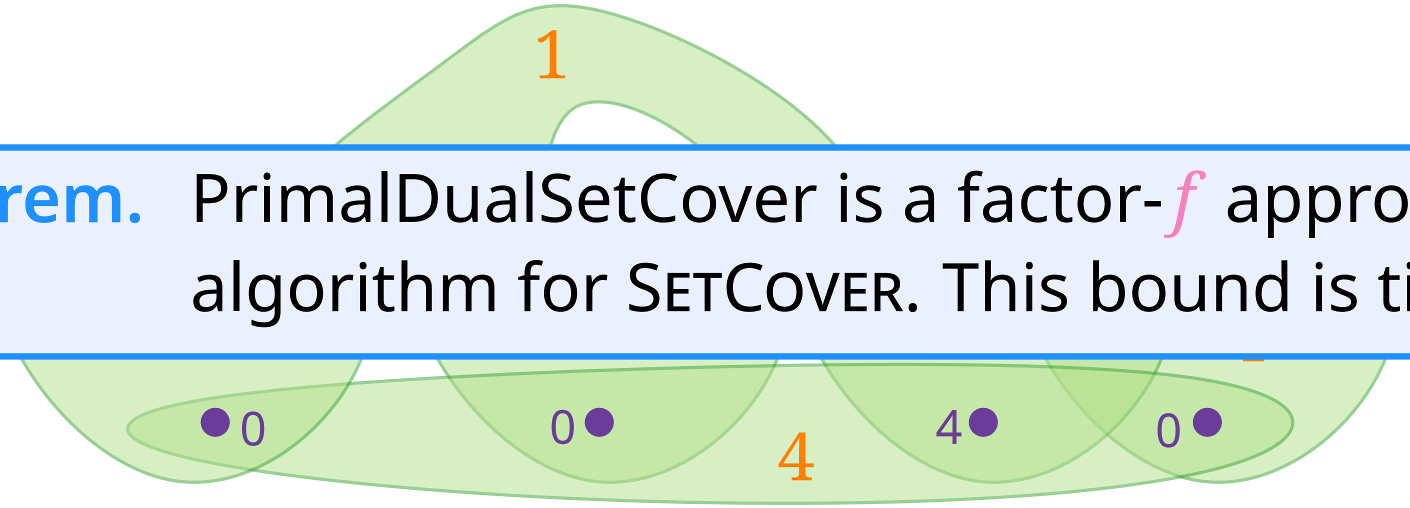
  Increase y_u until a set S is critical ($\sum_{u' \in S} y_{u'} = c_S$).

  Select all critical sets and update x.

  Mark all elements in these sets as covered.

until all elements are covered.

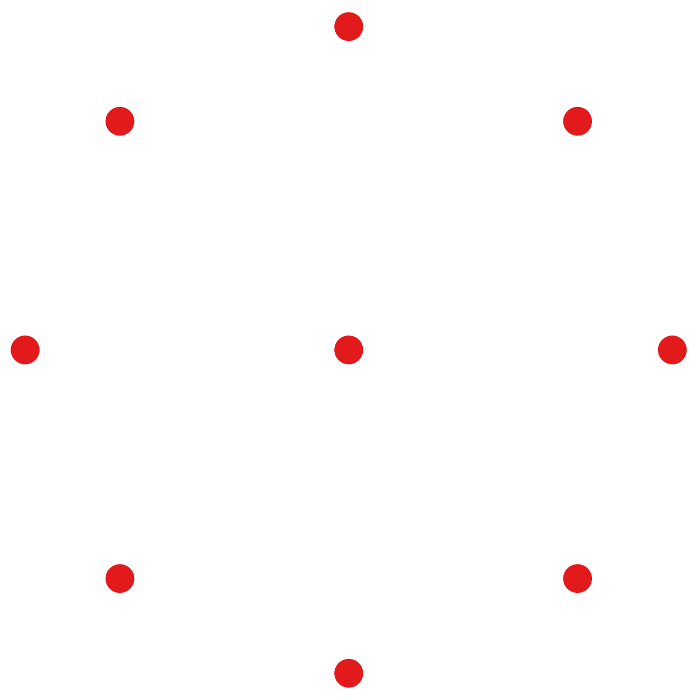
return x



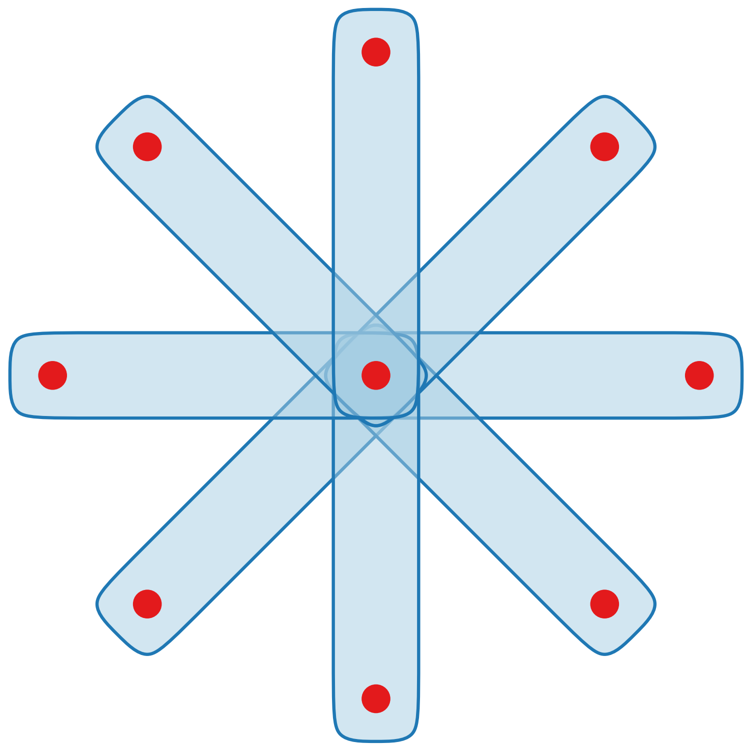
Theorem. PrimalDualSetCover is a factor-f approximation algorithm for SetCover. This bound is tight.

