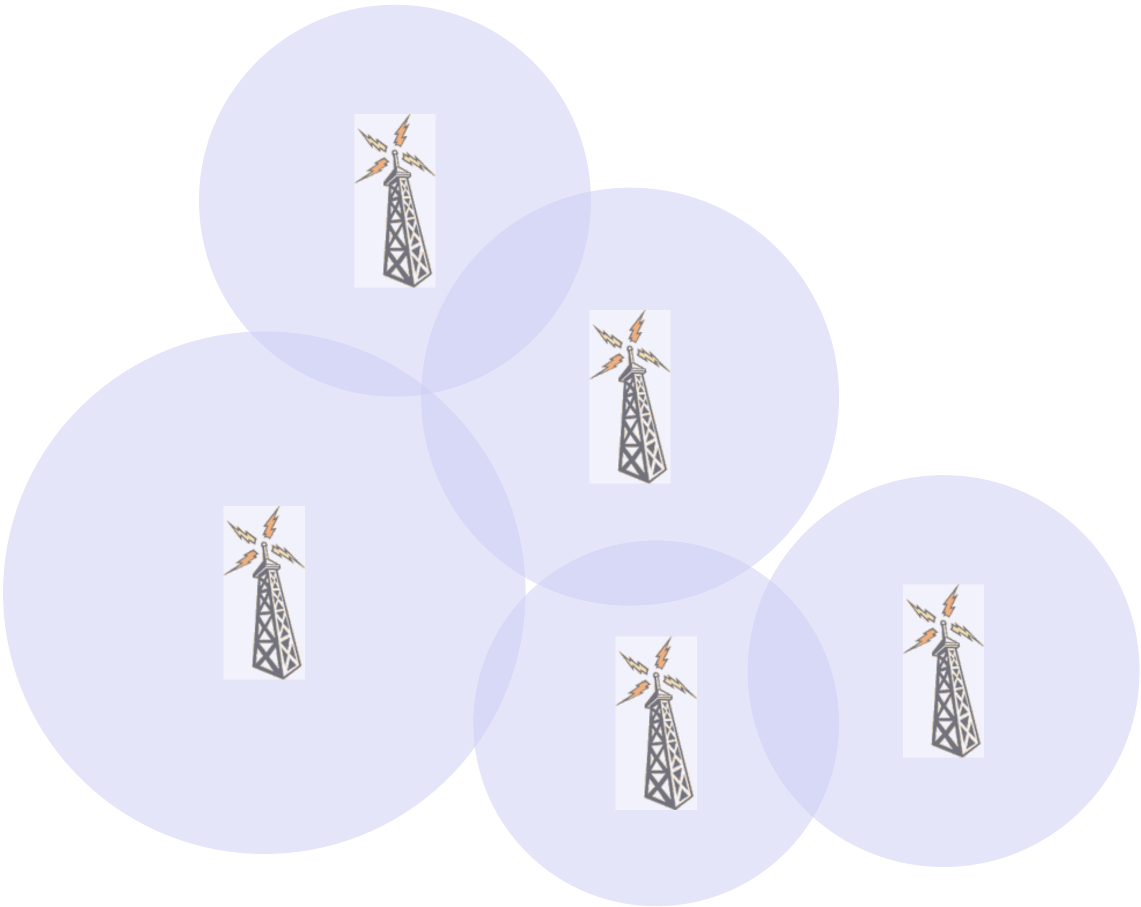
Mathematical Tools for Engineering and Management

