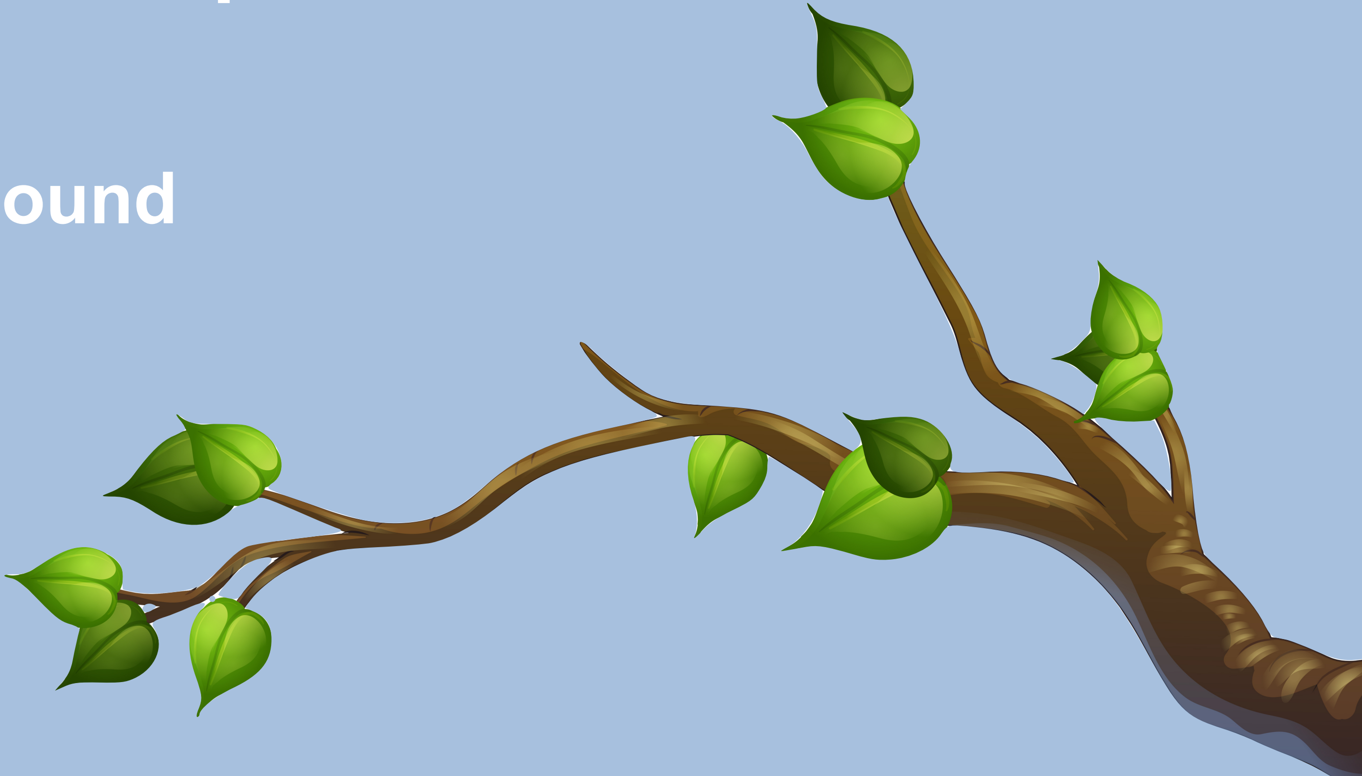
Mathematical Foundations of Computer Graphics & Vision

Branch and Bound

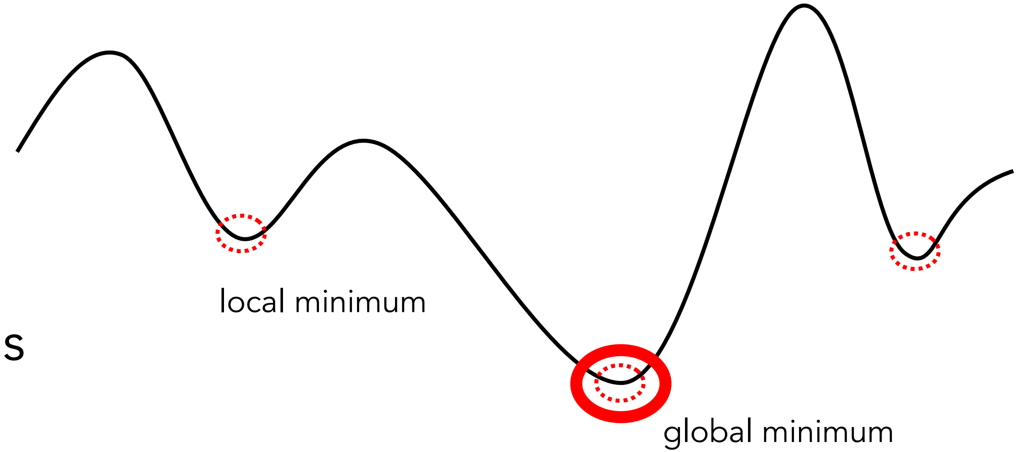


Dr. Baran Gözcü
Computer Graphics Lab
ETH Zürich

Branch and bound

Branch and Bound

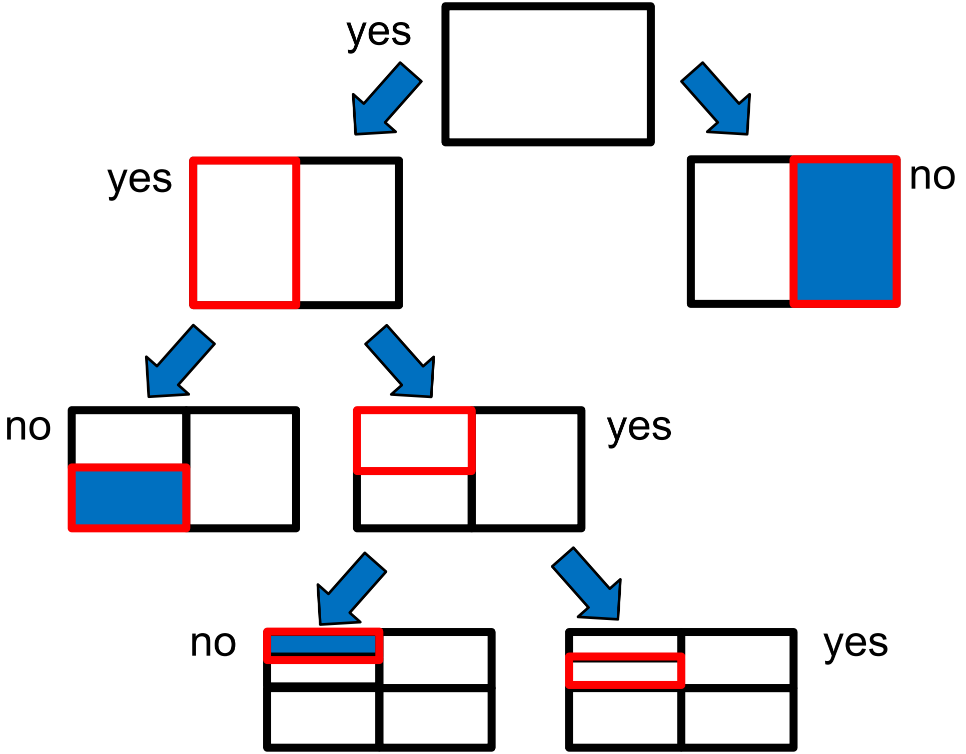
- Algorithm design paradigm
- Discrete optimization problems



Branch and bound

BnB main ideas

- Divide the search space into smaller search spaces
- For each small search space
  - decide if it might contain an optimal (or satisfying) solution
  - Using an upper bound (and/or lower bound)
- Iterate by splitting the remaining spaces



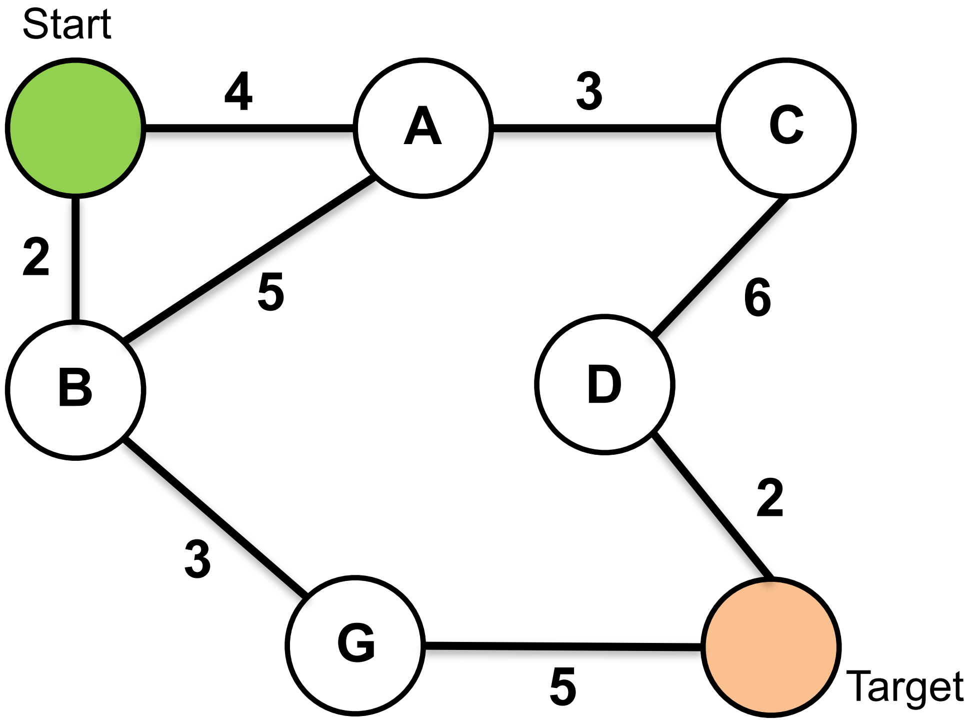
Overview

1. Optimal Path Search with BnB
2. Integer Linear Programming
3. Consensus Set Maximization

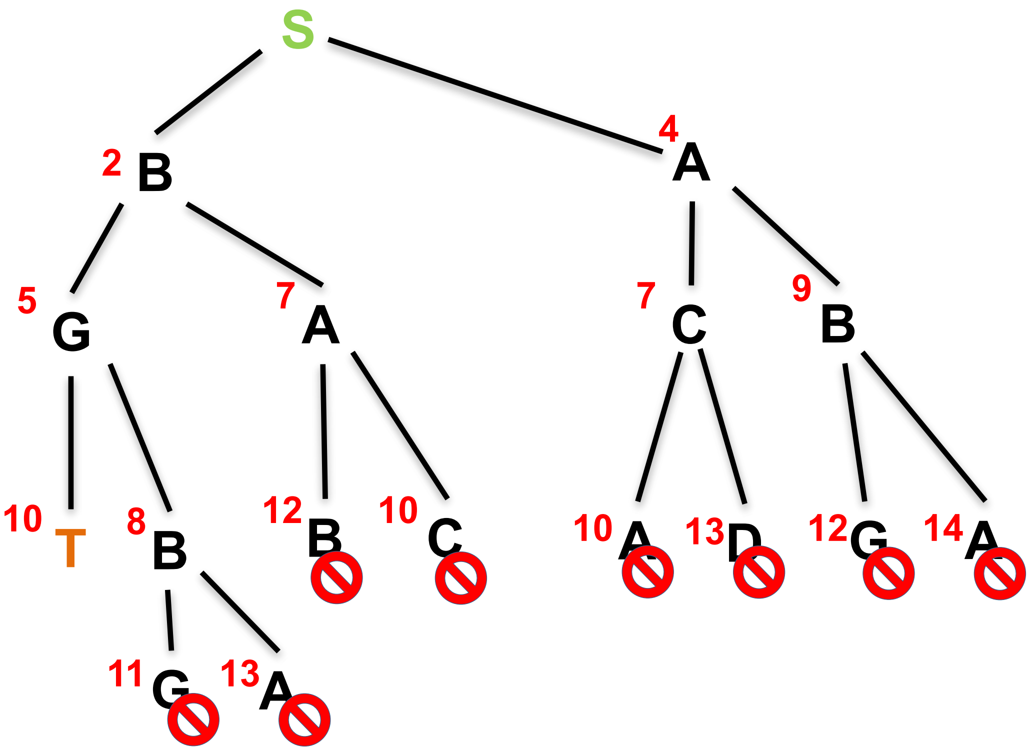
Overview

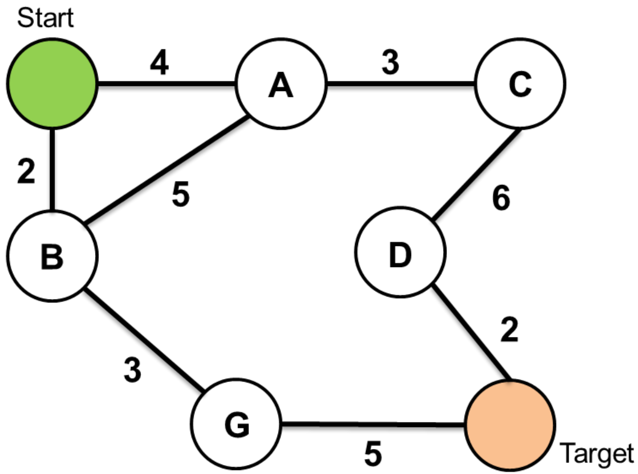
1. Optimal Path Search with BnB
2. Integer Linear Programming
3. Consensus Set Maximization