Tight Example

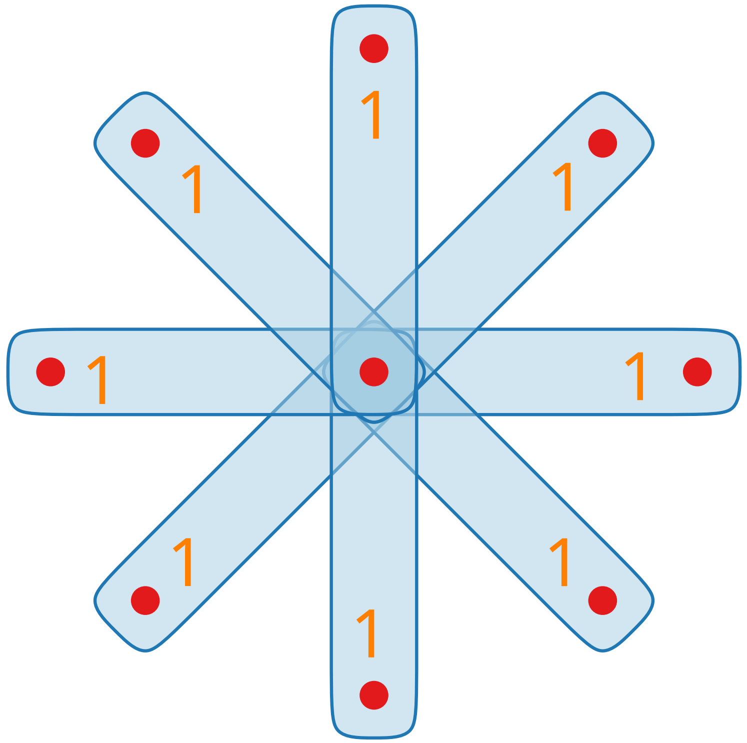
Tight Example



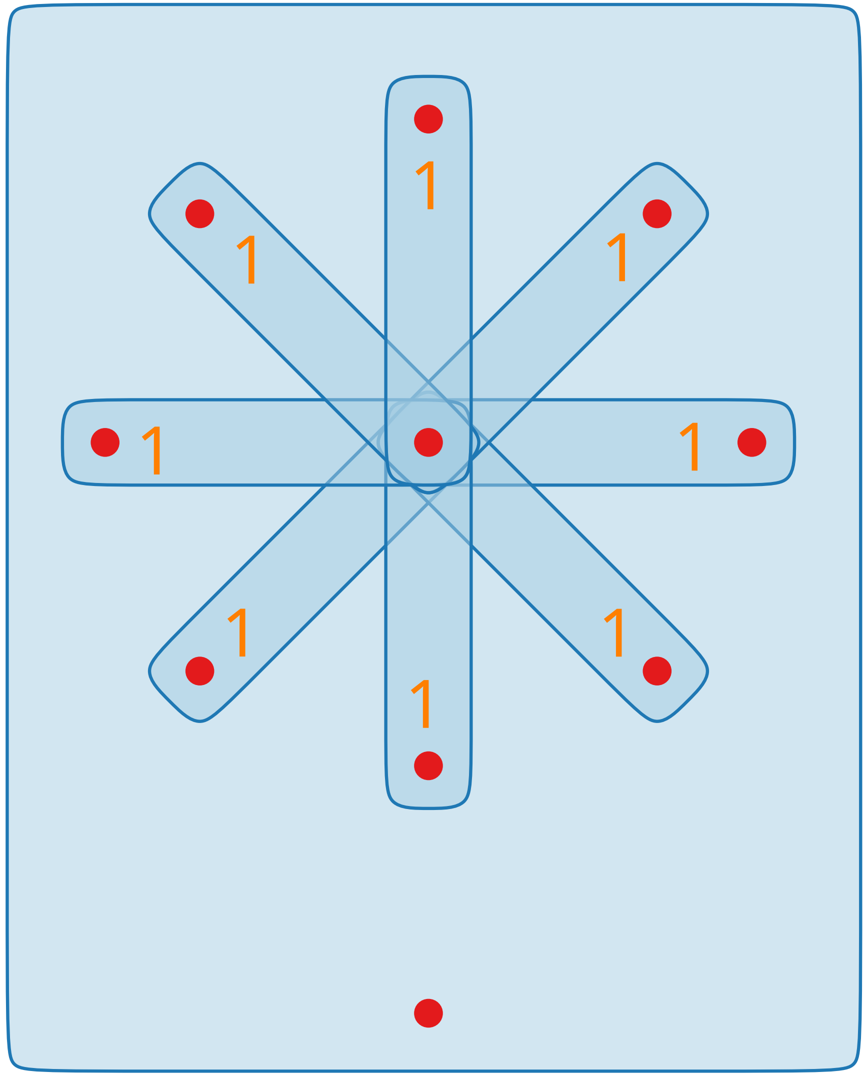
Tight Example



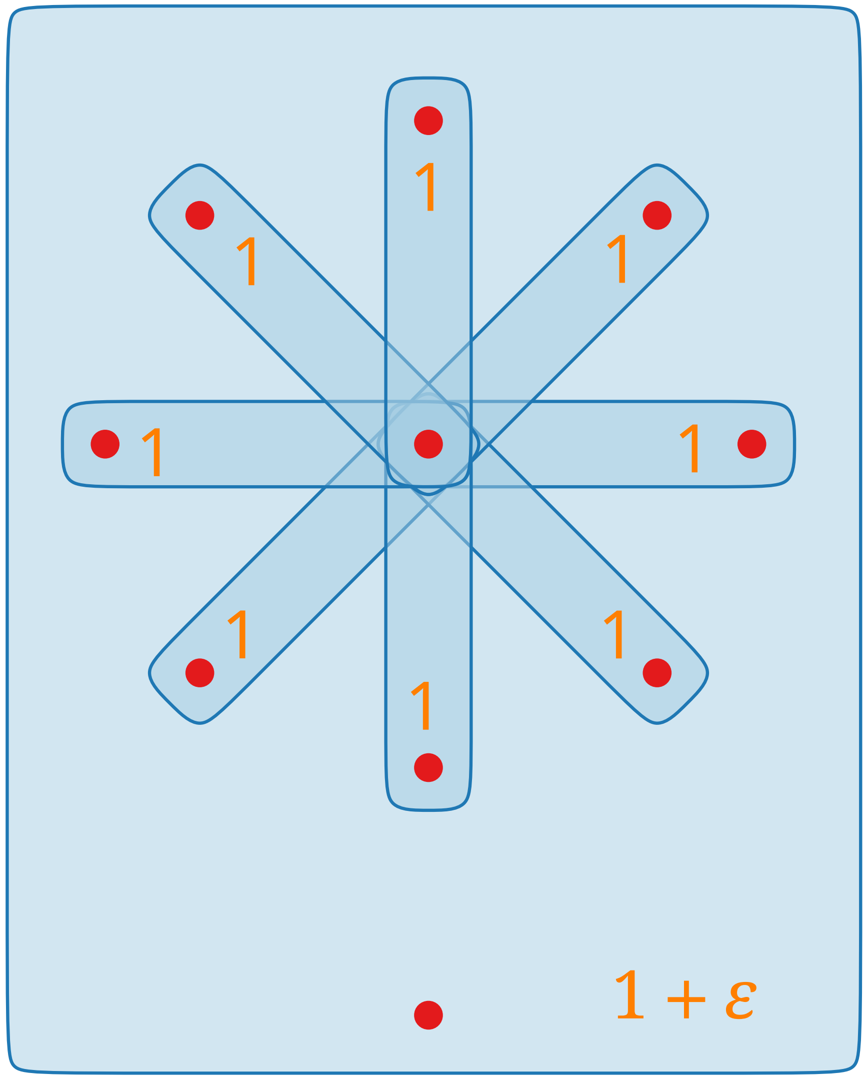
Tight Example



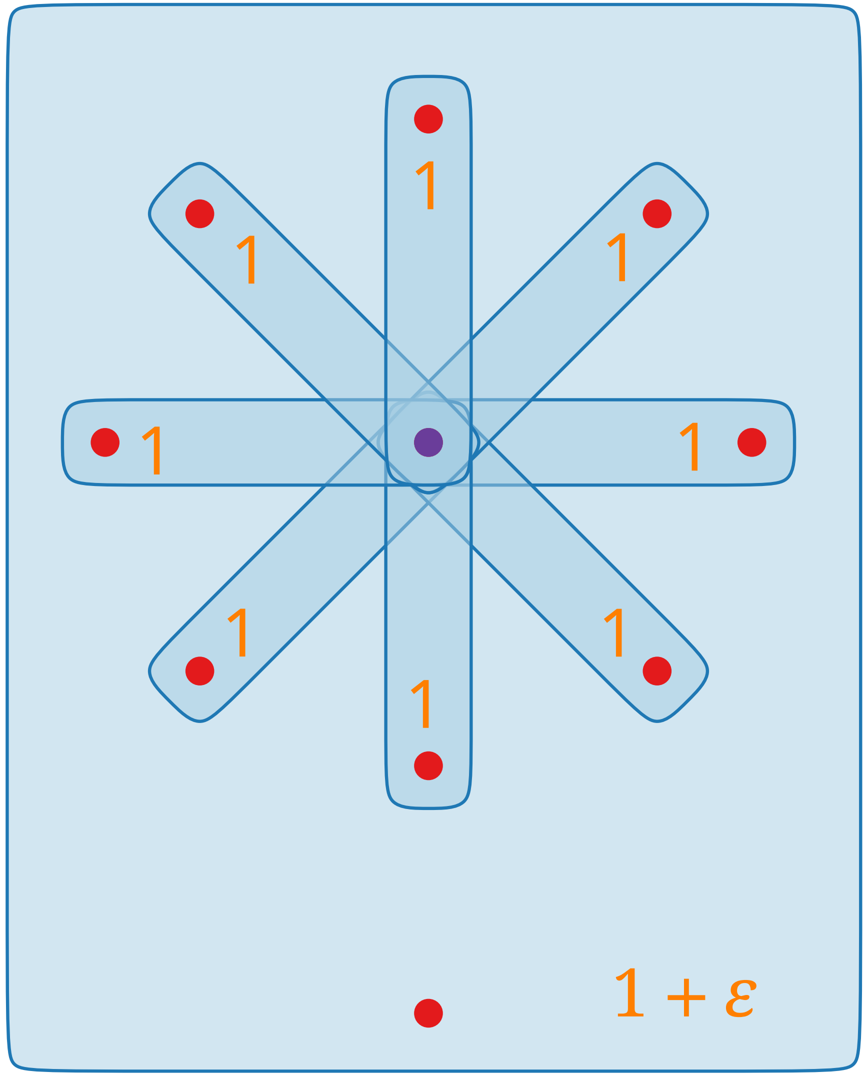
Tight Example



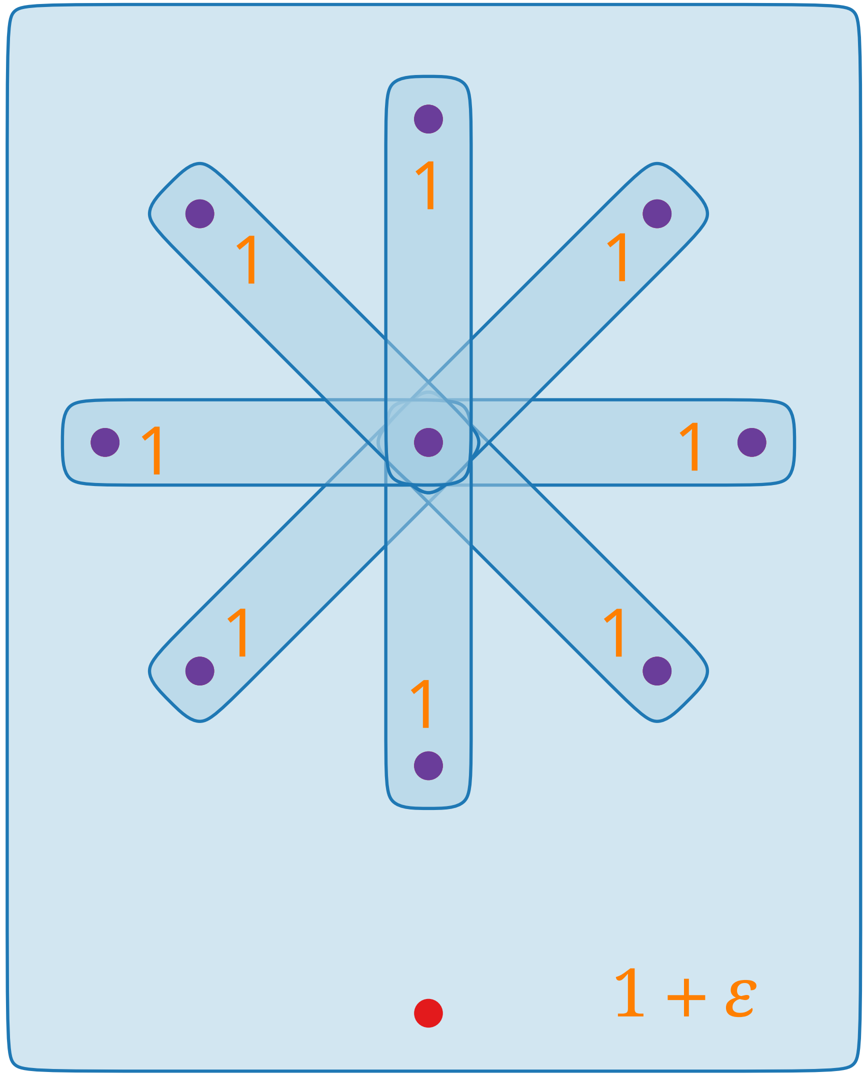
Tight Example



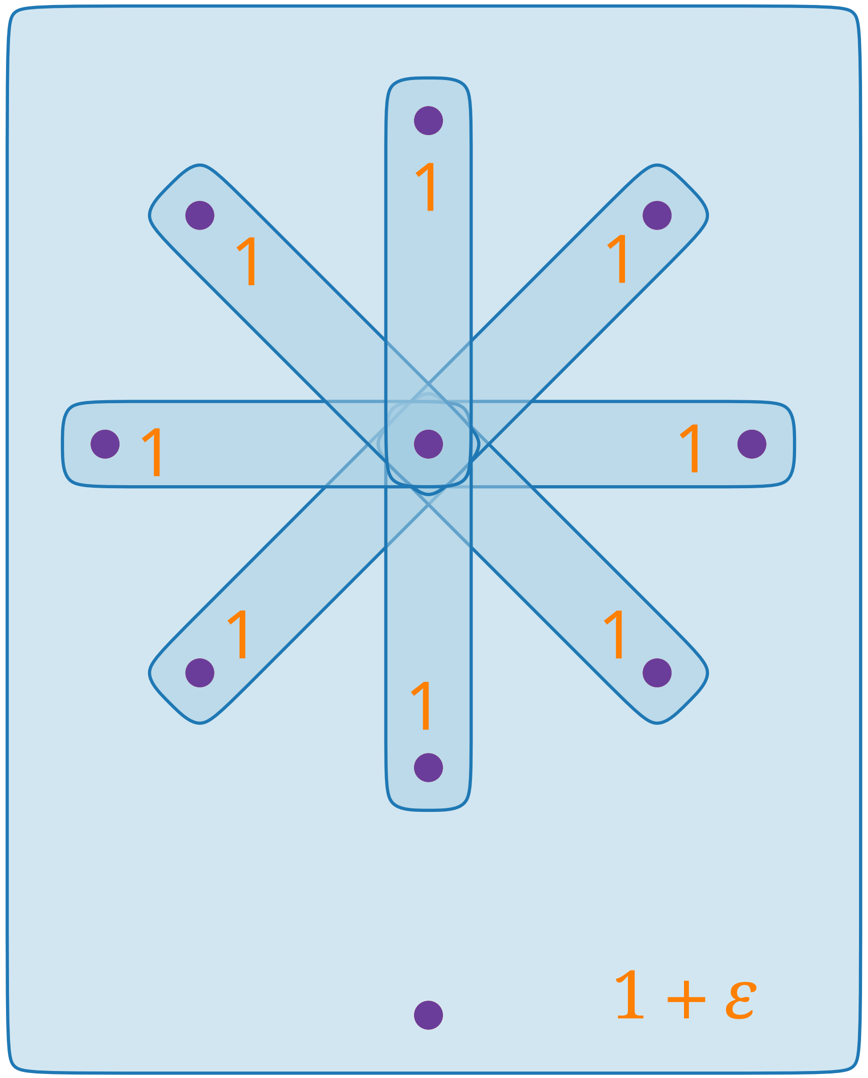