Lecture 6

23 Nov 2011

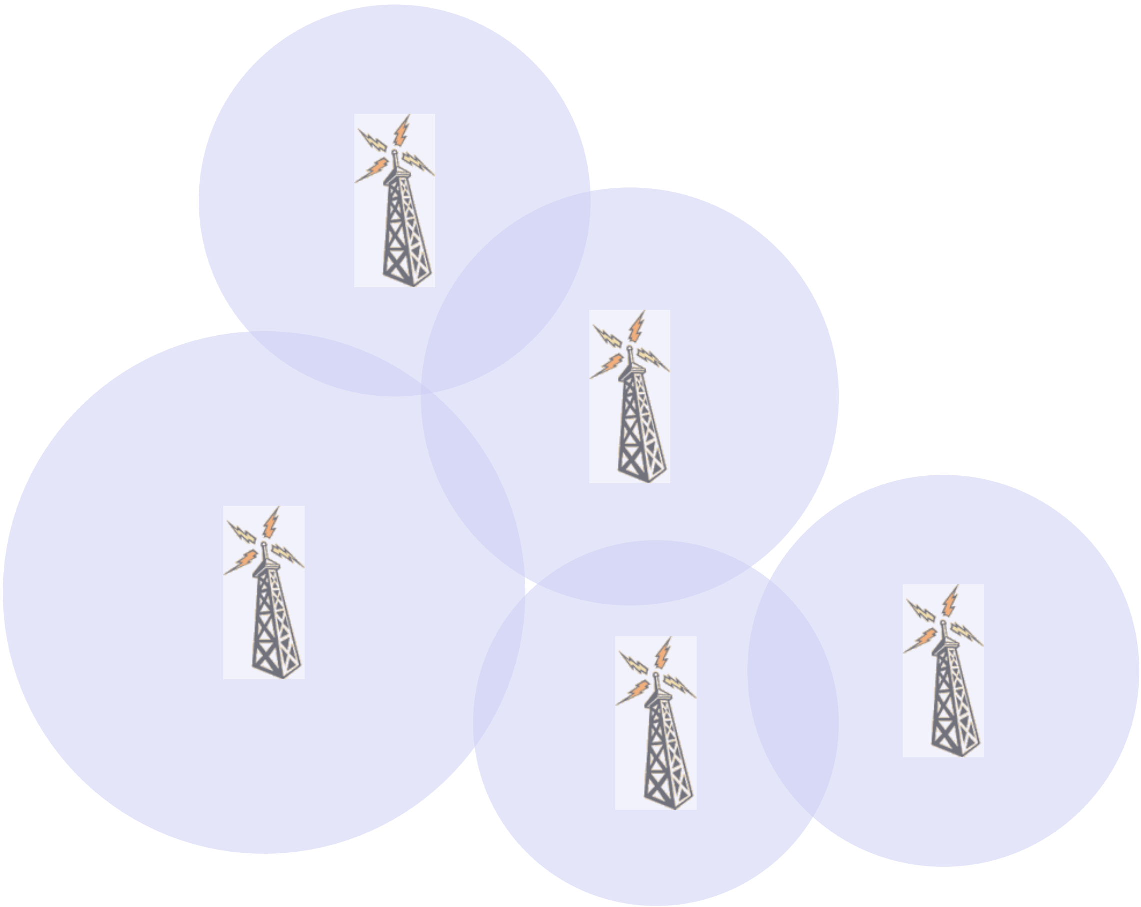
- Models, Data and Algorithms
- Linear Optimization
- Mathematical Background: Polyhedra, Simplex-Algorithm
- Sensitivity Analysis; (Mixed) Integer Programming
- MIP Modelling
- MIP Modelling: More Examples; Branch & Bound
- Cutting Planes; Combinatorial Optimization: Examples, Graphs, Algorithms
- Complexity Theory
- Nonlinear Optimization
- Scheduling
- Lot Sizing
- Multicriteria Optimization
- Oral exam

- Problem: assign transmission frequencies to antennas



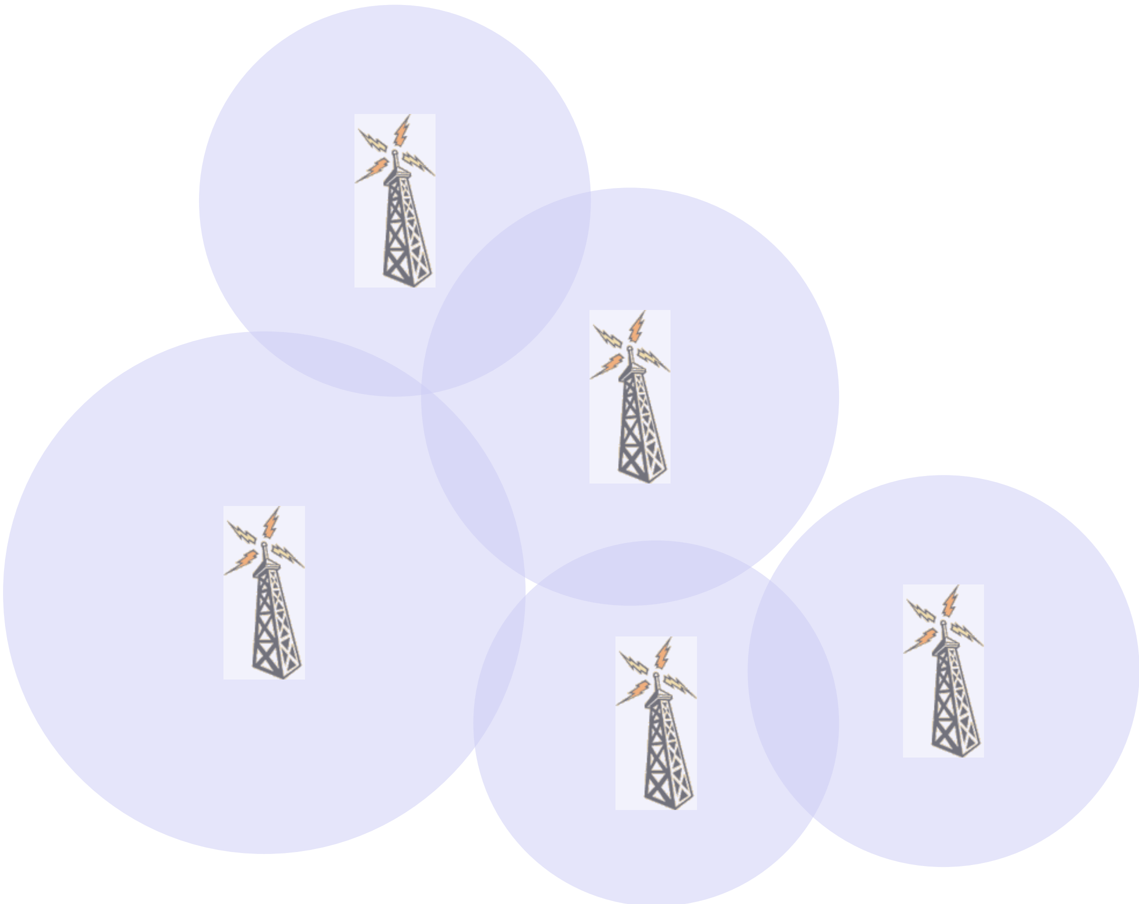
▷ Problem: assign transmission frequencies to antennas such that