Optimal Path Search with BnB

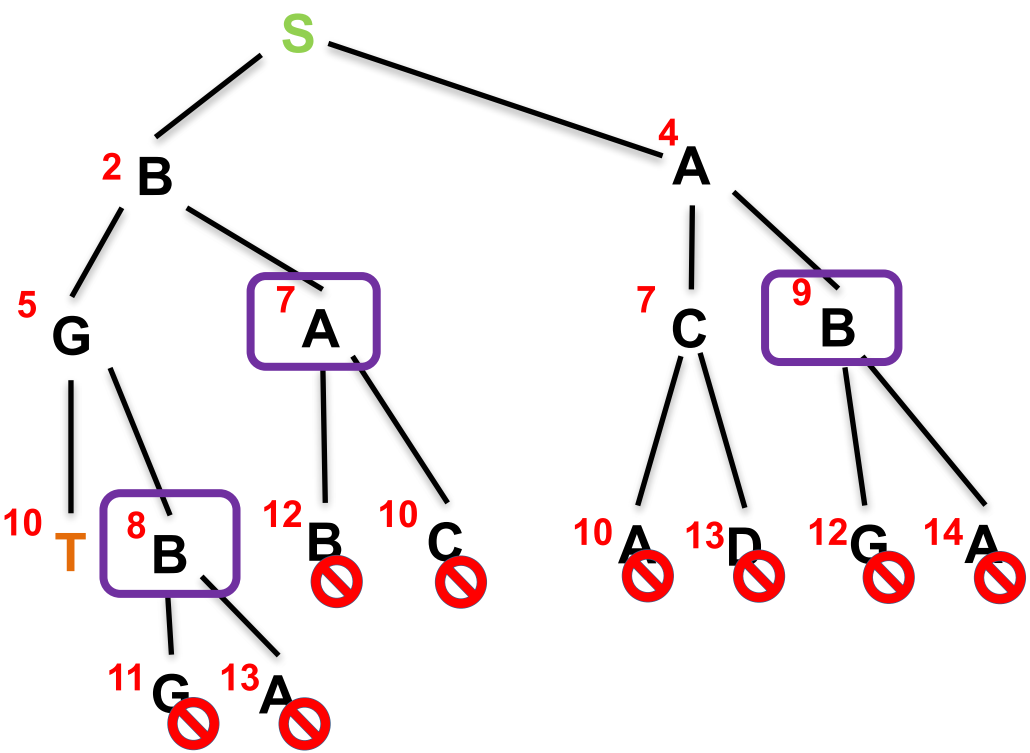


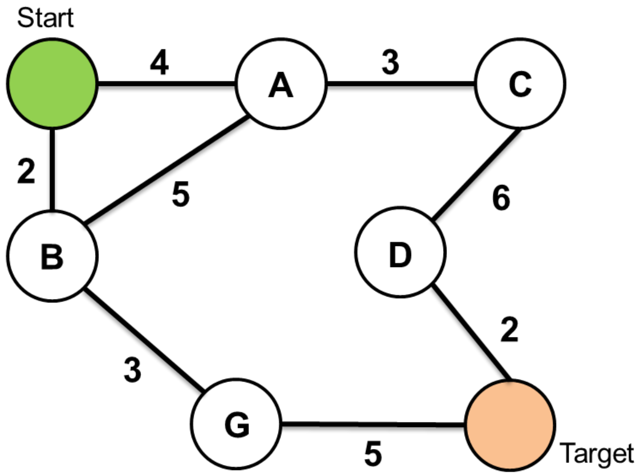
Optimal Path Search with BnB



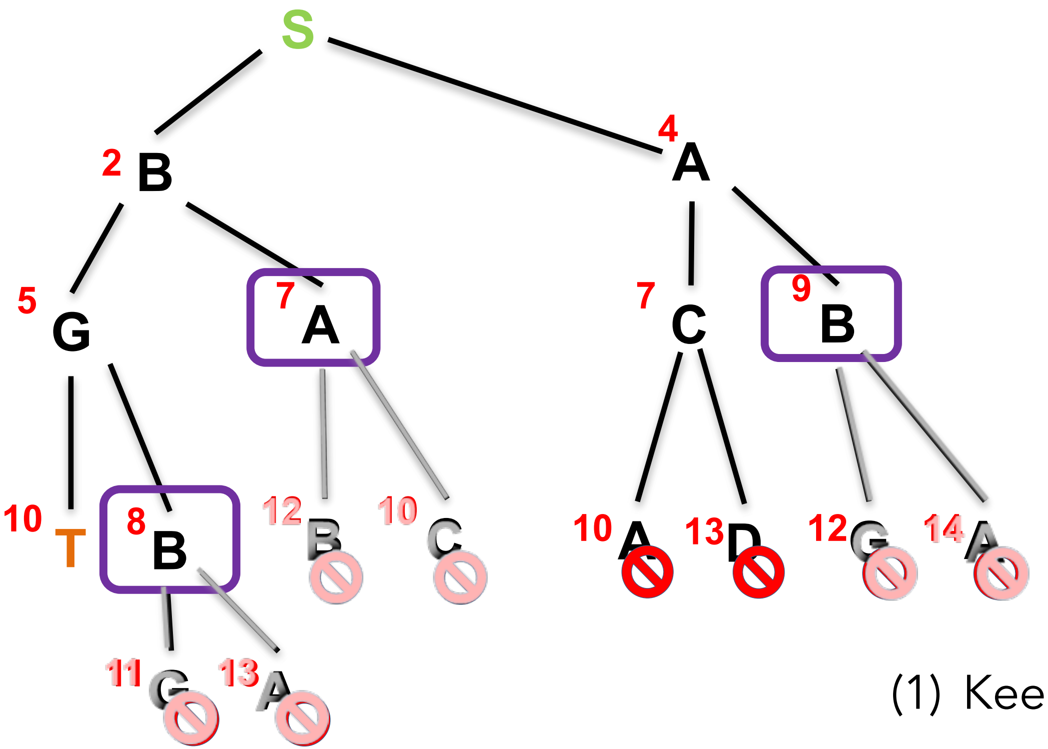


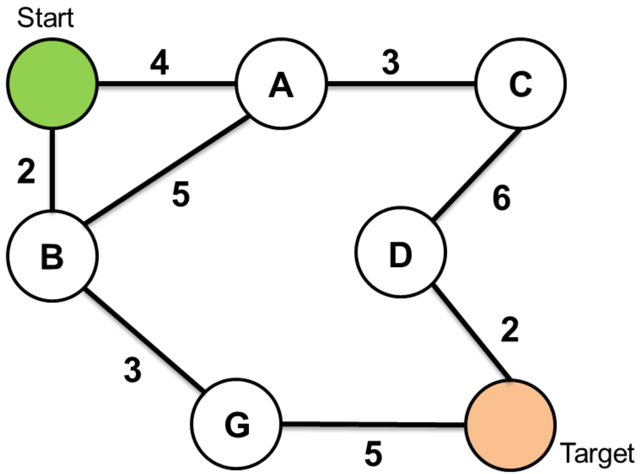
Optimal Path Search – Improvements (1)





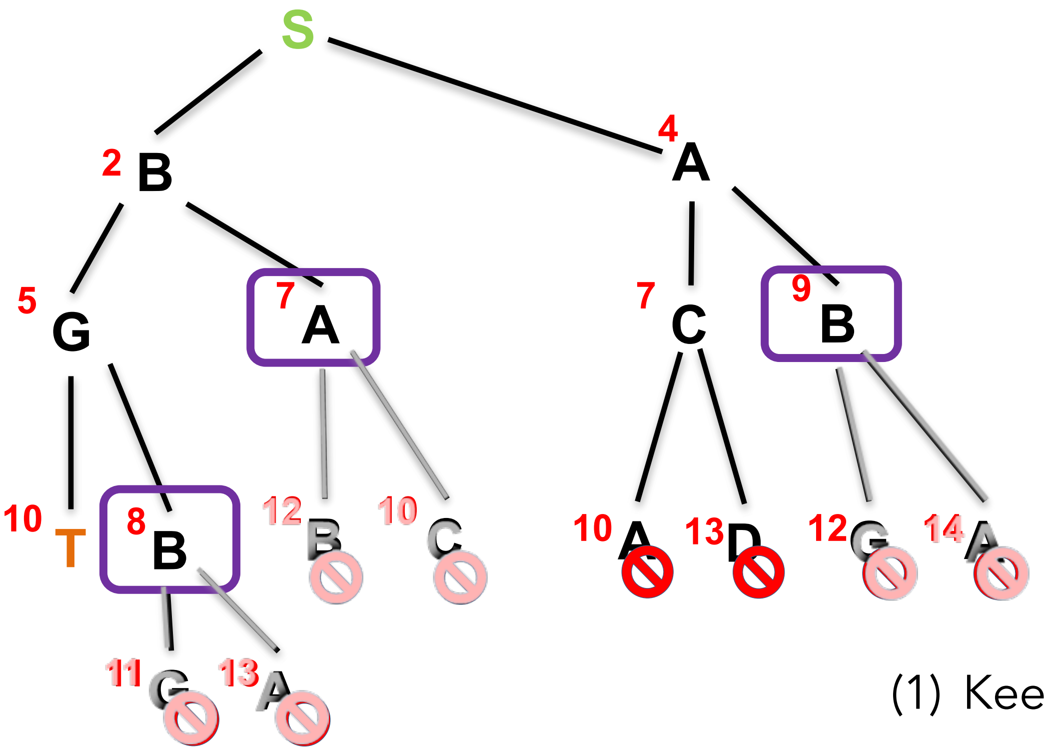
Optimal Path Search – Improvements (1)

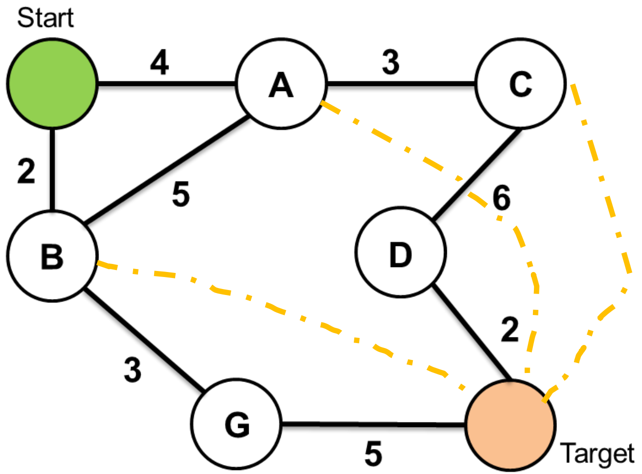




(1) Keep track of already visited nodes

Optimal Path Search – Improvements (2)





A* algorithm is BnB + these improvements

(1) Keep track of already visited nodes

(2) Lower bound estimate for remaining path (bird's eye distance)

Overview

1. Optimal Path Search with BnB

2. Integer Linear Programming

3. Consensus Set Maximization

Linear vs Integer Programming

Our problem

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$

Linear Programming - EASY

$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } 0 \leq x_1 \leq 3,\; 0 \leq x_2 \leq 7,\; 0 \leq x_3 \leq 5,\; 0 \leq x_4 \leq 5$$

$$\text{and } x_i \in \mathbb{R}, \text{ for all } i = 1, \ldots, 4$$

Integer Programming – More difficult

$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$
$$\text{and } 0 \leq x_1 \leq 3,\; 0 \leq x_2 \leq 7,\; 0 \leq x_3 \leq 5,\; 0 \leq x_4 \leq 5$$

$$\text{and } x_i \in \mathbb{Z}, \text{ for all } i = 1, \ldots, 4$$

Naive approach – Exhaustive search

Our problem

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$

$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \leq 15$$

$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60$$

$$\text{and } x_1 + x_2 + x_3 + x_4 \leq 8$$

$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16$$

$$\text{and } x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5]$$

$$\text{and } x_i \in \mathbb{Z}, \text{ for all } i = 1, \ldots, 4$$

$$x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5]$$

$$[0,3] \times [0,7] \times [0,5] \times [0,5]$$

$$\left.\begin{array}{l} x_1=0,\ x_2=0,\ x_3=0,\ x_4=0 \\ x_1=1,\ x_2=0,\ x_3=0,\ x_4=0 \\ x_1=2,\ x_2=0,\ x_3=0,\ x_4=0 \\ \cdots \\ \cdots \\ x_1=3,\ x_2=7,\ x_3=5,\ x_4=5 \end{array}\right\} \text{4*8*6*6=1152 combinations}$$

Solution

$$\boxed{\begin{array}{l} x^*=(0,7,0,1) \\ c^*=23 \end{array}}$$

Exhaustive search is not always possible - Some examples:

$$x_i \in \{0,1\}, \text{ for all } i = 1, \ldots, n$$

2^n combinations
n=10=> 1024
n=20 => 1,048,576
n=30 => 1.0737e+09

$$x_i \in \{0, \ldots, K\}, \text{ for all } i = 1, \ldots, n$$

$(K+1)^n$ combinations – For example with K=10
n=10=> 2.5937e+10
n=20 => 6.7275e+20
n=30 => 1.7449e+31

Naive approach – Round LP Solution

Integer Linear Programming

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \le 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \le 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \le 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \le 16$$
$$\text{and } x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5]$$
$$\text{and } x_i \in \mathbb{Z}, \text{ for all } i = 1, \ldots, 4$$

Relaxation to Linear Programming

$$\max_{\mathbf{x}} \; 2x_1 + 3x_2 + x_3 + 2x_4$$
$$\text{s.t. } 5x_1 + 2x_2 + x_3 + x_4 \le 15$$
$$\text{and } 2x_1 + 6x_2 + 10x_3 + 8x_4 \le 60$$
$$\text{and } x_1 + x_2 + x_3 + x_4 \le 8$$
$$\text{and } 2x_1 + 2x_2 + 3x_3 + 3x_4 \le 16$$
$$\text{and } 0 \le x_1 \le 3, \; 0 \le x_2 \le 7, \; 0 \le x_3 \le 5, \; 0 \le x_4 \le 5$$
$$\boxed{\text{and } x_i \in \mathbb{R}, \text{ for all } i = 1, \ldots, 4}$$

LP

Rounding

ILP solution

ILP solution		Rounded	LP
x*=(0,7,0,1) c*=23	≠	x*=(0,7,0,0) c*=21	x*=(0.08,7,0,0.62) c*=22.4

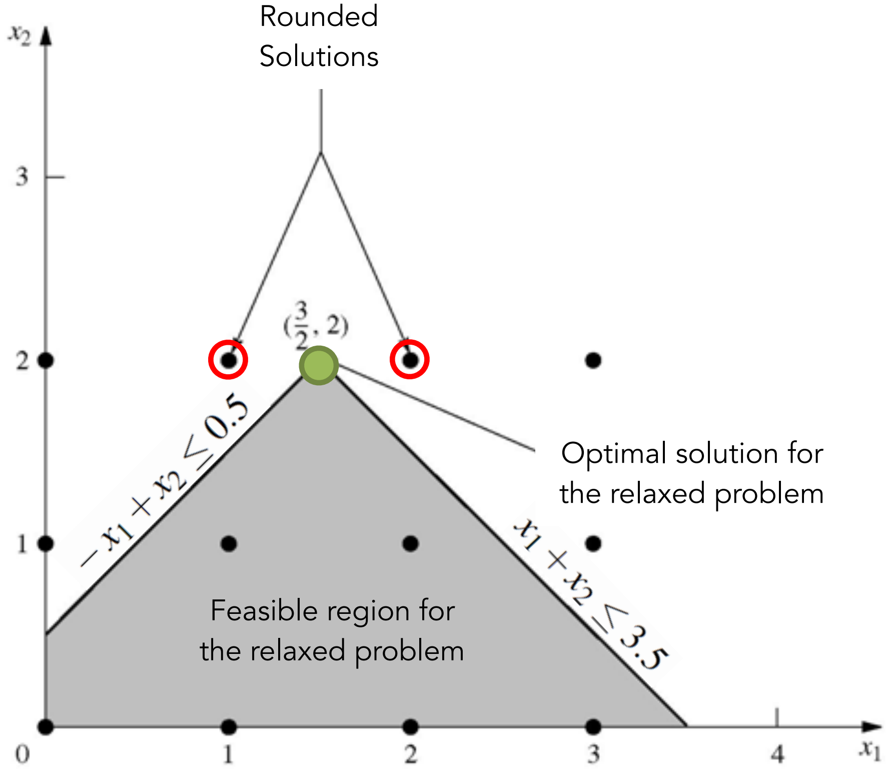
We are lucky this time: Not too far from the optimal solution for this example.