Tight Example



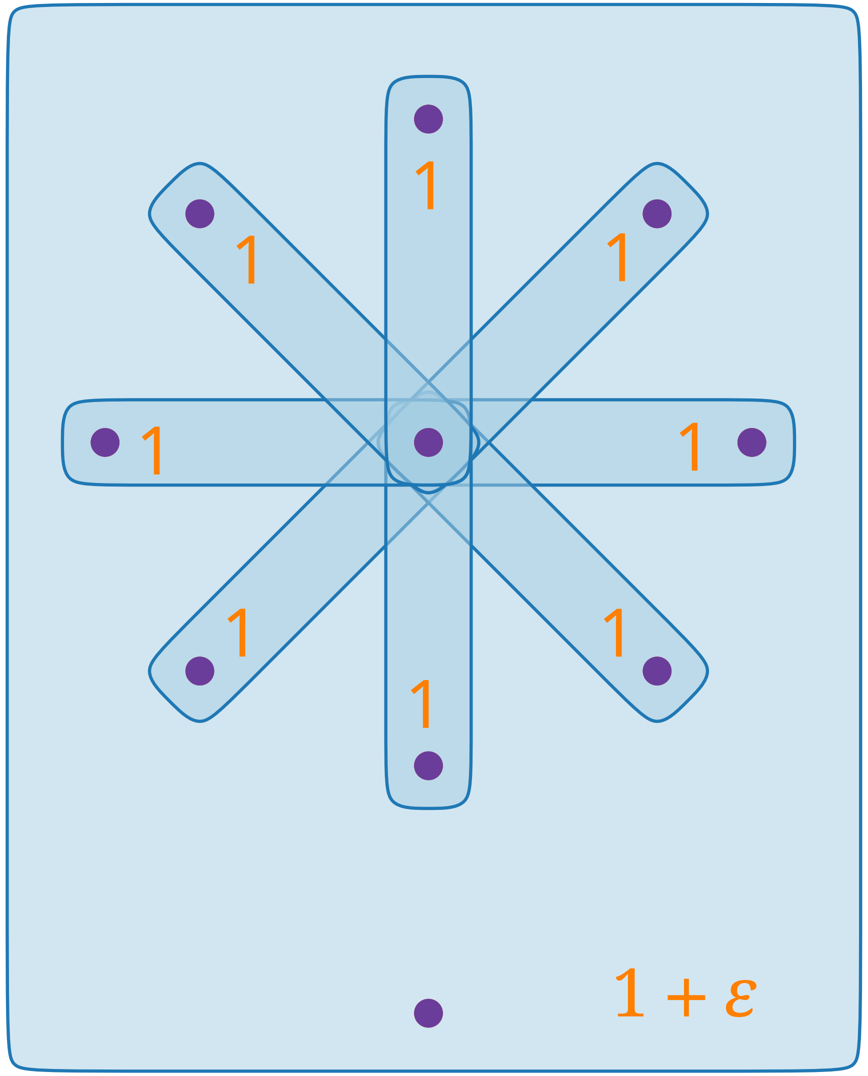
Tight Example



Tight Example

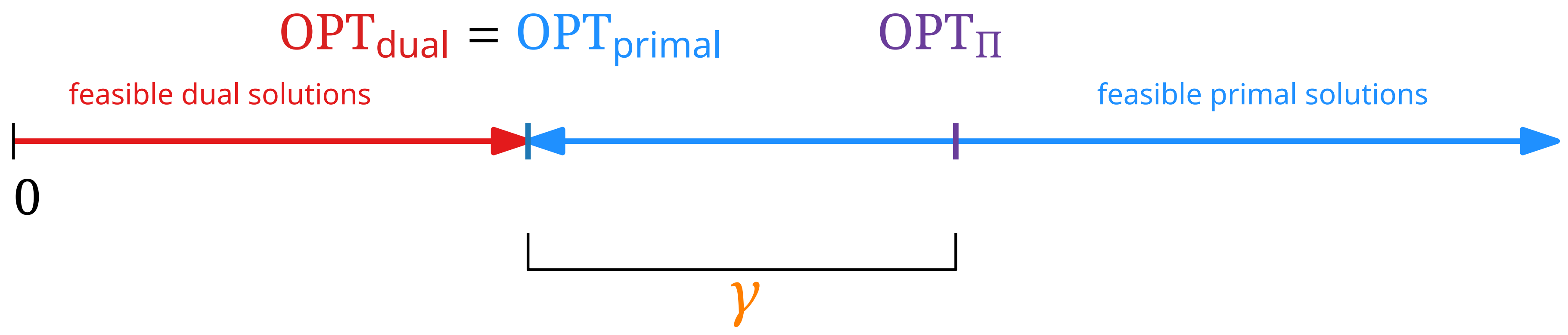


Tight Example



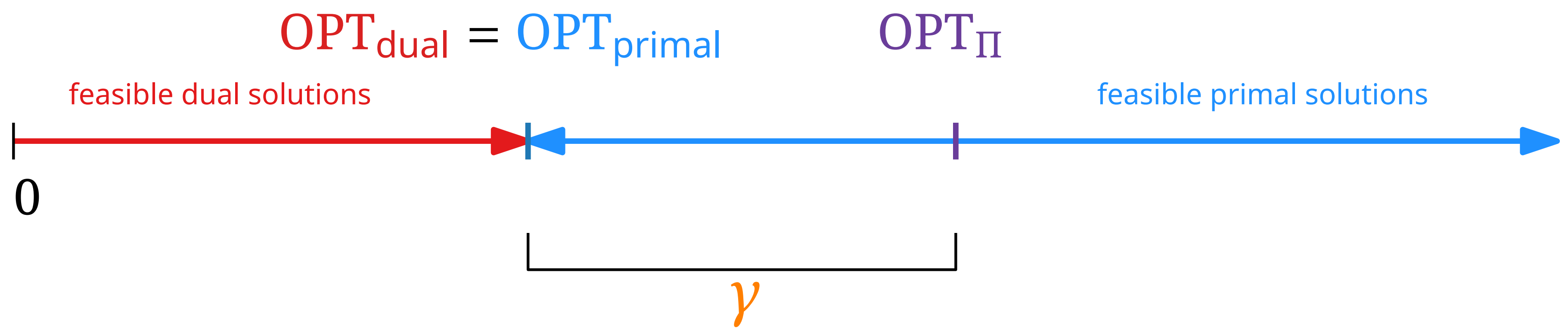
Algorithm may choose all sets at cost $n + \varepsilon$ whereas the optimum has cost $1 + \varepsilon$

Integrality Gap



Consider a minimization problem Π in ILP form.