Issues with Rounding the LP Solution

Integer programming *problem 1*

$$
\begin{aligned}
\max_{x_1,x_2} \ & z = x_2 \\
\text{s.t.} \ & -x_1 + x_2 \leq 0.5 \\
& x_1 + x_2 \leq 3.5 \\
& x_1 \geq 0, x_2 \geq 0 \\
& x_i \in \mathbb{Z}, \ \text{for all } i = 1, 2
\end{aligned}
$$

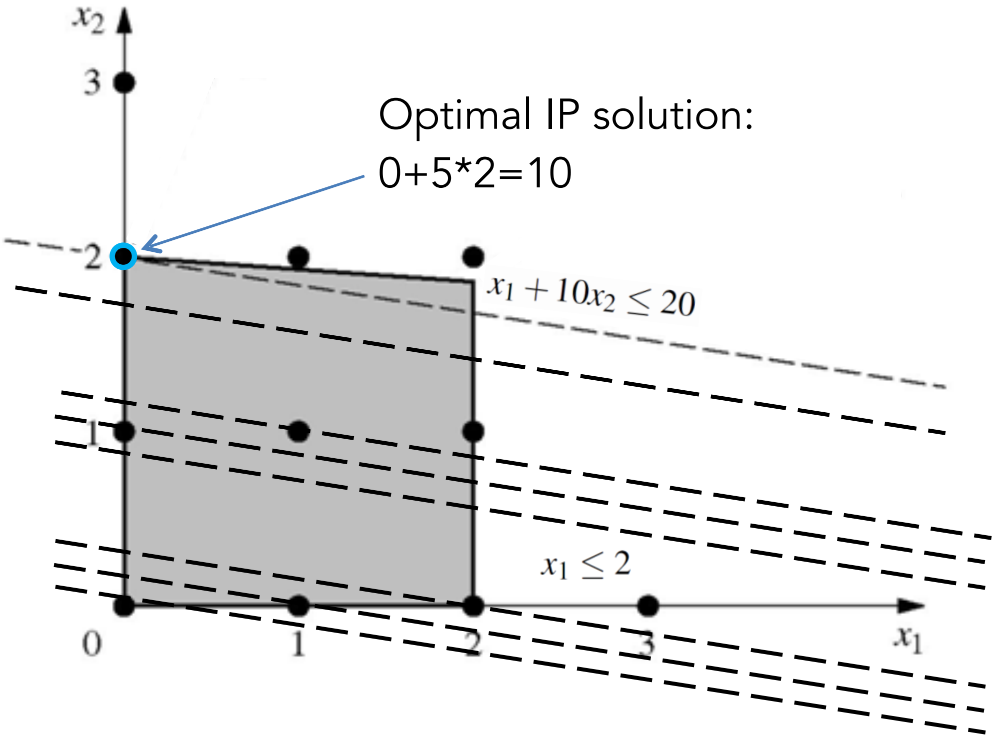
➔ Rounding does not guarantee the **feasibility**



Issues with Rounding the LP Solution

Integer programming *problem 2*

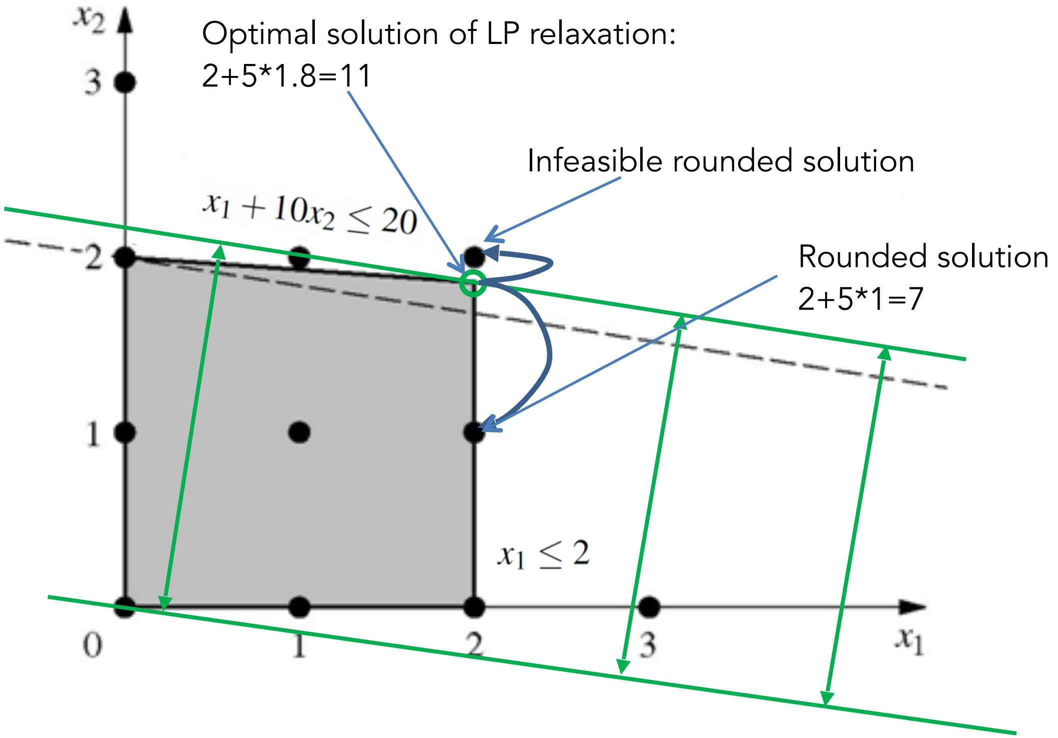
$$\max_{x_1,x_2} \; z = x_1 + 5x_2$$
$$\text{s.t. } x_1 + 10x_2 \leq 20$$
$$x_1 \leq 2$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{Z}, \text{ for all } i = 1,2$$



Issues with Rounding the LP Solution

Integer programming *problem 2*

$$\max_{x_1,x_2} \; z = x_1 + 5x_2$$
$$\text{s.t. } x_1 + 10x_2 \leq 20$$
$$x_1 \leq 2$$
$$x_1 \geq 0, x_2 \geq 0$$
$$x_i \in \mathbb{Z}, \text{ for all } i = 1,2$$



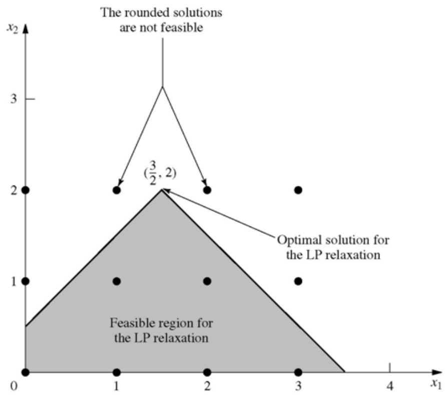
➔ Rounding does not guarantee the **optimality**

Naive approaches – In Summary

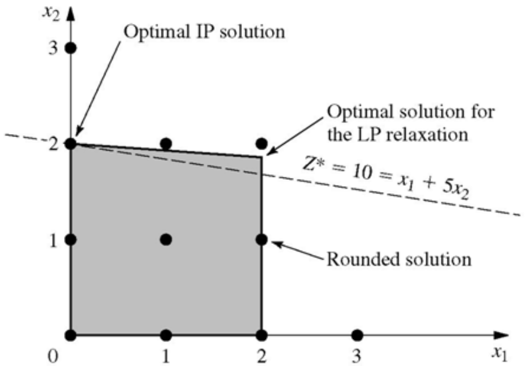
Exhaustive search

- Combinatorial number of solutions to explore
- Exhaustive search is not always possible … but can be good for a small search space

Rounding the LP Solution



➔ Rounding does not guarantee the **feasibility**



➔ Rounding does not guarantee the **optimality**

Branch and bound for IP

Problem statement for

Integer Programming (IP) / Mixed Integer Programming (MIP)

$$\max_{x} c^T x$$

subject to
$$\begin{aligned} Ax &\leq b \\ x_j &\geq 0 \qquad \text{for } j \in N = \{1, \dots, n\} \\ x_j &\in \mathbb{Z} \qquad \text{for } j \in Z \subset N \end{aligned}$$

Set of the indices of the integer variables

$x_j \in N \setminus Z$ are continuous

Branch and bound for IP

For minimization

- **Algorithm:**
- **#1: Initialization.**
  - Set list of problems to $\{P_0\}$.
  - Initialize OPT.
  - Solve LP relaxation of $P_0 \Rightarrow x_*(P_0)$.
  - If $x^*(P_0)$ feasible for P_0, OPT = $c^T x_*(P_0)$ and stop.
- **#2: Problem Selection**
  - Choose a problem P from list whose $x_*(P)$ has $c^T x_*(P_0) <$ OPT . If no such P exists, stop.
- **#3: Variable Selection**
  - Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.
- **#4: Branching**
  - Create two new problems P′ and P″ with $x_p \leq \mathrm{intinf}[x^*_p(P)]$ and $x_p \geq \mathrm{intsup}[x^*_p(P)]$, respectively.
  - Solve continuous relaxations of P′ and P″ $\Rightarrow x^*(P'), x^*(P'')$.
  - **Update OPT**
    - If P′ feasible, $x^*(P')$ feasible for P_0 and $c^T x_*(P') <$ OPT $\Rightarrow$ OPT = $c^T x_*(P')$. Same for P ″ .
  - **Further Inspection**
    - If P′ feasible and $c^T x_*(P') <$ OPT $\Rightarrow$ add P′ to list of problems. Same for P″.
- Afterwards, go back to (2).

Branch and bound for IP

For maximization

- **Algorithm:**
- **#1: Initialization.**
  - Set list of problems to $\{P_0\}$.
  - Initialize OPT.
  - Solve LP relaxation of $P_0 \Rightarrow x_*(P_0)$.
  - If $x^*(P_0)$ feasible for P_0, OPT = $c^T x_*(P_0)$ and stop.
- **#2: Problem Selection**
  - Choose a problem P from list whose $x_*(P)$ has $c^T x_*(P_0) > \text{OPT}$. If no such P exists, stop.
- **#3: Variable Selection**
  - Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.
- **#4: Branching**
  - Create two new problems P′ and P″ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.
  - Solve continuous relaxations of P′ and P″ $\Rightarrow x^*(P'), x^*(P'')$.
  - **Update OPT**
    - If P′ feasible, $x^*(P')$ feasible for P_0 and $c^T x_*(P') > \text{OPT} \Rightarrow \text{OPT} = c^T x_*(P')$. Same for P ″ .
  - **Further Inspection**
    - If P′ feasible and $c^T x_*(P') > \text{OPT} \Rightarrow$ add P′ to list of problems. Same for P″.
- Afterwards, go back to (2).