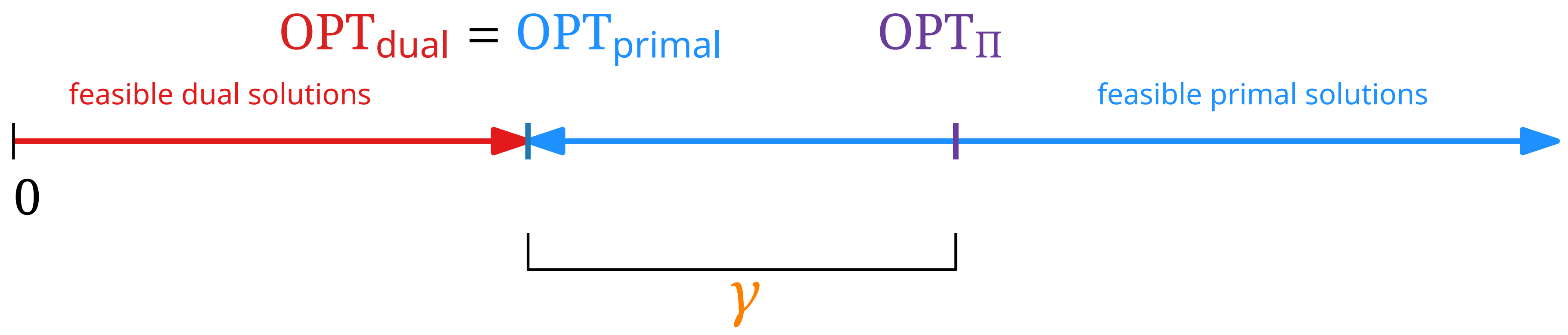
Integrality Gap



Consider a minimization problem Π in ILP form.

Dual methods (without outside help) are limited by the integrality gap of the LP-relaxation

Integrality Gap

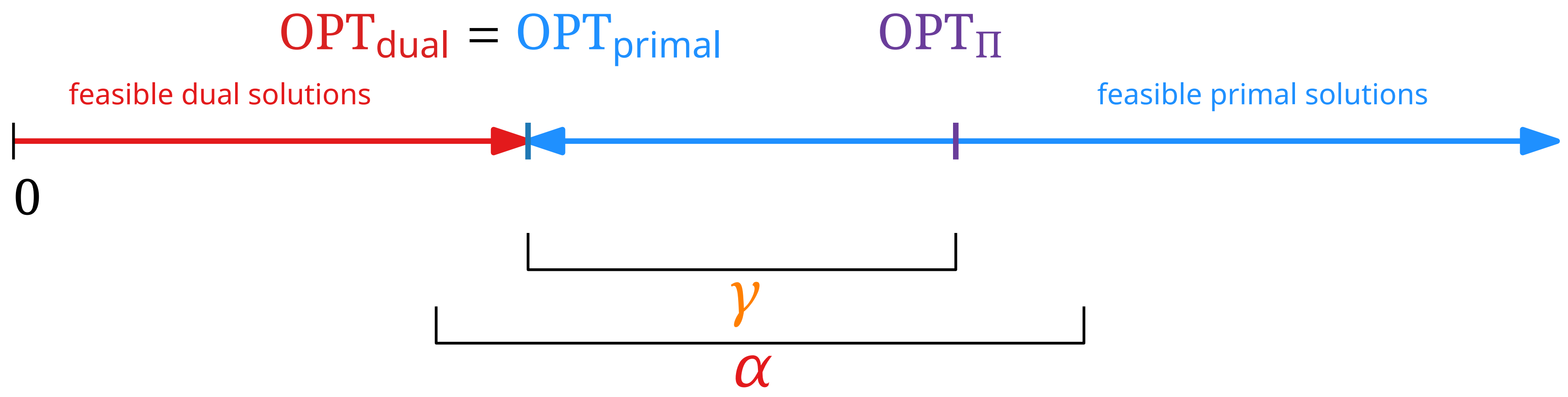


Consider a minimization problem Π in ILP form.

Dual methods (without outside help) are limited by the integrality gap of the LP-relaxation

$$\gamma = \sup_I \frac{\mathrm{OPT}_{\Pi}(I)}{\mathrm{OPT}_{\mathrm{primal}}(I)}$$

Integrality Gap



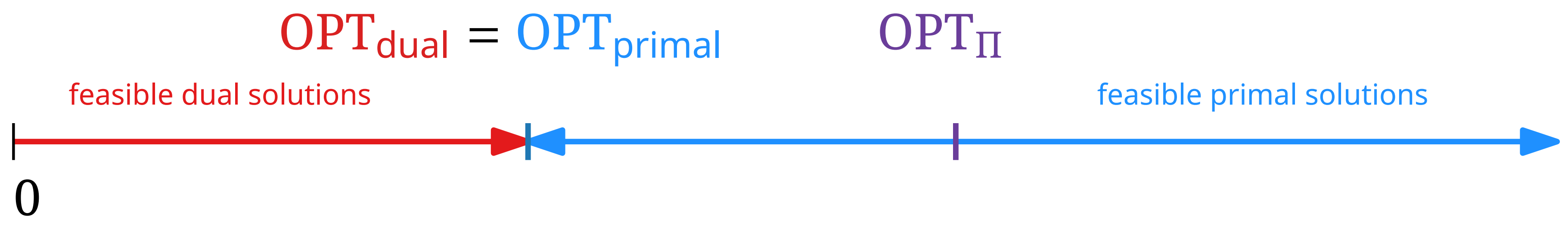
Consider a minimization problem Π in ILP form.

Dual methods (without outside help) are limited by the integrality gap of the LP-relaxation

$$\alpha \geq \gamma = \sup_I \frac{\mathrm{OPT}_{\Pi}(I)}{\mathrm{OPT}_{\text{primal}}(I)}$$

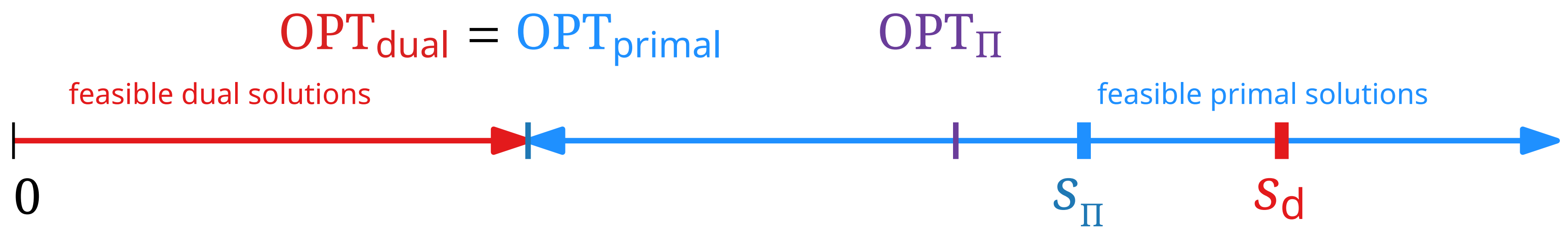
dual fitting

Technique III) Dual Fitting



Consider a minimization problem Π in ILP form.

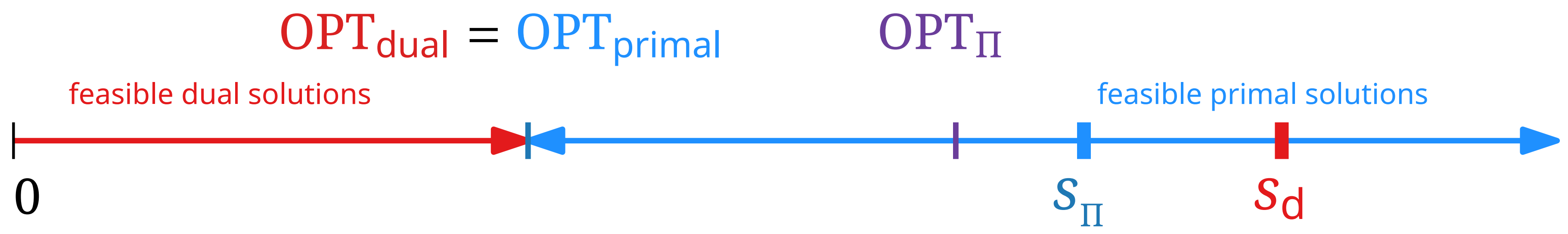
Technique III) Dual Fitting



Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely "pays" for s_{Π},

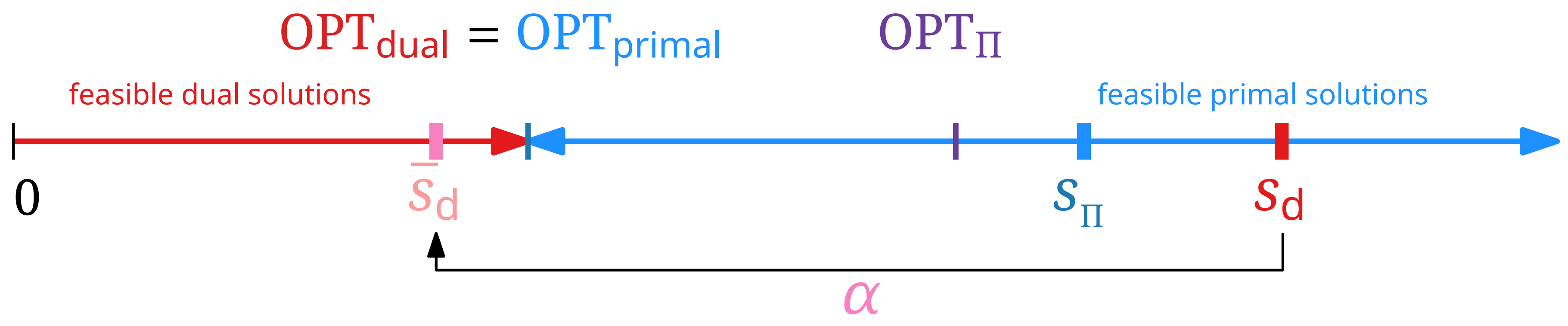
Technique III) Dual Fitting



Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely "pays" for s_{Π}, i.e., $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\mathrm{d}})$.

Technique III) Dual Fitting

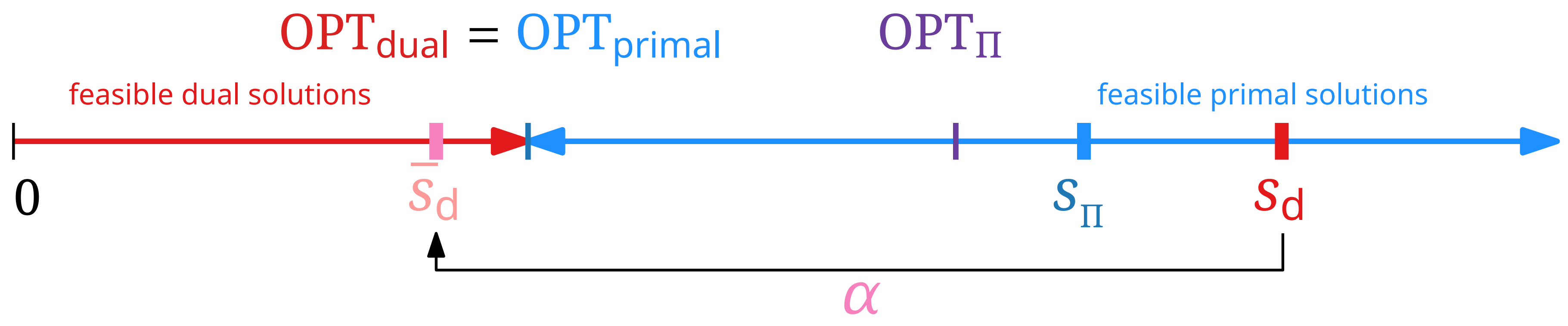


Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely "pays" for s_{Π}, i.e., $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\mathrm{d}})$.

Scale the dual variables $\rightsquigarrow$ feasible dual solution $\bar{s}_{\mathrm{d}}$.

Technique III) Dual Fitting



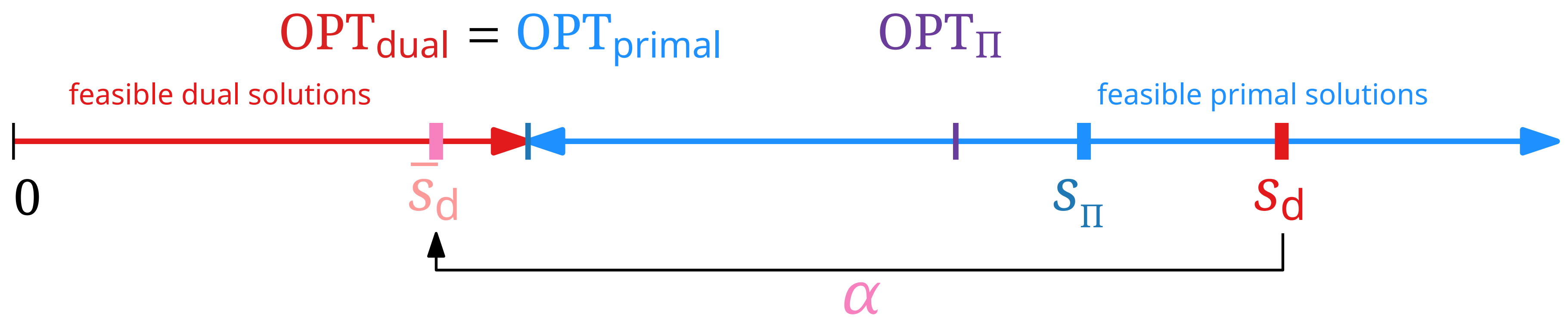
Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely “pays” for s_{Π}, i.e., $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\text{d}})$.

Scale the dual variables $\rightsquigarrow$ feasible dual solution $\bar{s}_{\text{d}}$.

$\Rightarrow$ $\mathrm{obj}(\bar{s}_{\text{d}}) \leq \mathrm{OPT}_{\text{dual}} \leq \mathrm{OPT}_{\Pi}$

Technique III) Dual Fitting



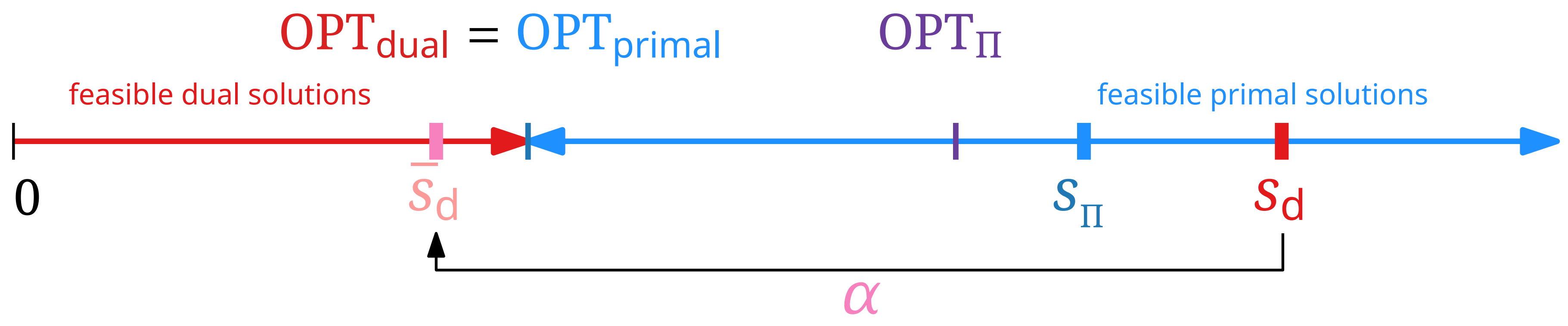
Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely "pays" for s_{Π}, i.e., $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\mathrm{d}})$.

Scale the dual variables $\rightsquigarrow$ feasible dual solution $\bar{s}_{\mathrm{d}}$.

$$\Rightarrow \qquad \mathrm{obj}(s_{\mathrm{d}})/\alpha = \mathrm{obj}(\bar{s}_{\mathrm{d}}) \leq \mathrm{OPT}_{\mathrm{dual}} \leq \mathrm{OPT}_{\Pi}$$

Technique III) Dual Fitting



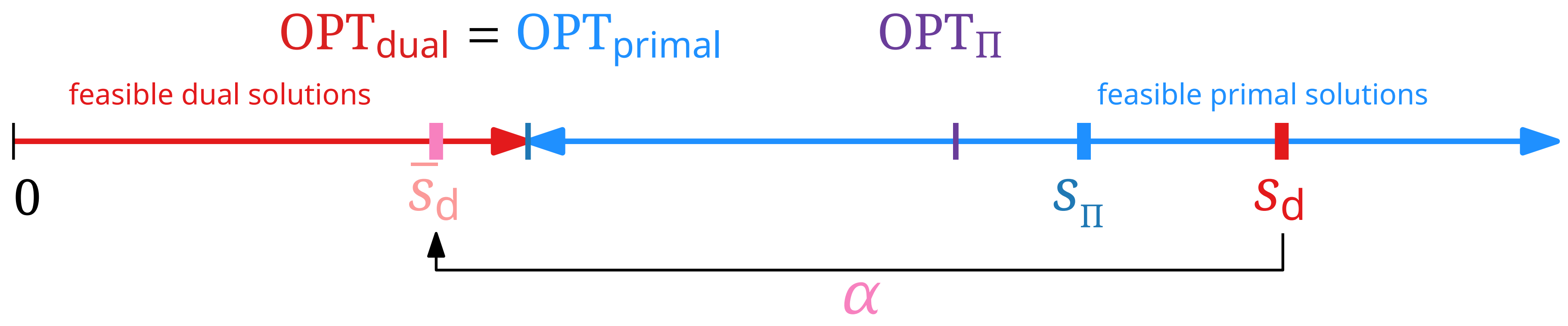
Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely "pays" for s_{Π}, i.e., $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\mathrm{d}})$.

Scale the dual variables $\rightsquigarrow$ feasible dual solution $\bar{s}_{\mathrm{d}}$.

$$\Rightarrow \mathrm{obj}(s_{\Pi})/\alpha \leq \mathrm{obj}(s_{\mathrm{d}})/\alpha = \mathrm{obj}(\bar{s}_{\mathrm{d}}) \leq \mathrm{OPT}_{\mathrm{dual}} \leq \mathrm{OPT}_{\Pi}$$

Technique III) Dual Fitting



Consider a minimization problem Π in ILP form.

Combinatorial algorithm (e.g., greedy) computes feasible primal solution s_{Π} and infeasible dual solution s_{d} that completely "pays" for s_{Π}, i.e., $\mathrm{obj}(s_{\Pi}) \leq \mathrm{obj}(s_{\mathrm{d}})$.

Scale the dual variables $\rightsquigarrow$ feasible dual solution $\bar{s}_{\mathrm{d}}$.