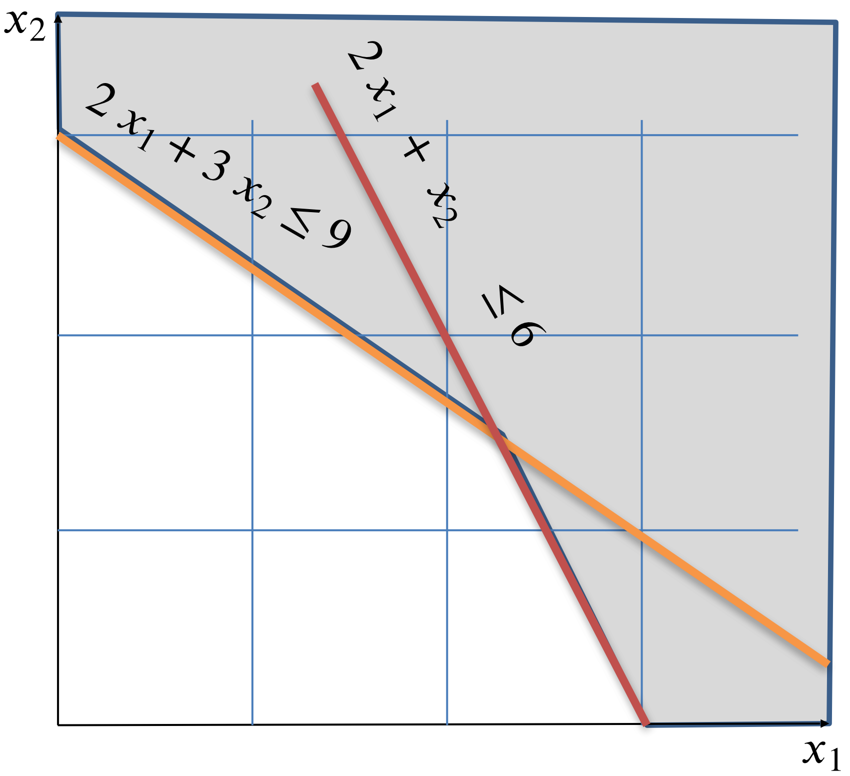
BnB for IP – Example

Maximize $3\,x_1 + 4\,x_2$

s.t $2\,x_1 + x_2 \leq 6$

$2\,x_1 + 3\,x_2 \leq 9$

$x_1,\ x_2 \in \mathbb{N}$



BnB for IP – Example

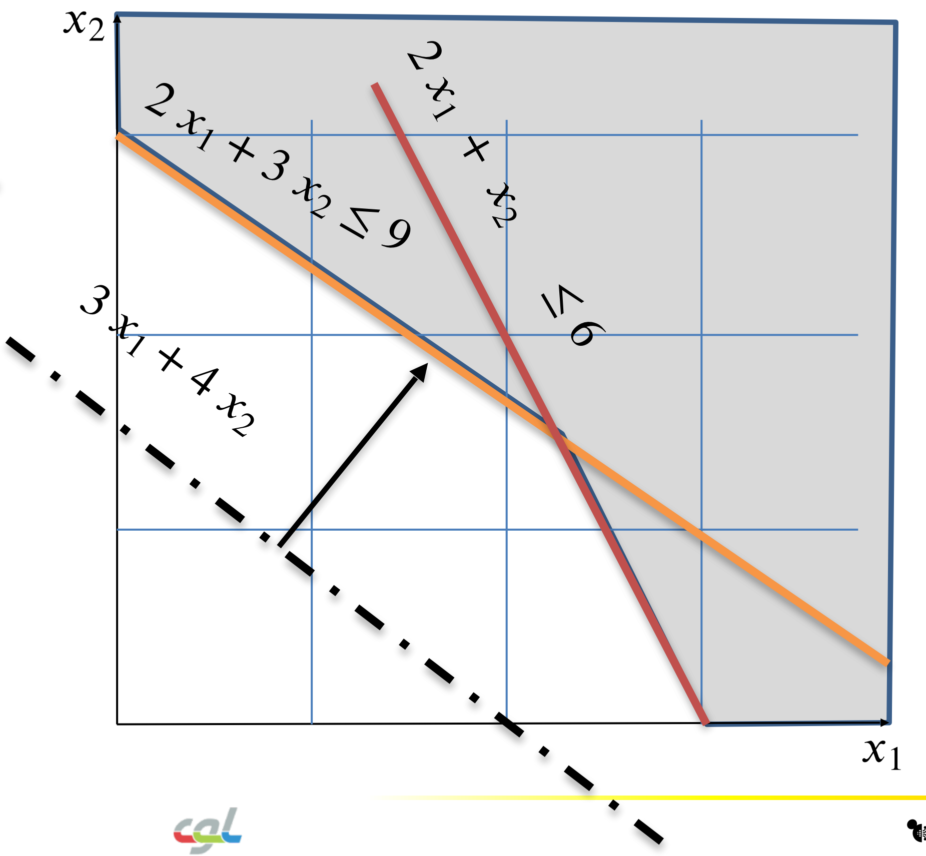
Maximize $3\,x_1 + 4\,x_2$

s.t $2\,x_1 + x_2 \leq 6$

$2\,x_1 + 3\,x_2 \leq 9$

~~$x_1, x_2 \in \mathbb{N}$~~

Solving the relaxed problem



BnB for IP – Example

Maximize $3\,x_1 + 4\,x_2$

s.t $2\,x_1 + x_2 \leq 6$

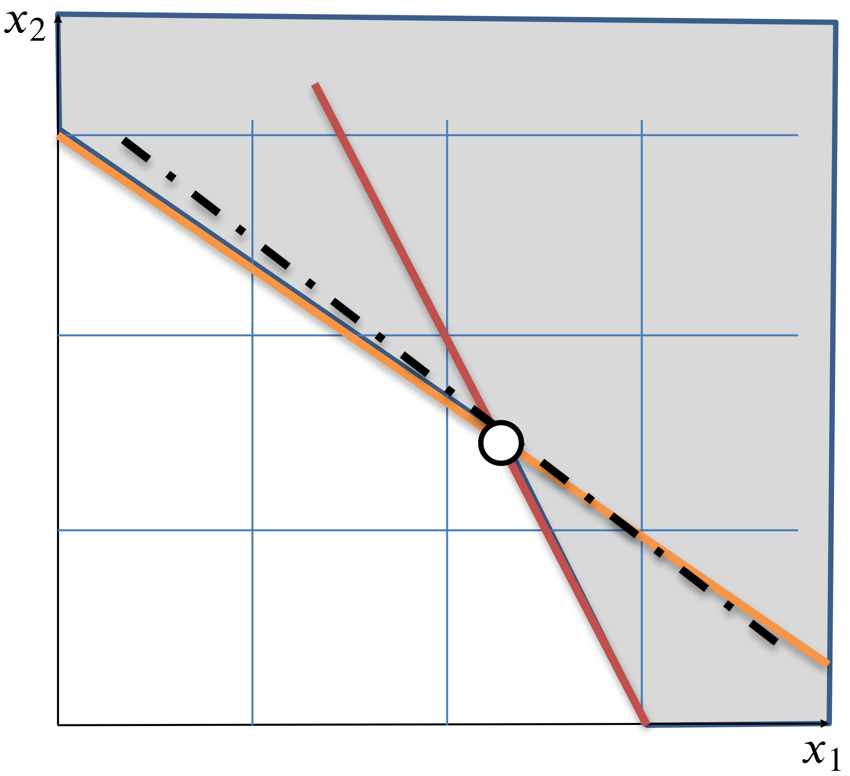
$2\,x_1 + 3\,x_2 \leq 9$

$\cancel{x_1,\ x_2 \in \mathbb{N}}$

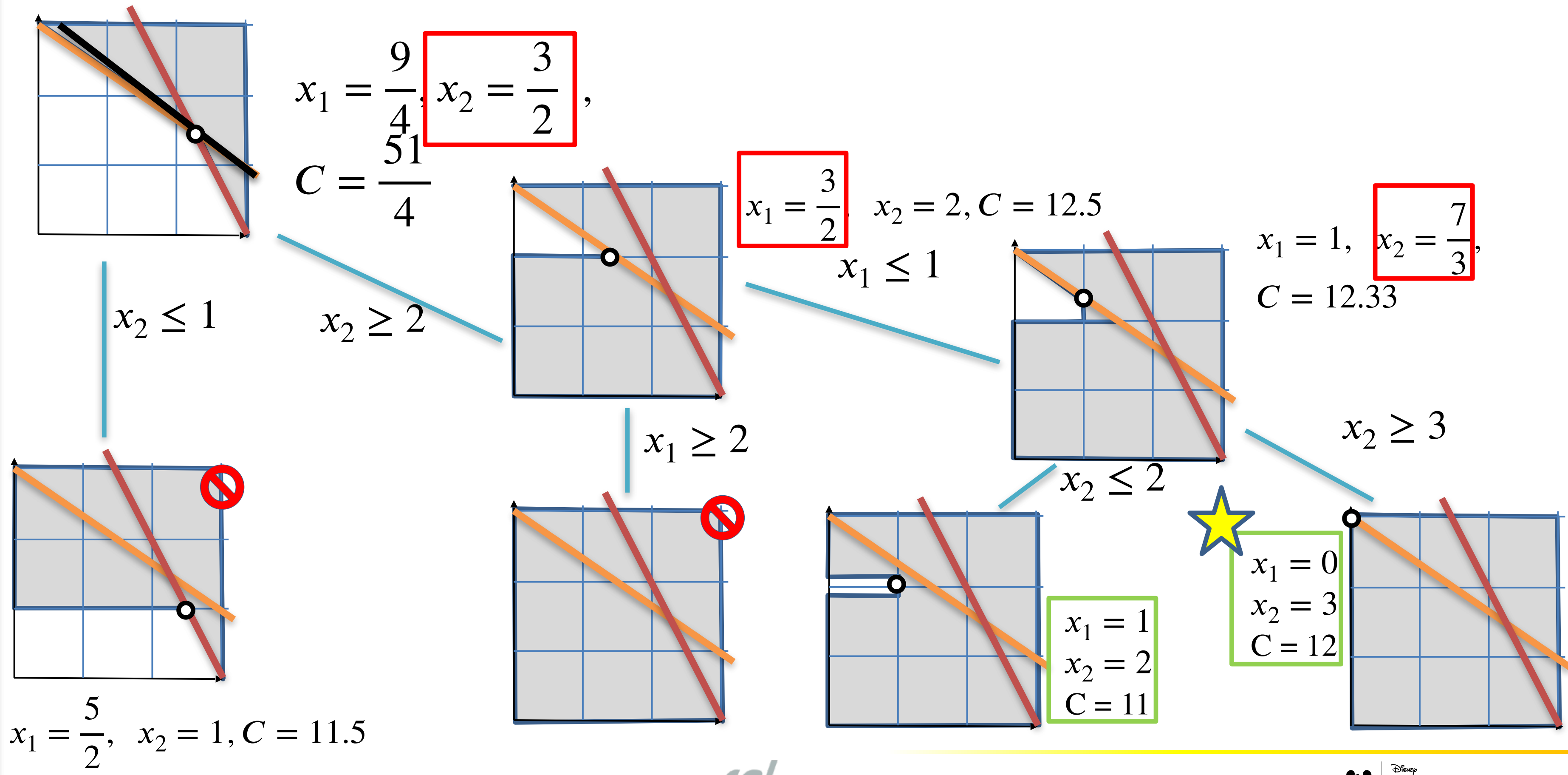
Solving the relaxed problem

Solution $x_1 = \dfrac{9}{4}$ $x_2 = \dfrac{3}{2}$

Objective $C = \dfrac{51}{4}$



BnB for IP – Example



Additional Example

B&B example

$$
\begin{aligned}
\max_{\mathbf{x}} \;& 2x_1 + 3x_2 + x_3 + 2x_4 \\
\text{s.t. } & 5x_1 + 2x_2 + x_3 + x_4 \leq 15 \\
\text{and } & 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60 \\
\text{and } & x_1 + x_2 + x_3 + x_4 \leq 8 \\
\text{and } & 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16 \\
\text{and } & x_1 \in [0,3], x_2 \in [0,7], x_3 \in [0,5], x_4 \in [0,5] \\
\text{and } & x_i \in \mathbb{Z}, \text{ for all } i = 1, \ldots, 4
\end{aligned}
$$

Original (integer) system

$\Rightarrow$

$$
\begin{aligned}
\max_{\mathbf{x}} \;& 2x_1 + 3x_2 + x_3 + 2x_4 \\
\text{s.t. } & 5x_1 + 2x_2 + x_3 + x_4 \leq 15 \\
\text{and } & 2x_1 + 6x_2 + 10x_3 + 8x_4 \leq 60 \\
\text{and } & x_1 + x_2 + x_3 + x_4 \leq 8 \\
\text{and } & 2x_1 + 2x_2 + 3x_3 + 3x_4 \leq 16 \\
\text{and } & 0 \leq x_1 \leq 3,\; 0 \leq x_2 \leq 7,\; 0 \leq x_3 \leq 5,\; 0 \leq x_4 \leq 5 \\
\text{and } & x_i \in \mathbb{R}, \text{ for all } i = 1, \ldots, 4
\end{aligned}
$$

Relaxed system

B&B example

#1: Initialization.

Set list of problems to $\{P_0\}$.

Initialize OPT.

- objective function value of best

Solve LP relaxation of $P_0 \Rightarrow x^*(P_0)$.

If $x^*(P_0)$ feasible for P_0, OPT = $c^T x^*(P_0)$ and stop.

$x^*(P_0)=(0.08,7,0,0.62)$

$c^*(P_0)=22.4$

P_0

$$
\begin{aligned}
\max_{\mathbf{x}}\ & 2x_1+3x_2+x_3+2x_4 \\
\text{s.t. } & 5x_1+2x_2+x_3+x_4 \le 15 \\
\text{and } & 2x_1+6x_2+10x_3+8x_4 \le 60 \\
\text{and } & x_1+x_2+x_3+x_4 \le 8 \\
\text{and } & 2x_1+2x_2+3x_3+3x_4 \le 16 \\
\text{and } & 0 \le x_1 \le 3,\ 0 \le x_2 \le 7,\ 0 \le x_3 \le 5,\ 0 \le x_4 \le 5 \\
\text{and } & x_i \in \mathbb{R},\ \text{for all } i=1,\dots,4
\end{aligned}
$$

Problem list = $\{P_0\}$

Opt=$-\infty$

B&B example

#2: Problem Selection

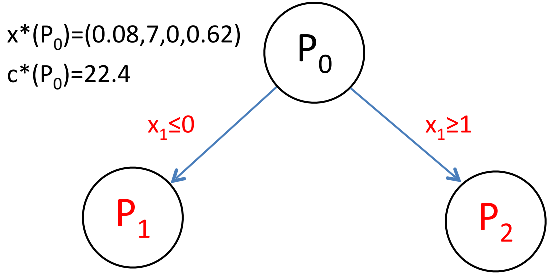
Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0)$ **>** OPT. If no such P exists, stop.

#3: Variable Selection

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.



Problem list = $\{P_1, P_2\}$

Opt=$-\infty$

B&B example

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.

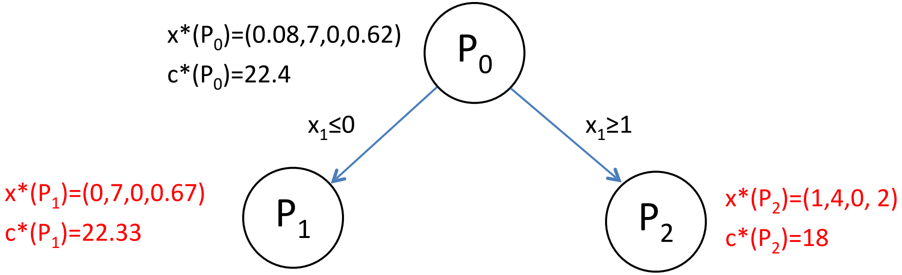
Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

Update OPT

If P′ feasible, $x^*(P')$ feasible for P_0 and $c^T x^*(P')$ **>** OPT ⇒ OPT = $c^T x^*(P')$. Same for P ′′ .

Further Inspection

If P′ feasible and $c^T x^*(P')$ **>** OPT ⇒ add P′ to list of problems. Same for P′′.



Problem list = $\{P_1\}$

Opt=18

B&B example

#2: Problem Selection

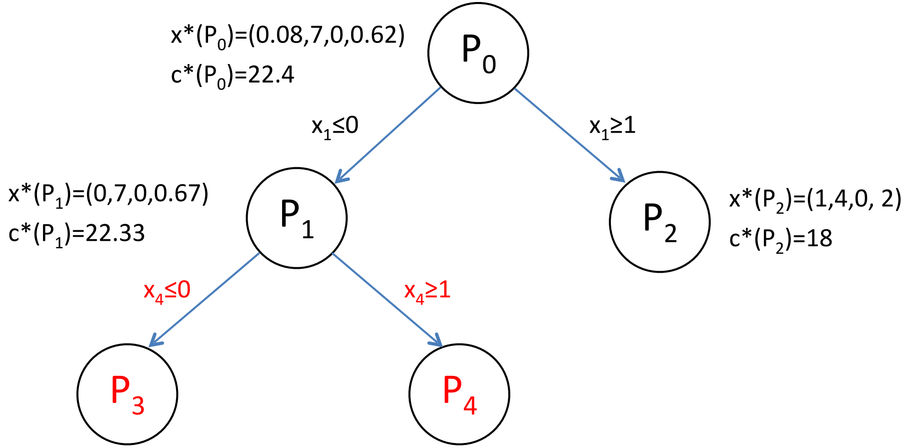
Choose a problem P from list whose $x_*(P)$ has $c^T x_*(P_0)$ **>** OPT. If no such P exists, stop.

#3: Variable Selection

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.



Problem list = $\{P_3, P_4\}$

Opt=18

B&B example

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \mathrm{intinf}[x^*_p(P)]$ and $x_p \geq \mathrm{intsup}[x^*_p(P)]$, respectively.

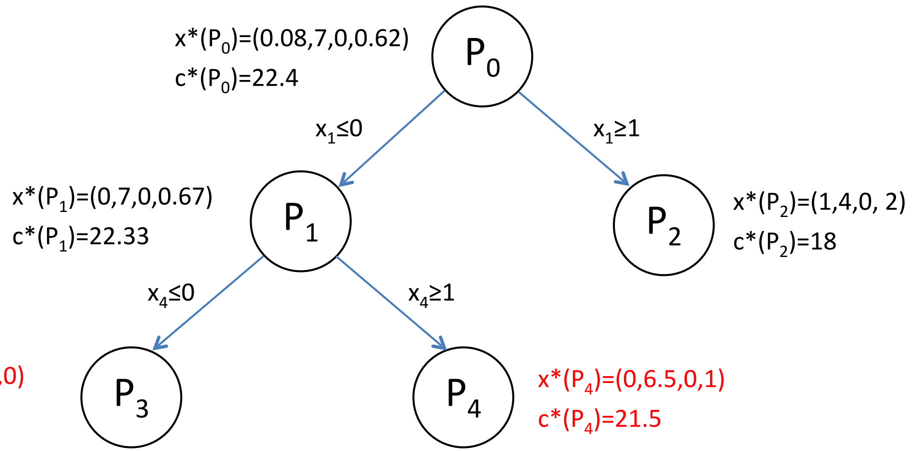
Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

Update OPT

If P′ feasible, $x^*(P')$ feasible for P_0 and $c^T x^*(P')$ > OPT ⇒ OPT = $c^T x^*(P')$. Same for P ′′ .

Further Inspection

If P′ feasible and $c^T x^*(P')$ > OPT ⇒ add P′ to list of problems. Same for P′′.



Problem list = $\{P_3,P_4\}$

Opt=18

B&B example

#2: Problem Selection

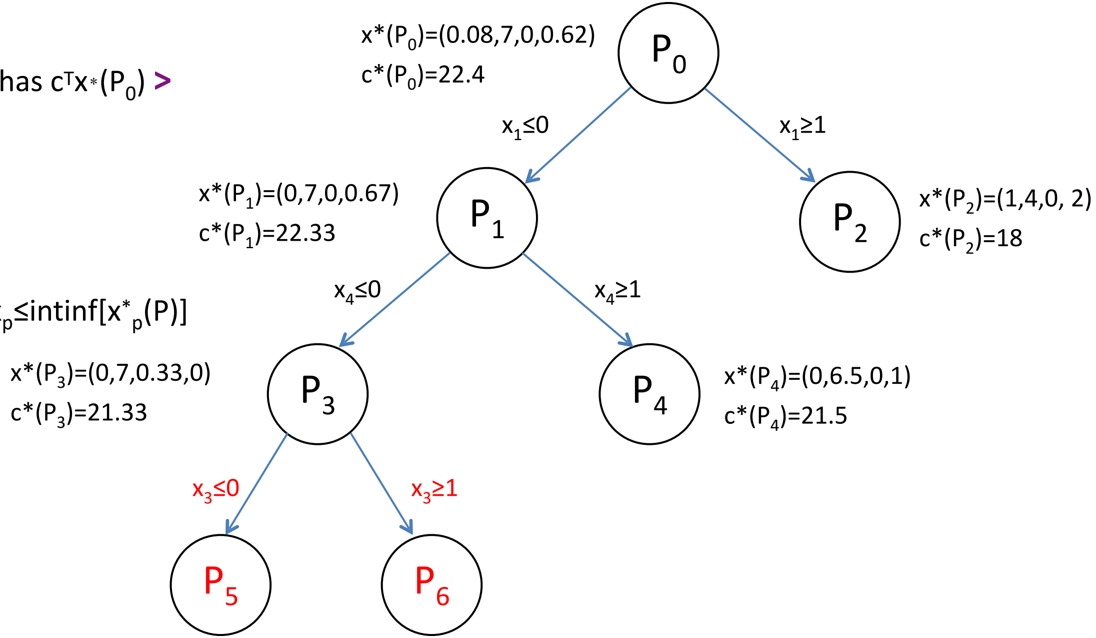
Choose a problem P from list whose $x^*(P)$ has $c^T x^*(P_0)$ > OPT. If no such P exists, stop.