$\Rightarrow \mathrm{obj}(s_{\Pi})/\alpha \leq \mathrm{obj}(s_{\mathrm{d}})/\alpha = \mathrm{obj}(\bar{s}_{\mathrm{d}}) \leq \mathrm{OPT}_{\mathrm{dual}} \leq \mathrm{OPT}_{\Pi}$

$\Rightarrow$ Scaling factor α is approximation factor

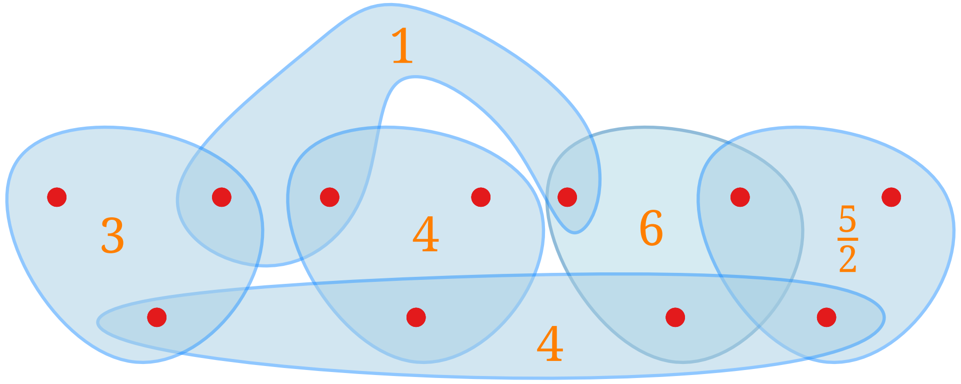
Dual Fitting for SetCover

Combinatorial (greedy) algorithm:

GreedySetCover$(U, \mathcal{S}, c)$

- $C \leftarrow \emptyset$
- $\mathcal{S}' \leftarrow \emptyset$
- **while** $C \neq U$ **do**
  - $S \leftarrow$ Set from $\mathcal{S}$ that minimizes $\frac{c(S)}{|S \setminus C|}$
  - **foreach** $u \in S \setminus C$ **do**
    - price$(u) \leftarrow \frac{c(S)}{|S \setminus C|}$
  - $C \leftarrow C \cup S$
  - $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S\}$
- **return** $\mathcal{S}'$ // Cover of U

What does the algorithm do on this example?



Dual Fitting for SetCover

Combinatorial (greedy) algorithm:

GreedySetCover$(U, \mathcal{S}, c)$

$C \leftarrow \emptyset$

$\mathcal{S}' \leftarrow \emptyset$

while $C \neq U$ **do**

  $S \leftarrow$ Set from $\mathcal{S}$ that minimizes $\frac{c(S)}{|S \setminus C|}$

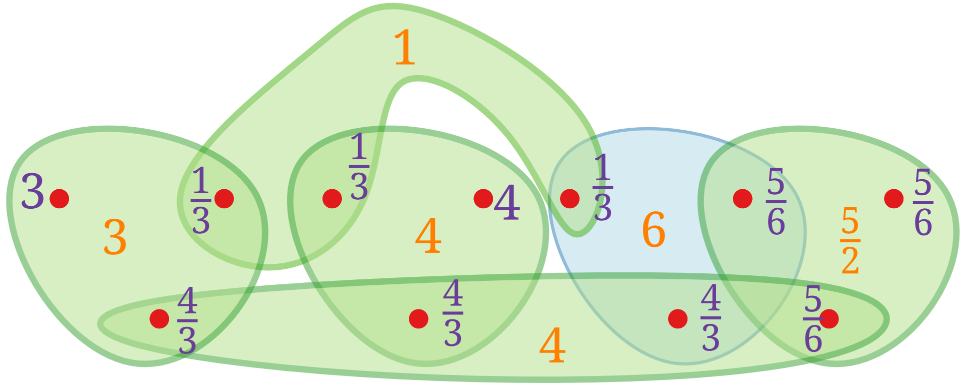
  foreach $u \in S \setminus C$ **do**

    $\text{price}(u) \leftarrow \frac{c(S)}{|S \setminus C|}$

  $C \leftarrow C \cup S$

  $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S\}$

return $\mathcal{S}'$ `// Cover of` U



Dual Fitting for SetCover

Combinatorial (greedy) algorithm:

GreedySetCover$(U, \mathcal{S}, c)$

$C \leftarrow \emptyset$

$\mathcal{S}' \leftarrow \emptyset$

while $C \neq U$ **do**

  $S \leftarrow$ Set from $\mathcal{S}$ that minimizes $\frac{c(S)}{|S \setminus C|}$

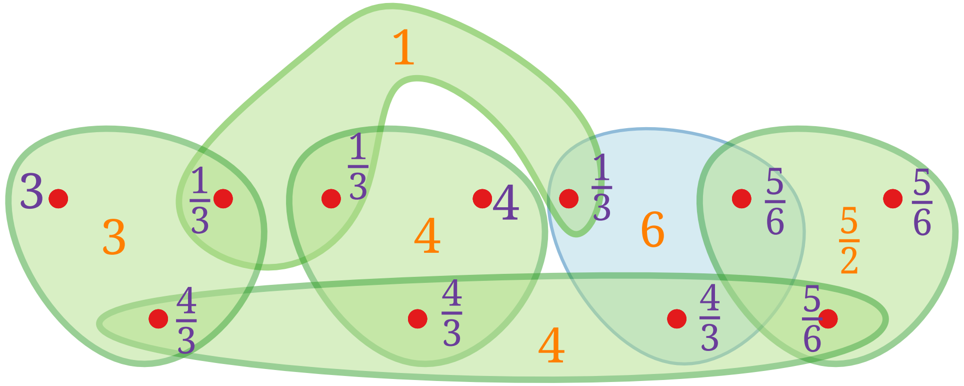
  foreach $u \in S \setminus C$ **do**

    price$(u) \leftarrow \frac{c(S)}{|S \setminus C|}$

  $C \leftarrow C \cup S$

  $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S\}$

return $\mathcal{S}'$ `// Cover of` U



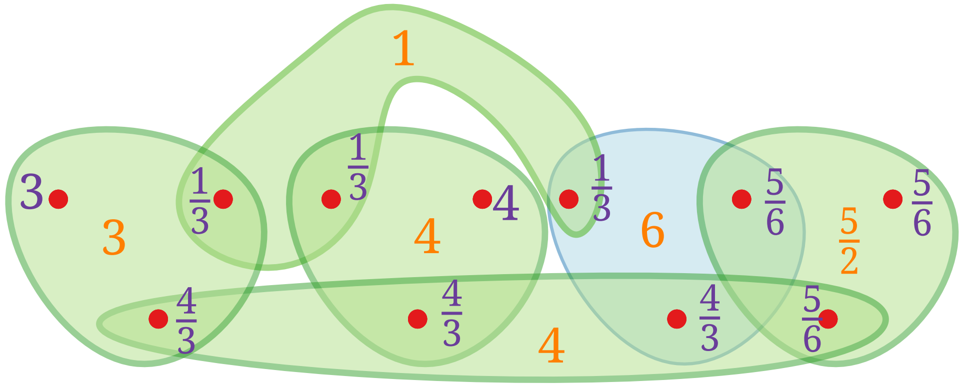
Reminder: $\sum_{u \in U}$ price(u) …

Dual Fitting for SetCover

Combinatorial (greedy) algorithm:

GreedySetCover$(U, \mathcal{S}, c)$

$C \leftarrow \emptyset$
$\mathcal{S}' \leftarrow \emptyset$
while $C \neq U$ **do**
  $S \leftarrow$ Set from $\mathcal{S}$ that minimizes $\frac{c(S)}{|S \setminus C|}$
  foreach $u \in S \setminus C$ **do**
    $\text{price}(u) \leftarrow \frac{c(S)}{|S \setminus C|}$
  $C \leftarrow C \cup S$
  $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{S\}$
return $\mathcal{S}'$ // Cover of U



Reminder: $\sum_{u \in U} \text{price}(u)$ completely pays for $\mathcal{S}'$.

New: LP-based Analysis

Observation. For each $u \in U$, $\mathrm{price}(u)$ is a dual variable

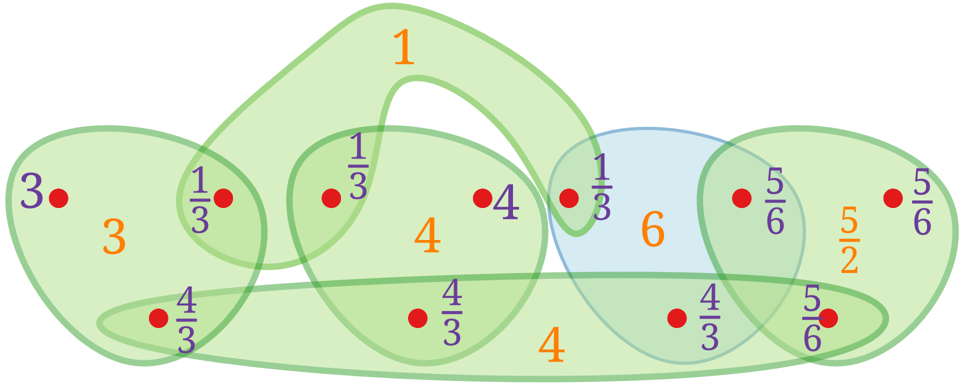
New: LP-based Analysis

Observation. For each $u \in U$, price(u) is a dual variable

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

New: LP-based Analysis

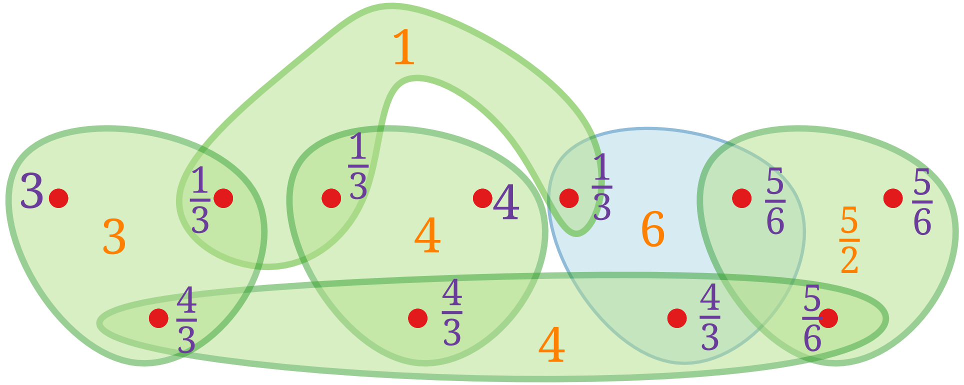
Observation. For each $u \in U$, price(u) is a dual variable



$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\
 & y_u \ge 0 & u \in U
\end{array}
$$

New: LP-based Analysis

Observation. For each $u \in U$, $\text{price}(u)$ is a dual variable
But this dual solution is in general not feasible.