#3: Variable Selection

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.



Problem list = $\{P_4, P_5, P_6\}$

Opt=18

B&B example

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.

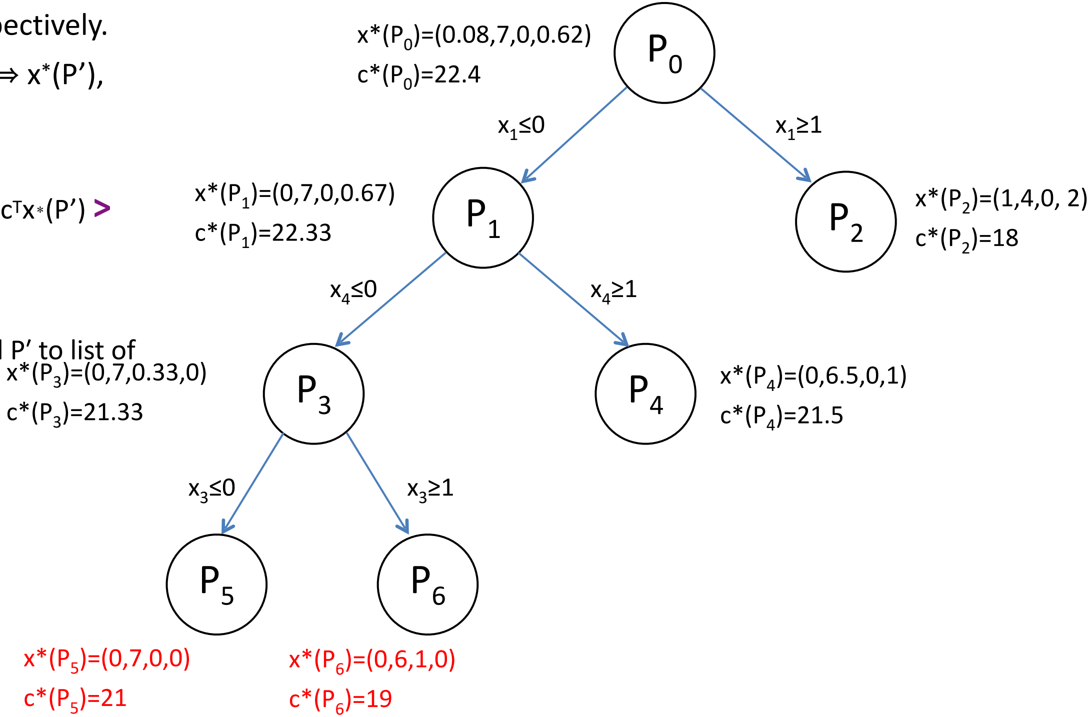
Solve continuous relaxations of P′ and P′′ ⇒ x*(P′), x*(P′′).

Update OPT

If P′ feasible, x*(P′) feasible for P_0 and $c^T x^*(P') >$ OPT ⇒ OPT = $c^T x^*(P')$. Same for P ′′ .

Further Inspection

If P′ feasible and $c^T x^*(P') >$ OPT ⇒ add P′ to list of problems. Same for P′′.



Problem list = $\{P_4\}$

Opt=21

B&B example

#2: Problem Selection

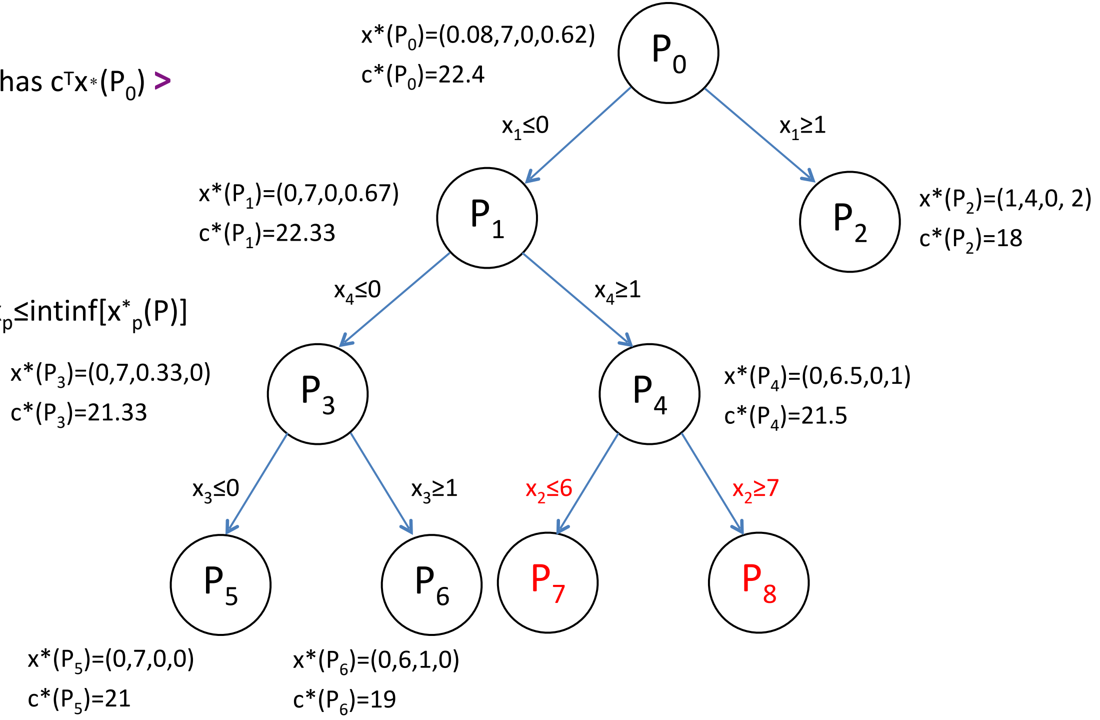
Choose a problem P from list whose $x^*(P)$ has $c^Tx^*(P_0)$ > OPT. If no such P exists, stop.

#3: Variable Selection

Choose $x_p \in Z$ with $x^*_p(P) \notin Z$.

#4: Branching

Create two new problems P′ and P′′ with $x_p \leq \text{intinf}[x^*_p(P)]$ and $x_p \geq \text{intsup}[x^*_p(P)]$, respectively.



Problem list = $\{P_7,P_8\}$

Opt=21

B&B example

#4: Branching

Create two new problems P′ and P′′ with $x_p \le \mathrm{intinf}[x^*_p(P)]$ and $x_p \ge \mathrm{intsup}[x^*_p(P)]$, respectively.

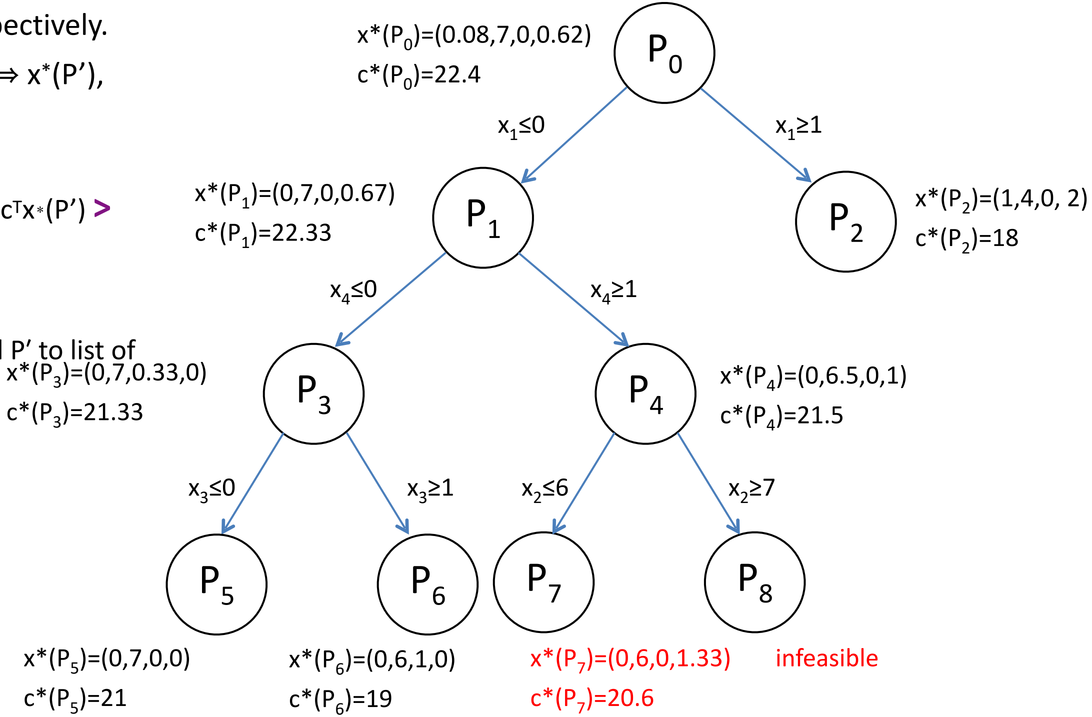
Solve continuous relaxations of P′ and P′′ ⇒ $x^*(P')$, $x^*(P'')$.

Update OPT

If P′ feasible, $x^*(P')$ feasible for P_0 and $c^T x^*(P')$ > OPT ⇒ OPT = $c^T x^*(P')$. Same for P ′′ .

Further Inspection

If P′ feasible and $c^T x^*(P')$ > OPT ⇒ add P′ to list of problems. Same for P′′.



Problem list = {}

Opt=21

$x^*=(0,7,0,0)$

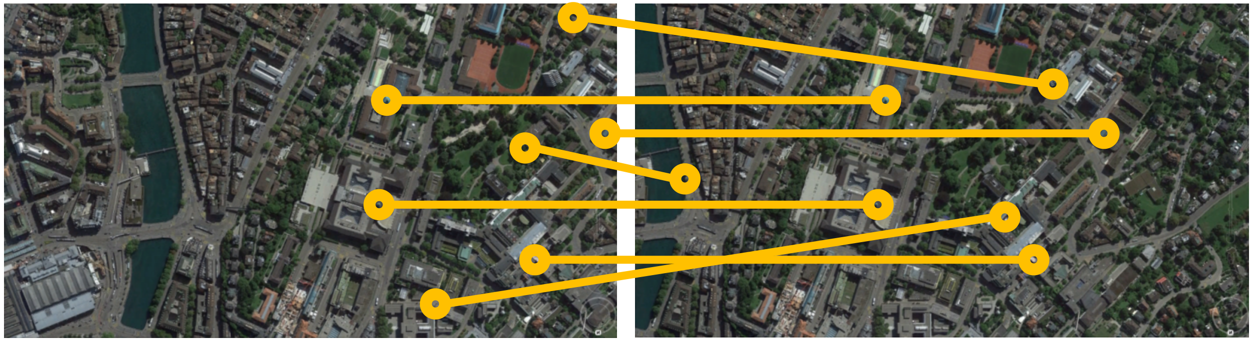
Overview

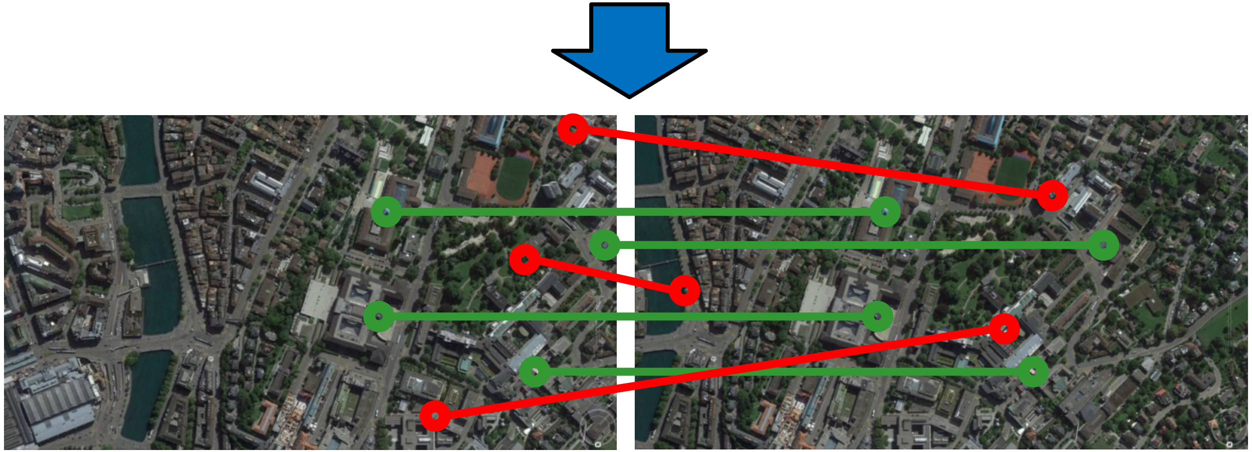
– Optimal path search with BnB

– Integer Linear Programming

– Consensus Set Maximization

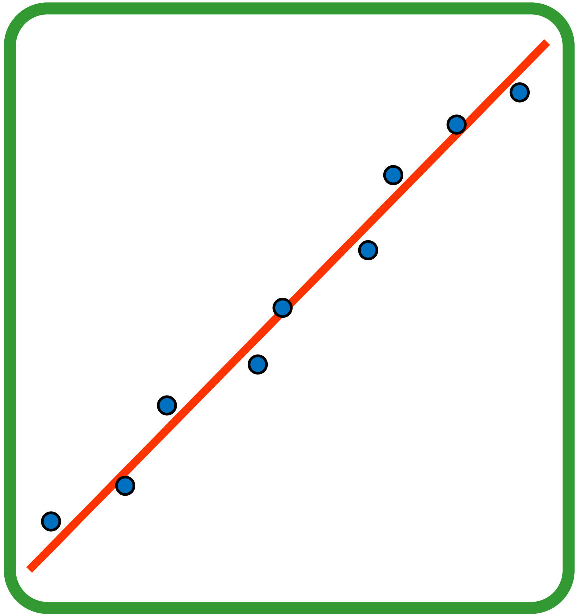
BnB – simple bounds



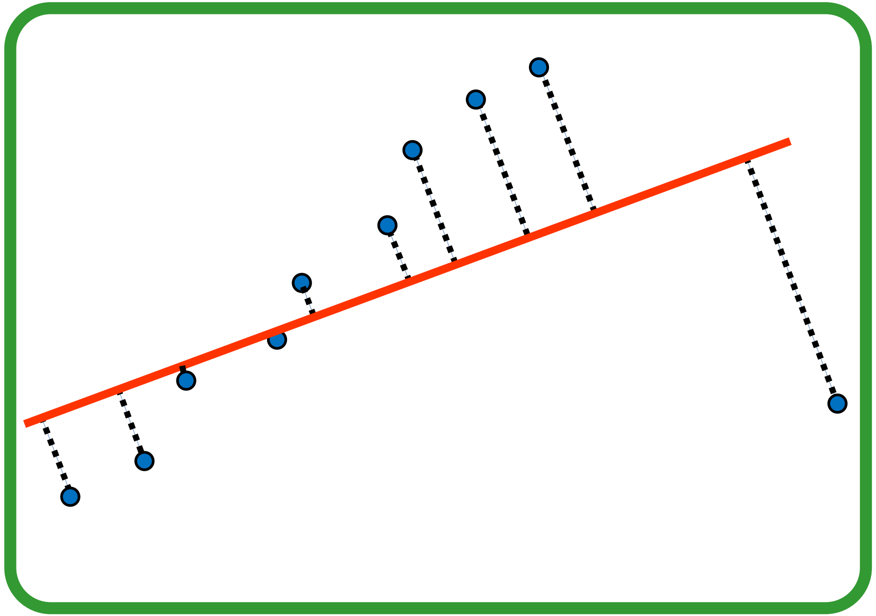


From Google Earth views

BnB - a motivation example

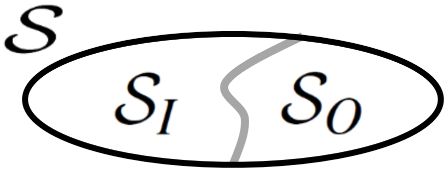


Only **inliers**



Effects due to a single **outlier**

Mathematical formulation

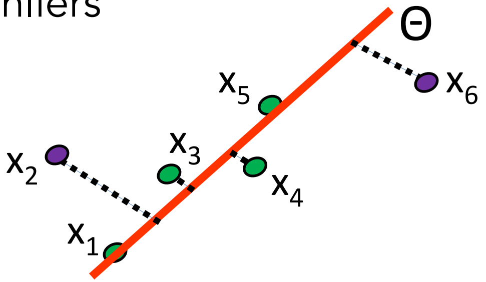


Input: data S

- We note x_i is the i-th data point

Output: model Θ and inliers/outliers ($\mathcal{S}_I$,$\mathcal{S}_O$)

- More precisely, the "model leading to the highest nb of inliers"



What are the unknowns?

- The model Θ
- The set of inliers/outliers ($\mathcal{S}_I$, $\mathcal{S}_O$)

Cost: to distinguish inliers/outliers

- An inlier has a "small" cost, *i.e.* $f(\Theta, x_i) \leq T$

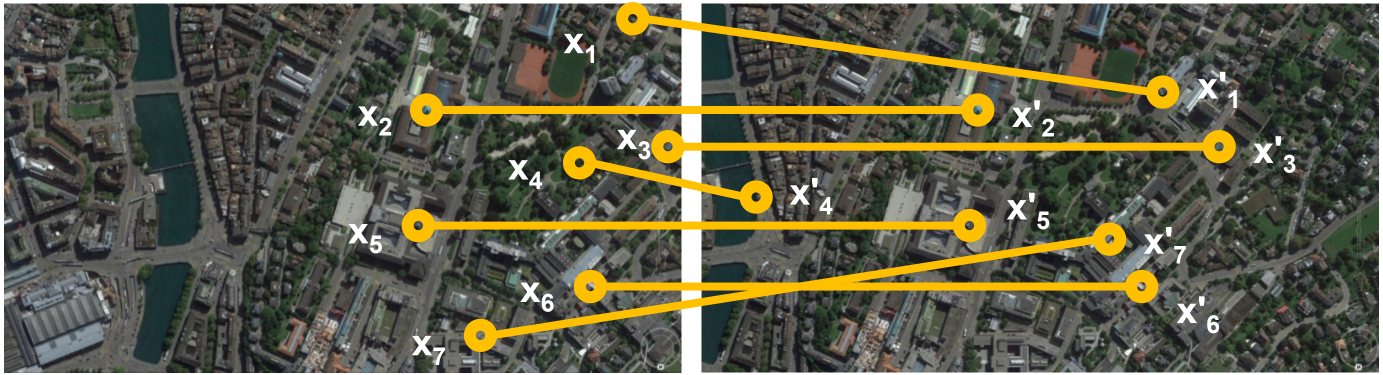
$$\max_{\mathcal{S}_{\mathrm{I}}, \Theta} \quad \mathbf{card}\ (\mathcal{S}_{\mathrm{I}})$$

maximize the nb of inliers

$$\text{s.t.} \qquad f(\Theta, x_i) \leq T, \forall i \in \mathcal{S}_{\mathrm{I}}$$

all the inliers must have a "small" error

BnB – simple bounds



Motion model (1D translation) $x'_i = x_i + T$

Inlier definition $d(x_i, x'_i) \leq \varepsilon$ $\xrightarrow[\text{bounds}]{\text{Resulting model}}$ $T_l \leq T \leq T_u$

Example

Inlier Compatibility Test

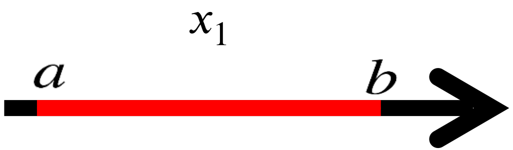
x′ = 15, x = 9, $\varepsilon = 2$ → $4 \leq T \leq 8$

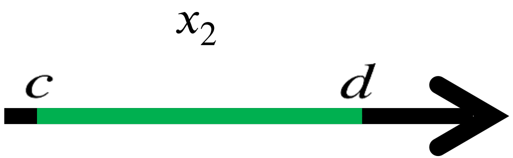
$-1 \leq T \leq 3$ FAIL!

$-1 \leq T \leq 6$

$-1 \leq T \leq 4$

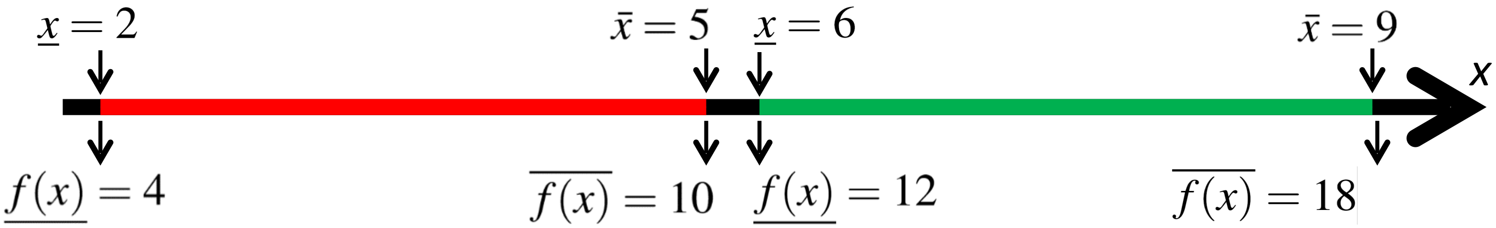
BnB – Interval analysis





$$
\begin{aligned}
x_1 + x_2 \quad & [a,b] + [c,d] = [a+c, b+d] \\
x_1 - x_2 \quad & [a,b] - [c,d] = [a-d, b-c] \\
x_1 \,.\, x_2 \quad & [a,b] \times [c,d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \\
x_1 / x_2 \quad & [a,b] / [c,d] = [\min(\tfrac{a}{c}, \tfrac{a}{d}, \tfrac{b}{c}, \tfrac{b}{d}), \max(\tfrac{a}{c}, \tfrac{a}{d}, \tfrac{b}{c}, \tfrac{b}{d})]
\end{aligned}
$$

$$f(x) = 2x$$

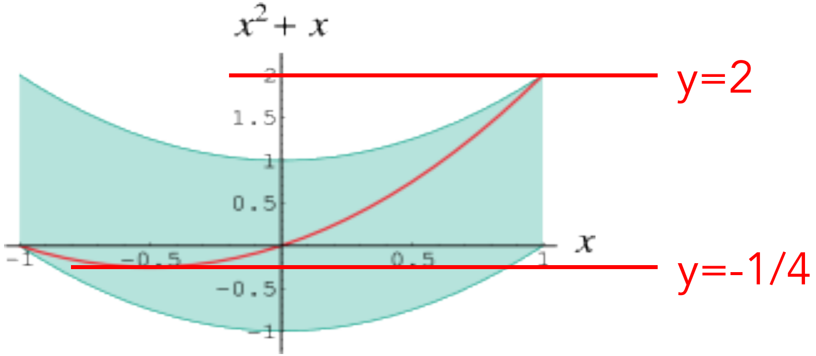


BnB – Interval analysis

For odd n: $[a,b]^n = [a^n, b^n]$

For even n:

$$[a,b]^n = \begin{cases} [a^n, b^n] \text{ if } a \geq 0 \\ [b^n, a^n] \text{ if } b < 0 \\ [0, \max(a^n, b^n)] \text{ otherwise} \end{cases}$$



Example

$$f(x) = x^2 + x \qquad x \in [a,b] = [-1,1]$$

$$[-1,1]^2 + [-1,1] = [0, \max(a^n, b^n)] + [-1,1] = [0,1] + [-1,1] = [-1,2]$$

Let's rewrite the function

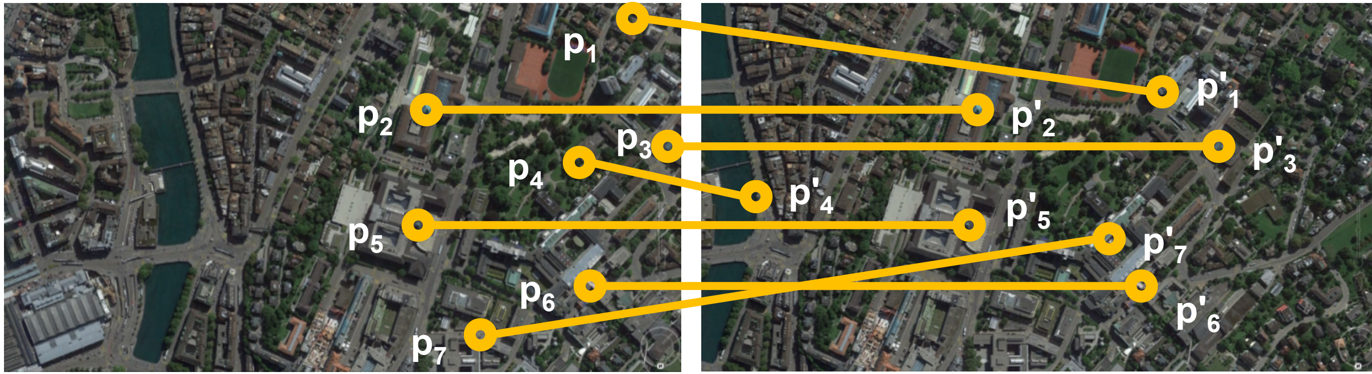
$$f(x) = x^2 + x \quad \Longrightarrow \quad f(x) = (x + \tfrac{1}{2})^2 - \tfrac{1}{4}$$

$$([-1,1] + \tfrac{1}{2})^2 - \tfrac{1}{4} = [-\tfrac{1}{2}, \tfrac{3}{2}]^2 - \tfrac{1}{4} = [0, \tfrac{9}{4}] - \tfrac{1}{4} = \boxed{[-\tfrac{1}{4}, 2]}$$

We obtain tighter bound

BnB – simple bounds

$p_1=(950\,,\,20)$	$p'_1=(700\,,\,130)$
$p_2=(640\,,\,160)$	$p'_2=(420\,,\,160)$
$p_3=(1000\,,\,210)$	$p'_3=(780\,,\,210)$
$p_4=(875\,,\,245)$	$p'_4=(400\,,\,480)$
$p_5=(630\,,\,330)$	$p'_5=(410\,,\,330)$
$p_6=(890\,,\,430)$	$p'_6=(670\,,\,430)$
$p_7=(725\,,\,500)$	$p'_7=(620\,,\,350)$



Motion model (2D translation)

$$(x'_i, y'_i) = (x_i, y_i) + (\boldsymbol{T}_x, \boldsymbol{T}_y)$$

Inlier definition

$$d(x_i, x'_i) = |x'_i - (x_i + \boldsymbol{T}_x)| \leq \varepsilon$$

$$d(y_i, y'_i) = |y'_i - (y_i + \boldsymbol{T}_y)| \leq \varepsilon$$

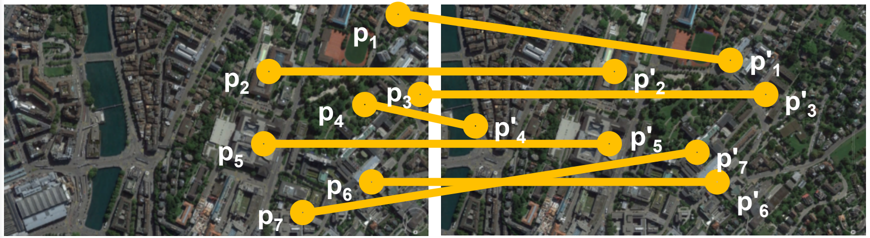
$$\varepsilon = 10 \text{ pixels}$$

Questions

→ Compute upper bound ?

→ Compute lower bound ?

BnB – simple bounds

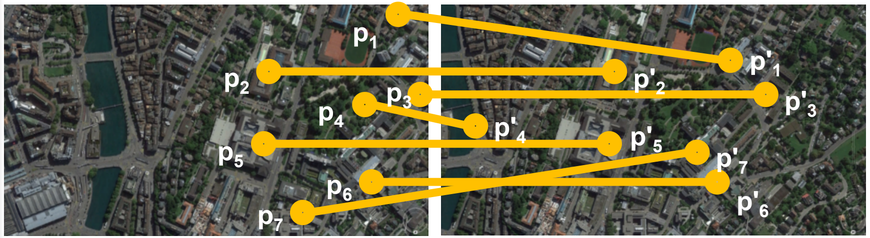


Upper bound

Lower bound

		T_x	T_y
p_1=(950, 20)	p'_1=(700 , 130)	-250 [-260, -240]	110 [100, 120]
p_2=(640 , 160)	p'_2=(420 , 160)	-220 [-230, -210]	0 [-10, 10]
p_3=(1000 , 210)	p'_3=(780 , 210)	-220 [-230, -210]	0 [-10, 10]
p_4=(875 , 245)	p'_4=(400 , 480)	-475 [-485, -465]	235 [225, 245]
p_5=(630 , 330)	p'_5=(410 , 330)	-220 [-230, -210]	0 [-10, 10]
p_6=(890 , 430)	p'_6=(670 , 430)	-220 [-230, -210]	0 [-10, 10]
p_7=(725 , 500)	p'_7=(620 , 350)	-105 [-115, -95]	-150 [-160, -140]

BnB – simple bounds

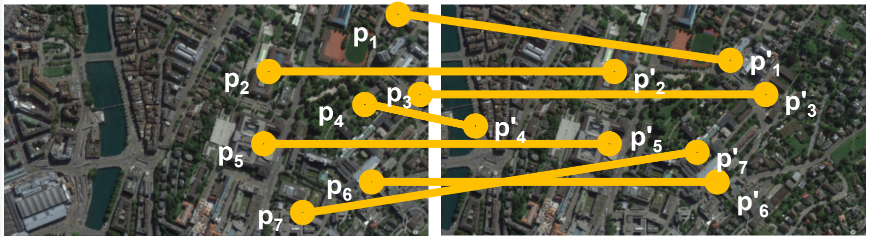


		T_x	T_y
p_1=(950, 20)	p'_1=(700 , 130)	-250 [-260, -240]	110 [100, 120]
p_2=(640 , 160)	p'_2=(420 , 160)	-220 [-230, -210]	0 [-10, 10]
p_3=(1000 , 210)	p'_3=(780 , 210)	-220 [-230, -210]	0 [-10, 10]
p_4=(875 , 245)	p'_4=(400 , 480)	-475 [-485, -465]	235 [225, 245]
p_5=(630 , 330)	p'_5=(410 , 330)	-220 [-230, -210]	0 [-10, 10]
p_6=(890 , 430)	p'_6=(670 , 430)	-220 [-230, -210]	0 [-10, 10]
p_7=(725 , 500)	p'_7=(620 , 350)	-105 [-115, -95]	-150 [-160, -140]

Upper bound	Lower bound
$-1000 \leq T_x \leq 1000$	$T_x = 0$ in [-1000,1000]
$-1000 \leq T_y \leq 1000$	$T_y = 0$ in [-1000,1000]
✓	FAIL!
✓	FAIL!
✓	FAIL!
✓	FAIL!
✓	FAIL!
✓	FAIL!
✓	FAIL!