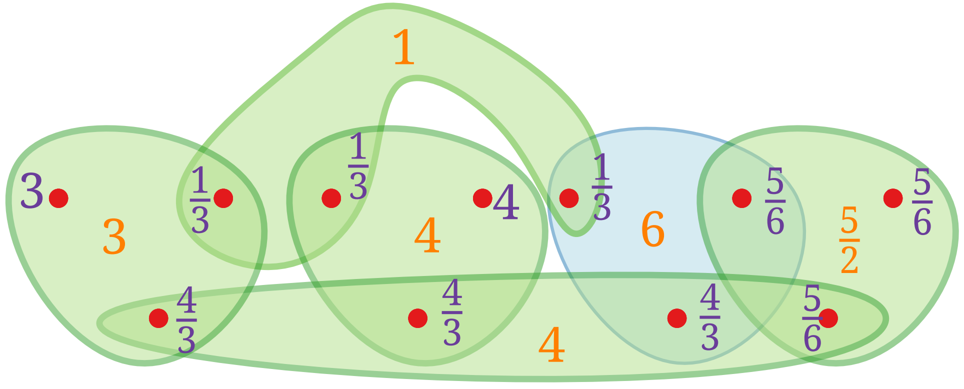
$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\
 & y_u \ge 0 & u \in U
\end{array}
$$

New: LP-based Analysis

Observation. For each $u \in U$, price(u) is a dual variable

But this dual solution is in general not feasible.

sets might be "overpacked"



maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \le c_S$	$S \in \mathcal{S}$
	$y_u \ge 0$	$u \in U$

New: LP-based Analysis

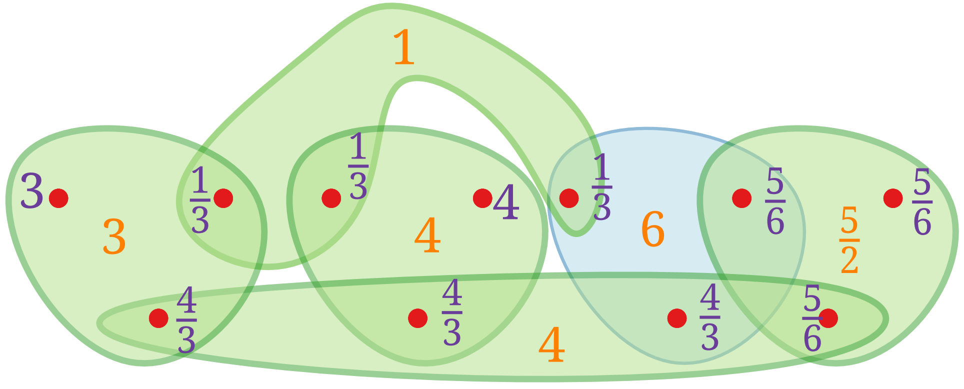
Observation. For each $u \in U$, price(u) is a dual variable

But this dual solution is in general not feasible.

sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.



$$\begin{array}{lll} \textbf{maximize} & \sum_{u \in U} y_u & \\ \textbf{subject to} & \sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

New: LP-based Analysis

Observation. For each $u \in U$, price(u) is a dual variable

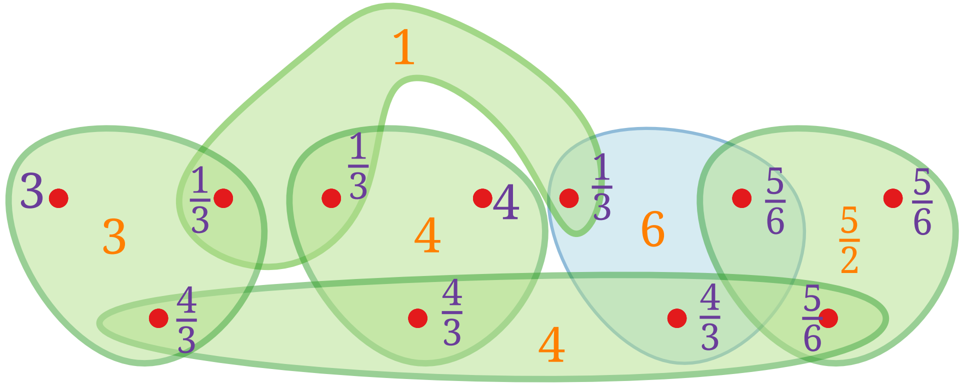
But this dual solution is in general not feasible.

sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.

Take $\bar{y}_u =$



maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

New: LP-based Analysis

Observation. For each $u \in U$, price(u) is a dual variable

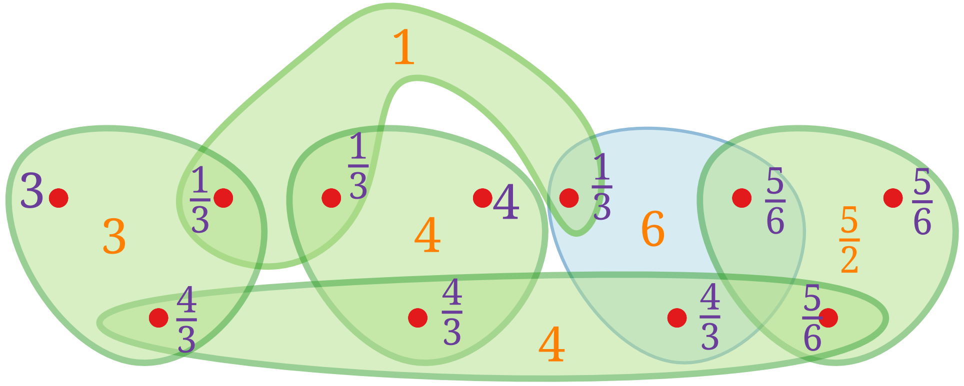
But this dual solution is in general not feasible.

sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.

Take $\bar{y}_u = \quad \text{price}(u)/$



$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\ & y_u \ge 0 & u \in U \end{array}$$

New: LP-based Analysis

Observation. For each $u \in U$, $\text{price}(u)$ is a dual variable

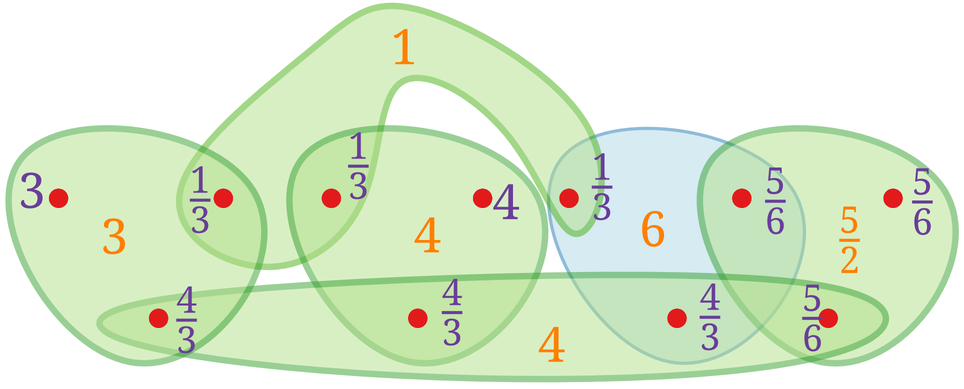
But this dual solution is in general not feasible.

sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.

Take $\bar{y}_u = \text{price}(u) / \mathcal{H}_k$.



$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\ & y_u \ge 0 & u \in U \end{array}$$

New: LP-based Analysis

Observation. For each $u \in U$, $\text{price}(u)$ is a dual variable

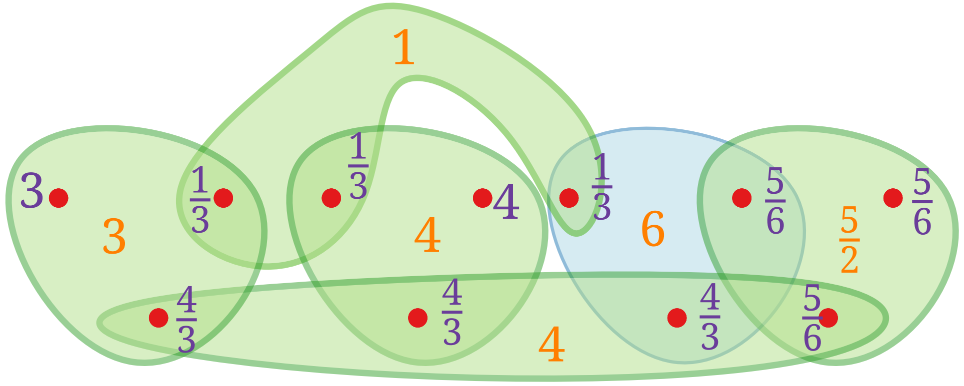
But this dual solution is in general not feasible.

sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.

Take $\bar{y}_u = \text{price}(u) / \mathcal{H}_k$. ($k =$ size of largest set in $\mathcal{S}$; $\mathcal{H}_k$: k^{th} harmonic number)



$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\
& y_u \ge 0 & u \in U
\end{array}
$$

New: LP-based Analysis

Observation. For each $u \in U$, $\text{price}(u)$ is a dual variable

But this dual solution is in general not feasible.