between 0 and 7 inliers

BnB – simple bounds

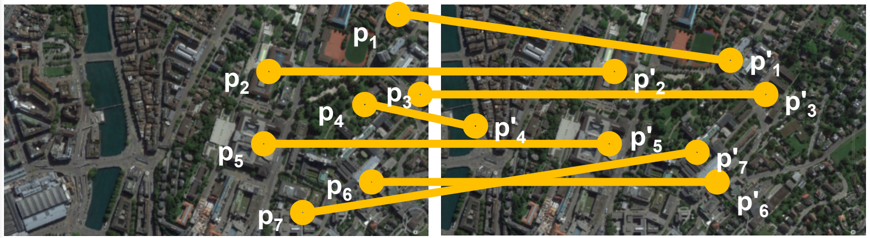


		T_x	T_y
p_1=(950, 20)	p'_1=(700 , 130)	-250 [-260, -240]	110 [100, 120]
p_2=(640 , 160)	p'_2=(420 , 160)	-220 [-230, -210]	0 [-10, 10]
p_3=(1000 , 210)	p'_3=(780 , 210)	-220 [-230, -210]	0 [-10, 10]
p_4=(875 , 245)	p'_4=(400 , 480)	-475 [-485, -465]	235 [225, 245]
p_5=(630 , 330)	p'_5=(410 , 330)	-220 [-230, -210]	0 [-10, 10]
p_6=(890 , 430)	p'_6=(670 , 430)	-220 [-230, -210]	0 [-10, 10]
p_7=(725 , 500)	p'_7=(620 , 350)	-105 [-115, -95]	-150 [-160, -140]

$-450 \le T_x \le 0$ $-1000 \le T_y \le 1000$	$T_x = -225$ in [-450, 0] $T_y = 0$ in [-1000,1000]
✓	FAIL!
✓	✓
✓	✓
FAIL!	FAIL!
✓	✓
✓	✓
✓	FAIL!

between 4 and 6 inliers

BnB – simple bounds

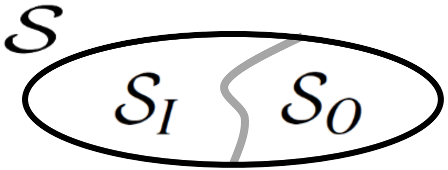


		T_x	T_y
p_1=(950, 20)	p'_1=(700 , 130)	-250 [-260, -240]	110 [100, 120]
p_2=(640 , 160)	p'_2=(420 , 160)	-220 [-230, -210]	0 [-10, 10]
p_3=(1000 , 210)	p'_3=(780 , 210)	-220 [-230, -210]	0 [-10, 10]
p_4=(875 , 245)	p'_4=(400 , 480)	-475 [-485, -465]	235 [225, 245]
p_5=(630 , 330)	p'_5=(410 , 330)	-220 [-230, -210]	0 [-10, 10]
p_6=(890 , 430)	p'_6=(670 , 430)	-220 [-230, -210]	0 [-10, 10]
p_7=(725 , 500)	p'_7=(620 , 350)	-105 [-115, -95]	-150 [-160, -140]

$-450 \leq T_x \leq 0$ $-30 \leq T_y \leq 30$	T_x = -225 in [-450, 0] T_y = 0 in [-30, 30]
FAIL!	FAIL!
✓	✓
✓	✓
FAIL!	FAIL!
✓	✓
✓	✓
FAIL!	FAIL!

between 4 and 4
inliers

Mathematical formulation

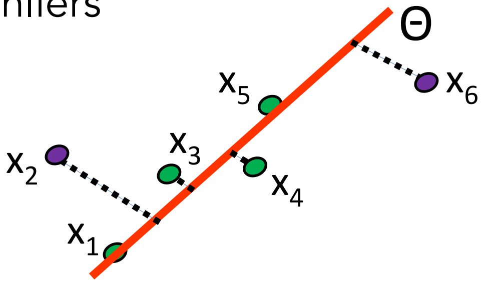


Input: data S

- We note x_i is the i-th data point

Output: model Θ and inliers/outliers ($\mathcal{S}_I$, $\mathcal{S}_O$)

- More precisely, the "model leading to the highest nb of inliers"



What are the unknowns?

- The model Θ
- The set of inliers/outliers ($\mathcal{S}_I$, $\mathcal{S}_O$)

Cost: to distinguish inliers/outliers

- An inlier has a "small" cost, *i.e.* $f(\Theta, x_i) \leq T$

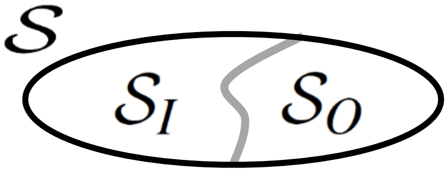
$$\max_{\mathcal{S}_\mathrm{I},\Theta} \quad \mathbf{card}\ (\mathcal{S}_\mathrm{I})$$

maximize the nb of inliers

$$\text{s.t.} \quad f(\Theta, x_i) \leq T, \forall i \in \mathcal{S}_\mathrm{I}$$

all the inliers must have a "small" error

Chicken-and-Egg Problem

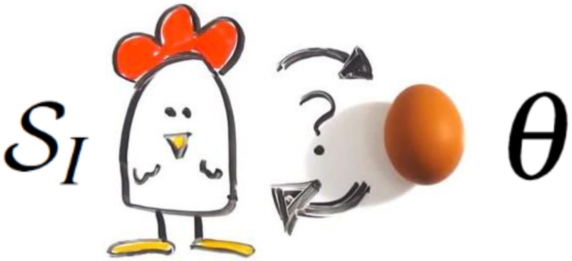


$$\max_{\mathcal{S}_I,\Theta} \quad \mathbf{card}\,(\mathcal{S}_I)$$

$$\text{s.t.} \quad f(\Theta, x_i) \le T, \forall i \in \mathcal{S}_I$$

nb of inliers

all the inliers must have a “small” error



Test **all** $\mathcal{S}_I$

→ Finite (2^N) but intractable

Test **all** θ

→ Non-finite (continuous space) and intractable

$$[\,\underline{\theta}\,,\,\overline{\theta}\,] \rightarrow \begin{matrix}\mathcal{S}_I \\ \theta\end{matrix}$$

Mathematical formulation (1)

Initial formulation

$$\max_{\mathcal{S}_{\mathrm{I}},\Theta} \quad \mathbf{card}\ (\mathcal{S}_{\mathrm{I}})$$
$$\text{s.t.} \qquad f(\Theta, x_i) \leq T, \forall i \in \mathcal{S}_{\mathrm{I}}$$

Alternative formulation

$$\max_{\mathbf{z},\Theta} \quad \sum_{i=1}^{N} z_i$$
$$\text{s.t.} \quad z_i f(\Theta, x_i) \leq z_i T, \forall i = 1..N$$
$$\text{and} \quad z_i \in \{0, 1\}, \forall i = 1..N$$

Let $z_i = 1$ if the i[th] data point is an inlier -> $f(\Theta, x_i) \leq T$
$= 0$ else -> $0 \leq 0$

Mathematical formulation (2)

Example with line fitting

- Model: $\Theta = (a, b, c)^T$
- Cost function: $|\mathbf{A}_i^T \Theta| = |ax_i + by_i + c|$

$$
\begin{aligned}
\max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\
\text{and} \quad & ||\Theta|| = 1 \\
\text{and} \quad & z_i \in \{0, 1\}, \forall i = 1..N
\end{aligned}
$$

geometric

scale

binary

Hard to solve: binary/continuous unknowns and bilinearities

Mathematical formulation (3)

Interesting observation:

➡ Relaxation is equivalent!

Intuitively: If z_i would be fractional, maximizing the objective would force $z_i = 1$
Since the binary variables appear on both sides of the inequalities, the constraints become active as soon as $z_i > 0$.
Hence, fractional solutions are not possible!

$$\begin{aligned} \max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\ \text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\ \text{and} \quad & \|\Theta\| = 1 \\ \text{and} \quad & \boxed{z_i \in \{0,1\}}, \forall i = 1..N \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\ \text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\ \text{and} \quad & \|\Theta\| = 1 \\ \text{and} \quad & \boxed{z_i \in [0,1]}, \forall i = 1..N \end{aligned}$$

Still hard to solve: bilinear, non-convex, etc.

How to handle bilinearities ?

Linear Programming (simple)

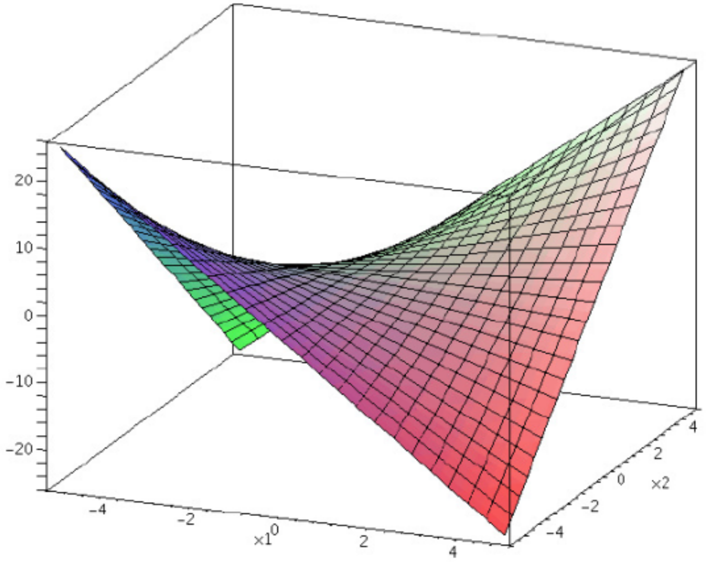
$$\boxed{\begin{aligned} &\max_x \; c^T x \\ &\text{s.t. } Ax \le b \\ &\text{and } x \ge 0 \end{aligned}}$$

$$\begin{aligned} \max_{x,y} \quad & x + 2y \\ \text{s.t.} \quad & 2x + 3y \le 20 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}$$

With bilinearities ?

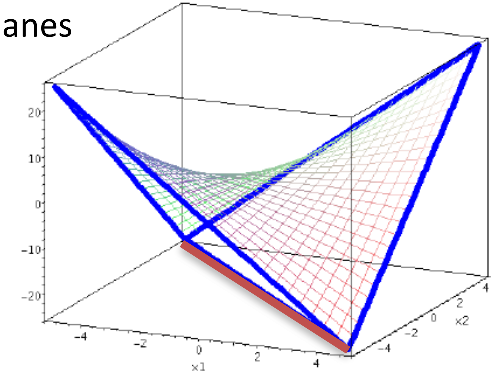
$$\begin{aligned} \max_{x,y} \quad & x + 2y \\ \text{s.t.} \quad & 2\boxed{xy} + 3y \le 20 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}$$

Envelopes for bilinearities

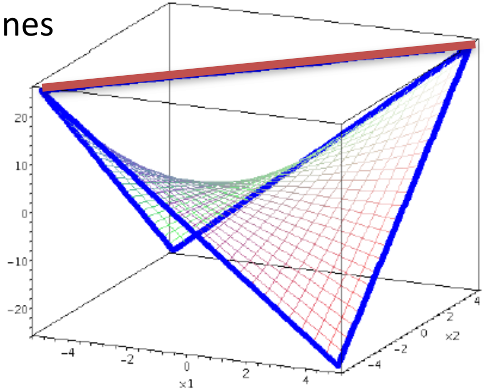


The Bilinear surface $x_1 x_2$

"above" these two planes



"below" these two planes



"Relaxations of multilinear convex envelopes: dual is better than primal"
Alberto Costa, Leo Liberti, International conference on Experimental Algorithms, 2012

Envelopes for bilinearities

Bilinear relaxation by concave and convex envelopes

bilinear equality $\gamma = \alpha\beta$ with the bounding-boxes $[\underline{\alpha}, \bar{\alpha}]$ and $[\underline{\beta}, \bar{\beta}]$ is relaxed by the following envelopes:

$$\gamma \geq \max(\underline{\alpha}\beta + \underline{\beta}\alpha - \underline{\alpha}\underline{\beta}, \bar{\alpha}\beta + \bar{\beta}\alpha - \bar{\alpha}\bar{\beta})$$

$$\gamma \leq \min(\bar{\alpha}\beta + \underline{\beta}\alpha - \bar{\alpha}\underline{\beta}, \underline{\alpha}\beta + \bar{\beta}\alpha - \underline{\alpha}\bar{\beta})$$

We collectively represent them as $conv(\alpha, \beta) \leq \gamma \leq conc(\alpha, \beta)$

Example:

$$\alpha \in [3,5],\ \beta \in [2,7]$$

$$\Rightarrow \quad \begin{aligned} \gamma &\geq \max(3\beta + 2\alpha - 3*2, 5\beta + 7\alpha - 5*7) \\ \gamma &\leq \min(5\beta + 2\alpha - 5*2, 3\beta + 7\alpha - 3*7) \end{aligned} \quad \Rightarrow \quad \begin{aligned} \gamma &\geq 3\beta + 2\alpha - 3*2 \\ \gamma &\geq 5\beta + 7\alpha - 5*7 \\ \gamma &\leq 5\beta + 2\alpha - 5*2 \\ \gamma &\leq 3\beta + 7\alpha - 3*7 \end{aligned}$$

Example with bilinearities (1)

$$\boxed{\begin{aligned}\max_{x,y}\ & x+2y\\ \text{s.t. } & 2xy+3y\le 20\\ & 3\le x\le 5\\ & 2\le y\le 7\\ & x\in\mathbb{R}, y\in\mathbb{R}\end{aligned}}$$

$\rightarrow$

$$x\in[3,5],\ y\in[2,7]$$

$$\boxed{w=xy}$$

$$\gamma \ge \max(\underline{\alpha}\beta+\underline{\beta}\alpha-\underline{\alpha}\underline{\beta},\ \bar{\alpha}\beta+\bar{\beta}\alpha-\bar{\alpha}\bar{\beta})$$
$$\gamma \le \min(\bar{\alpha}\beta+\underline{\beta}\alpha-\bar{\alpha}\underline{\beta},\ \underline{\alpha}\beta+\bar{\beta}\alpha-\underline{\alpha}\bar{\beta})$$

$$\boxed{\begin{aligned}w&\ge 3y+2x-6\\ w&\ge 5y+7x-35\\ w&\le 5y+2x-10\\ w&\le 3y+7x-21\end{aligned}}$$

$\rightarrow$

$$\boxed{\begin{aligned}\max_{x,y,w}\ & x+2y\\ \text{s.t. } & 2w+3y\le 20\\ & w\ge 3y+2x-6\\ & w\ge 5y+7x-35\\ & w\le 5y+2x-10\\ & w\le 3y+7x-21\\ & 3\le x\le 5\\ & 2\le y\le 7\\ & x\in\mathbb{R}, y\in\mathbb{R}\end{aligned}}$$

Example with bilinearities (2)

Canonical form

$$\max_x c^T x \quad \text{s.t. } Ax \le b \quad \text{and } x \ge 0$$

Matlab's linprog

$$\min_x c^T x \text{ such that } \begin{cases} Ax \le b \\ A_{eq}x = b_{eq} \\ l_b \le x \le u_b \end{cases}$$

$$\begin{aligned} \max_{x,y,w} \quad & x + 2y \\ \text{s.t.} \quad & 2w + 3y \le 20 \\ & w \ge 3y + 2x - 6 \\ & w \ge 5y + 7x - 35 \\ & w \le 5y + 2x - 10 \\ & w \le 3y + 7x - 21 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & x \in \mathbb{R}, y \in \mathbb{R} \end{aligned}$$

$$\begin{aligned} \max_{x,y,w} \quad & x + 2y + 0w \\ \text{s.t.} \quad & 0x + 3y + 2w \le 20 \\ & 2x + 3y - w \le 6 \\ & 7x + 5y - w \le 35 \\ & -2x - 5y + w \le -10 \\ & -7x - 3y + w \le -21 \\ & 3 \le x \le 5 \\ & 2 \le y \le 7 \\ & -\infty \le w \le +\infty \\ & x \in \mathbb{R}, y \in \mathbb{R}, w \in \mathbb{R} \end{aligned}$$

```
c=-[1 2 0]
A=[0 3 2;
    2 3 -1;
    7 5 -1;
   -2 -5 1;
   -7 -3 1]
b=[20 6 35 -10 -21]
lb=[3 2 -inf]
ub=[5 7 inf]
[sol,val]=linprog(c,A,b,[],[],lb,ub)
```

```
sol=[3.5 2 7]
val=-7.5
```

Here x*y=w

But not in general

Example with bilinearities (3)

$$\max_{x,y} \; x+2y$$
$$\text{s.t. } 2xy+3y \le 20$$
$$3 \le x \le 5$$
$$2 \le y \le 7$$
$$x \in \mathbb{R}, y \in \mathbb{R}$$

Minor modification of the problem

$$\max_{x,y} \; x+2y$$
$$\text{s.t. } 2xy+3y \le \boxed{50}$$
$$3 \le x \le 5$$
$$2 \le y \le 7$$
$$x \in \mathbb{R}, y \in \mathbb{R}$$

→

$$\max_{x,y,w} \; x+2y+0w$$
$$\text{s.t. } 0x+3y+2w \le 50$$
$$2x+3y-w \le 6$$
$$7x+5y-w \le 35$$
$$-2x-5y+w \le -10$$
$$-7x-3y+w \le -21$$
$$3 \le x \le 5$$
$$2 \le y \le 7$$
$$-\infty \le w \le +\infty$$
$$x \in \mathbb{R}, y \in \mathbb{R}, w \in \mathbb{R}$$

Matlab's linprog

$$\min_x c^T x \text{ such that } \begin{cases} Ax \le b \\ A_{eq}x = b_{eq} \\ l_b \le x \le u_b \end{cases}$$

→

```
c=-[1 2 0]
A=[0 3 2;
   2 3 -1;
   7 5 -1;
  -2 -5 1;
  -7 -3 1]
b=[50 6 35 -10 -21]
lb=[3 2 -inf]
ub=[5 7 inf]
[sol,val]=linprog(c,A,b,[],[],lb,ub)
```

→

```
sol=[3.70 5.24 17.13]
val=-14.19
```

x*y=19.41 ≠ w=17.13

In general, not equal

Mathematical formulation (3)

Interesting observation:

➡ Relaxation is equivalent!

Intuitively: If z_i would be fractional, maximizing the objective would force $z_i = 1$
Since the binary variables appear on both sides of the inequalities, the constraints become active as soon as $z_i > 0$.
Hence, fractional solutions are not possible!

$$
\begin{aligned}
\max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\
\text{and} \quad & \|\Theta\| = 1 \\
\text{and} \quad & \boxed{z_i \in \{0,1\}}, \forall i = 1..N
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
\max_{\mathbf{z},\Theta} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & z_i|\mathbf{A}_i^T\Theta| \leq z_i T, \forall i = 1..N \\
\text{and} \quad & \|\Theta\| = 1 \\
\text{and} \quad & \boxed{z_i \in [0,1]}, \forall i = 1..N
\end{aligned}
$$

Still hard to solve: bilinear, non-convex, etc.

Mathematical formulation (4)

Bilinear relaxation: by concave and convex envelopes

$$\max_{\mathbf{z},\Theta,w,\mu} \quad \sum_{i=1}^{N} z_i$$

$$\text{s.t.} \quad \underline{|A_i^T w_i|} \leq z_i T, \forall i \longrightarrow \boxed{z_i |\mathbf{A}_i^T \Theta|} \text{ and } \boxed{w_i = z_i \Theta}$$

$$\text{and} \quad \underline{\sum_{j=1}^{D} \mu_j} = 1 \text{ and } \underline{\Theta} \leq \Theta \leq \overline{\Theta} \longrightarrow \boxed{\mu_j = \Theta_j^2}$$

$$\text{and} \quad 0 \leq z_i \leq 1, \forall i$$

$$\text{and} \quad \underline{conv(z_i, \Theta_j)} \leq w_{ij} \leq \underline{conc(z_i, \Theta_j)}$$

$$i = 1 \ldots N, j = 1 \ldots D$$

$$\text{and} \quad \underline{conv(\Theta_j)} \leq \mu_j \leq \underline{conc(\Theta_j)}, \forall j = 1 \ldots D$$

Bilinearities have been removed

→ All **linear!!**

But how to get bounding boxes of Θ?

Bounds

Upper bound ("too good")

Obtained by LP via the envelope relaxation

$$
\begin{aligned}
\max_{\mathbf{z},\Theta,w,\mu} \quad & \sum_{i=1}^{N} z_i \\
\text{s.t.} \quad & |A_i^T w_i| \leq z_i T, \forall i \\
\text{and} \quad & \sum_{j=1}^{D} \mu_j = 1 \text{ and } \underline{\Theta} \leq \Theta \leq \overline{\Theta} \\
\text{and} \quad & 0 \leq z_i \leq 1, \forall i \\
\text{and} \quad & conv(z_i, \Theta_j) \leq w_{ij} \leq conc(z_i, \Theta_j) \\
& i = 1 \ldots N, j = 1 \ldots D \\
\text{and} \quad & conv(\Theta_j) \leq \mu_j \leq conc(\Theta_j), \forall j = 1 \ldots D
\end{aligned}
$$

Lower bound

Given Θ obtained by LP, simple test on *z*

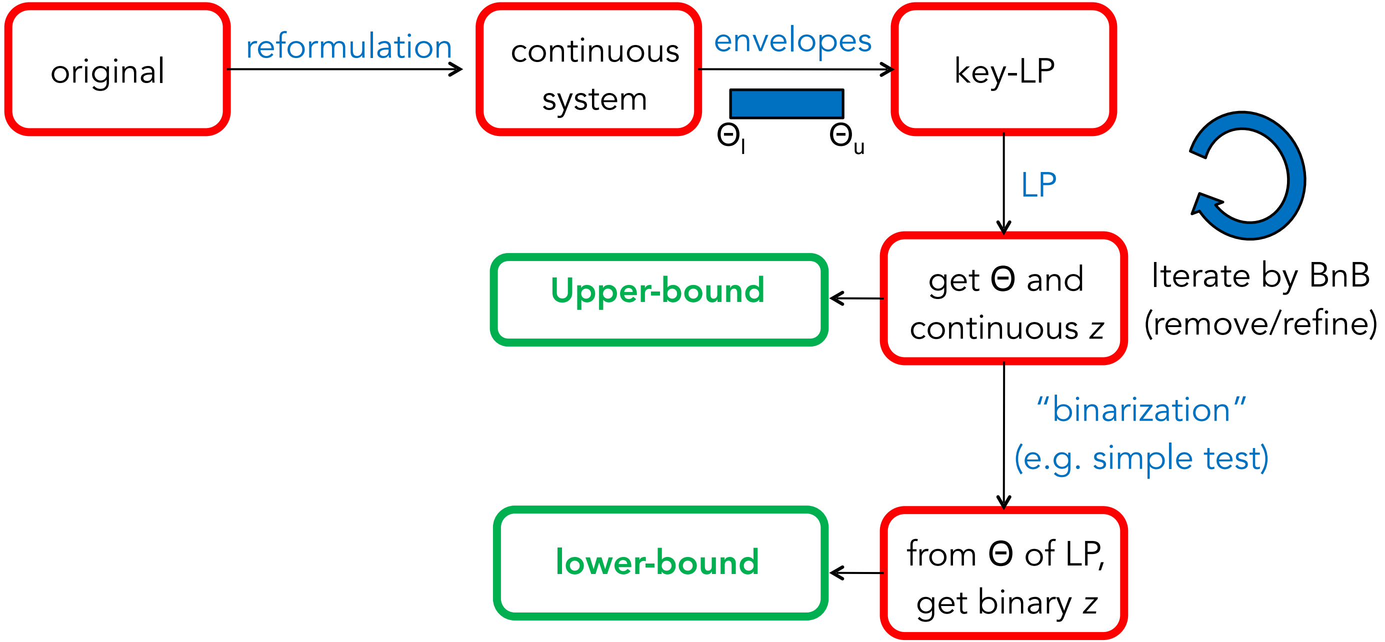
If $f(\Theta, x_i) \leq T$ then $z_i = 1$

else $z_i = 0$

Algorithm

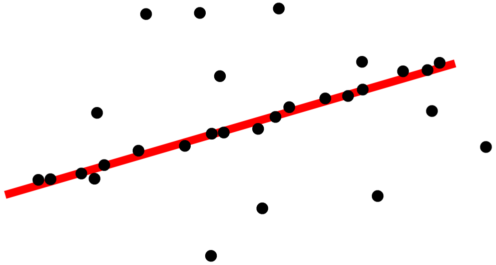
Branch-and-bound

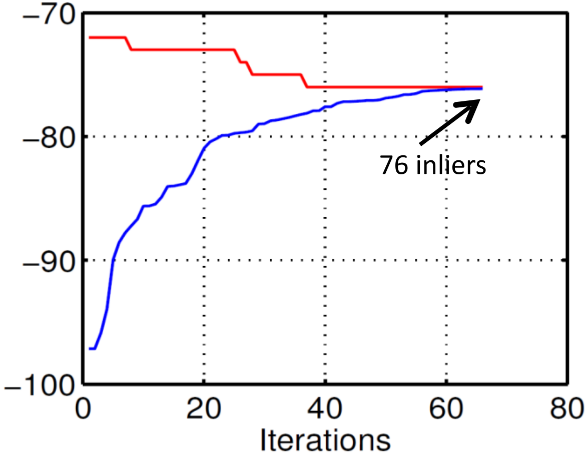
- Used to get bounding boxes of the model Θ
- Permits to reduce the relaxation gap: lower and upper bounds

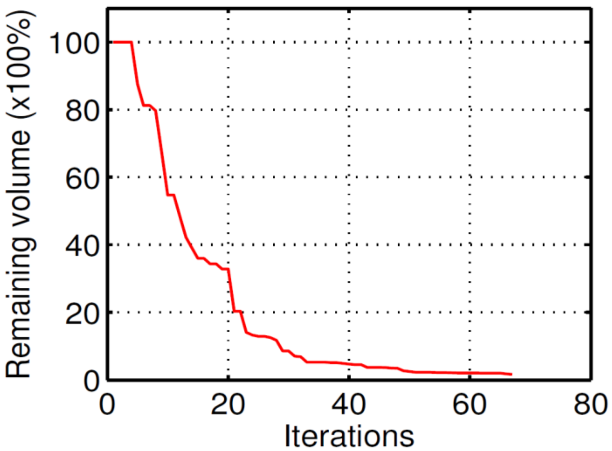


"Consensus Set Maximization with Guaranteed Global Optimality for Robust Geometry Estimation", Hongdong Li, ICCV'09

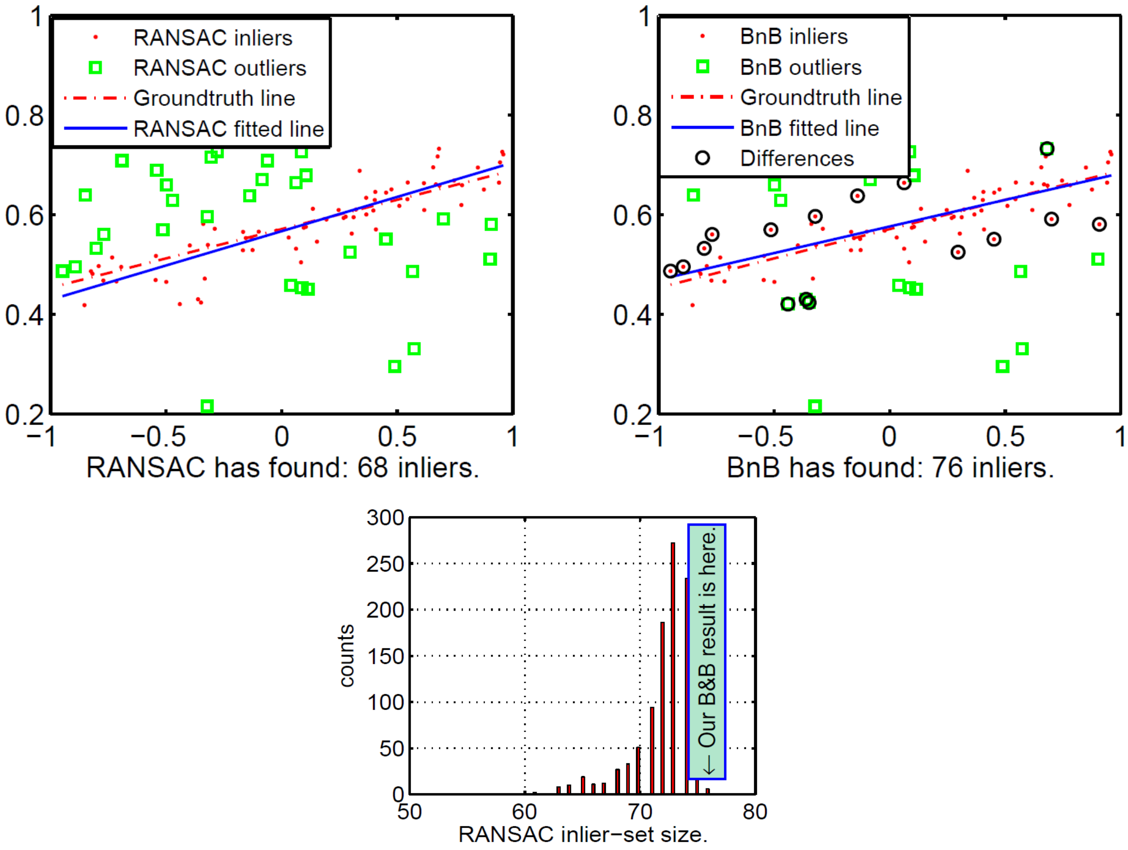
Line Fitting





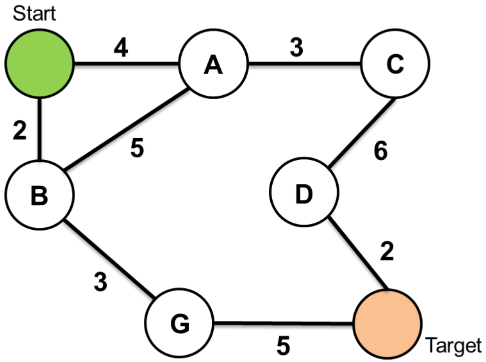


Line Fitting

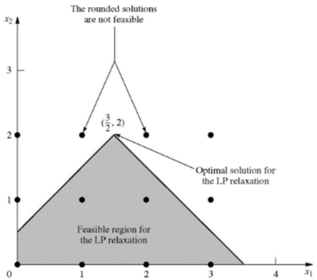


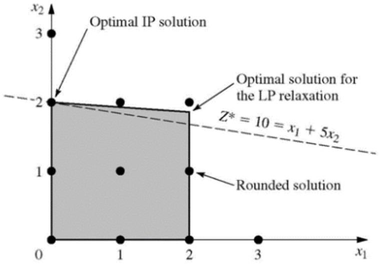
Summary

(1) Optimal Path Search with BnB



(2) Integer Linear Programming





(3) Consensus Set Maximization