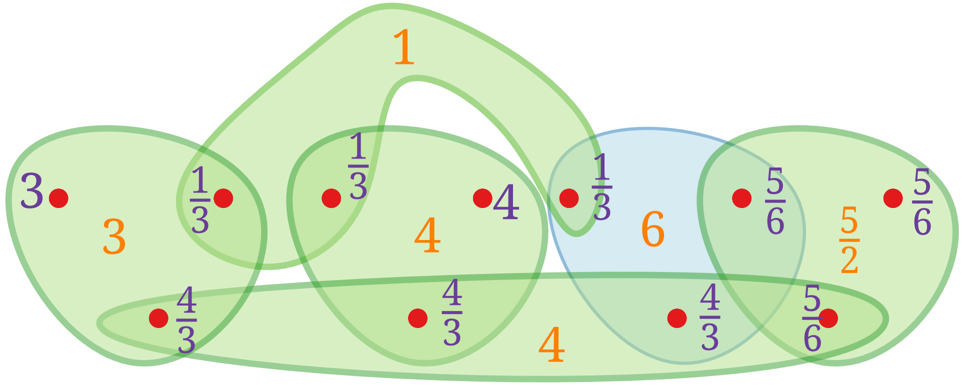
sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.

Take $\bar{y}_u = \text{price}(u) / \mathcal{H}_k$. ($k =$ size of largest set in $\mathcal{S}$; $\mathcal{H}_k$: k^{th} harmonic number)

The greedy algorithm uses these dual variables as lower bound for OPT.



$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\
 & y_u \ge 0 & u \in U
\end{array}
$$

New: LP-based Analysis

Observation. For each $u \in U$, $\text{price}(u)$ is a dual variable

But this dual solution is in general not feasible.

sets might be "overpacked"

Dual-fitting trick:

Scale dual variables such that no set is overpacked.

Take $\bar{y}_u = \quad \text{price}(u) / \mathcal{H}_k$. ($k =$ size of largest set in $\mathcal{S}$; $\mathcal{H}_k$: k^{th} harmonic number)

The greedy algorithm uses these dual variables as lower bound for OPT.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$
\begin{array}{lll}
\textbf{maximize} & \sum_{u \in U} y_u & \\
\textbf{subject to} & \sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\
 & y_u \ge 0 & u \in U
\end{array}
$$

Proof.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$\begin{array}{lll} \textbf{maximize} & \sum_{u \in U} y_u & \\ \textbf{subject to} & \sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\ & y_u \ge 0 & u \in U \end{array}$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \le c_S & S \in \mathcal{S} \\
 & y_u \ge 0 & u \in U
\end{array}
$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\
 & y_u \geq 0 & u \in U
\end{array}
$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S – in the order in which they are covered by greedy.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$\begin{array}{lll} \textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\ \textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$
\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\
 & y_u \geq 0 & u \in U
\end{array}
$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \le k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\ge \ell - i + 1$ elem. of S are uncovered.

So $\mathrm{price}(u_i) \le$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \le c_S$	$S \in \mathcal{S}$
	$y_u \ge 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\text{price}(u_i) \leq \; c(S)/(\ell - i + 1)$.

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$
\begin{array}{lll}
\textbf{maximize} & \sum_{u \in U} y_u & \\
\textbf{subject to} & \sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\
 & y_u \geq 0 & u \in U
\end{array}
$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \dots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\mathrm{price}(u_i) \leq \quad c(S)/(\ell - i + 1)$.

$\Rightarrow \bar{y}_{u_i} \leq$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\text{price}(u_i) \leq \quad c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell - i + 1}$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$\begin{array}{lll}
\textbf{maximize} & \displaystyle\sum_{u \in U} y_u & \\
\textbf{subject to} & \displaystyle\sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\
& y_u \geq 0 & u \in U
\end{array}$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\mathrm{price}(u_i) \leq \; c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell-i+1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.
Before that $\geq \ell - i + 1$ elem. of S are uncovered.
So $\text{price}(u_i) \leq \ c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell - i + 1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \Big($$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\mathrm{price}(u_i) \leq c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell - i + 1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \left(\frac{1}{\ell} + \cdots + \frac{1}{1}\right)$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$\begin{array}{lll} \textbf{maximize} & \sum_{u \in U} y_u & \\ \textbf{subject to} & \sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\text{price}(u_i) \leq \; c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell-i+1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \overbrace{\left(\frac{1}{\ell} + \cdots + \frac{1}{1}\right)}$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\mathrm{price}(u_i) \leq \; c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell - i + 1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \overbrace{\left(\frac{1}{\ell} + \cdots + \frac{1}{1}\right)}^{= \mathcal{H}_\ell}$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

$$\begin{array}{lll} \textbf{maximize} & \sum_{u \in U} y_u & \\ \textbf{subject to} & \sum_{u \in S} y_u \leq c_S & S \in \mathcal{S} \\ & y_u \geq 0 & u \in U \end{array}$$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \ldots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.

Before that $\geq \ell - i + 1$ elem. of S are uncovered.

So $\mathrm{price}(u_i) \leq \; c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell - i + 1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \overbrace{\left(\tfrac{1}{\ell} + \cdots + \tfrac{1}{1}\right)}^{= \mathcal{H}_\ell \,\leq\, \mathcal{H}_k}$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Proof. To prove: No set is overpacked by $\bar{y}$.

Let $S \in \mathcal{S}$ and $\ell = |S| \leq k$.

Let $u_1, \dots, u_\ell$ be the elements of S –
in the order in which they are covered by greedy.

Consider the iteration in which u_i is covered.
Before that $\geq \ell - i + 1$ elem. of S are uncovered.
So $\mathrm{price}(u_i) \leq \; c(S)/(\ell - i + 1)$.

$$\Rightarrow \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \frac{1}{\ell-i+1} \quad \Rightarrow \sum_{i=1}^{\ell} \bar{y}_{u_i} \leq \frac{c(S)}{\mathcal{H}_k} \cdot \overbrace{\left(\tfrac{1}{\ell} + \cdots + \tfrac{1}{1}\right)}^{= \mathcal{H}_\ell \, \leq \mathcal{H}_k}$$

$$\leq c(S) \qquad \square$$

Lemma.
The vector $\bar{y} = (\bar{y}_u)_{u \in U}$ is a feasible solution for the dual LP.

maximize	$\sum_{u \in U} y_u$	
subject to	$\sum_{u \in S} y_u \leq c_S$	$S \in \mathcal{S}$
	$y_u \geq 0$	$u \in U$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof. $\mathrm{ALG} = c(\mathcal{S}') \leq$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof. $\text{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \text{price}(u) =$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof. $\mathrm{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \mathrm{price}(u) = \mathcal{H}_k \cdot \sum_{u \in U} \bar{y}_u$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof.

$$\text{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \text{price}(u) = \mathcal{H}_k \cdot \sum_{u \in U} \bar{y}_u$$

$$\leq \mathcal{H}_k \cdot \text{OPT}_{\text{relax}}$$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof.

$$\begin{aligned} \text{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \text{price}(u) &= \mathcal{H}_k \cdot \sum_{u \in U} \bar{y}_u \\ &\leq \mathcal{H}_k \cdot \text{OPT}_{\text{relax}} \\ &\leq \mathcal{H}_k \cdot \text{OPT} \qquad \square \end{aligned}$$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof.

$$\begin{aligned} \mathrm{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \mathrm{price}(u) &= \mathcal{H}_k \cdot \sum_{u \in U} \bar{y}_u \\ &\leq \mathcal{H}_k \cdot \mathrm{OPT}_{\mathrm{relax}} \\ &\leq \mathcal{H}_k \cdot \mathrm{OPT} \qquad \square \end{aligned}$$

Also shows: Integrality gap of relaxation $\leq \mathcal{H}_k$

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof.

$$\begin{aligned} \text{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \text{price}(u) &= \mathcal{H}_k \cdot \sum_{u \in U} \bar{y}_u \\ &\leq \mathcal{H}_k \cdot \text{OPT}_{\text{relax}} \\ &\leq \mathcal{H}_k \cdot \text{OPT} \qquad \square \end{aligned}$$

Also shows: Integrality gap of relaxation $\leq \mathcal{H}_k$

Dual solution allows a per-instance estimation $c(\mathcal{S}')/\text{OPT}_{\text{relax}}$ of the quality of the greedy solution

Result for Dual Fitting

Theorem. GreedySetCover is a factor-$\mathcal{H}_k$ approximation algorithm for SetCover, where $k = \max_{S \in \mathcal{S}} |S|$.

Proof.

$$
\begin{aligned}
\mathrm{ALG} = c(\mathcal{S}') \leq \sum_{u \in U} \mathrm{price}(u) &= \mathcal{H}_k \cdot \sum_{u \in U} \bar{y}_u \\
&\leq \mathcal{H}_k \cdot \mathrm{OPT}_{\mathrm{relax}} \\
&\leq \mathcal{H}_k \cdot \mathrm{OPT} \qquad \square
\end{aligned}
$$

Also shows: Integrality gap of relaxation $\leq \mathcal{H}_k$

Dual solution allows a per-instance estimation $c(\mathcal{S}')/\mathrm{OPT}_{\mathrm{relax}}$

of the quality of the greedy solution

…which may be stronger than the worst-case bound $\mathcal{H}_k$.

Summary

Linear Programming provides lower bounds for minimization problems
(and upper bounds for maximization problems):

- LP relaxation
- Duality

Summary

Linear Programming provides lower bounds for minimization problems
(and upper bounds for maximization problems):

- LP relaxation
- Duality

This can be used to design approximation algorithms:

- LP rounding
- Primal-Dual Method
- Dual Fitting

Summary

Linear Programming provides lower bounds for minimization problems
(and upper bounds for maximization problems):

- LP relaxation
- Duality

This can be used to design approximation algorithms:

- LP rounding
- Primal-Dual Method
- Dual Fitting

Primal-Dual Method and Dual Fitting does not require to solve LPs
$\longrightarrow$ often faster algorithms

Summary

Linear Programming provides lower bounds for minimization problems
(and upper bounds for maximization problems):

- LP relaxation
- Duality

This can be used to design approximation algorithms:

- LP rounding
- Primal-Dual Method
- Dual Fitting

Primal-Dual Method and Dual Fitting does not require to solve LPs
$\longrightarrow$ often faster algorithms

next: Randomized Rounding