- any two antennas with overlapping regions send on different frequencies



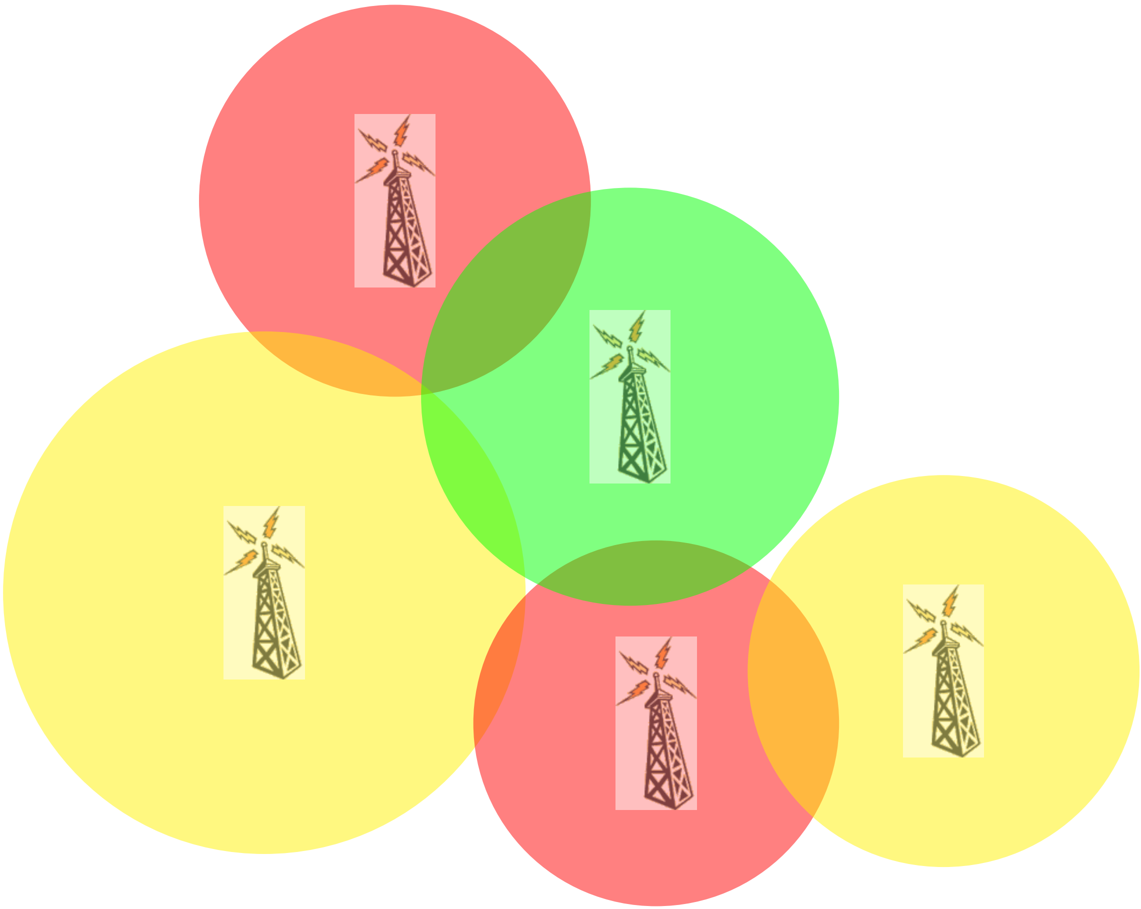
▷ Problem: assign transmission frequencies to antennas such that

- any two antennas with overlapping regions send on different frequencies
- as few frequencies as possible are used



▷ Problem: assign transmission frequencies to antennas such that

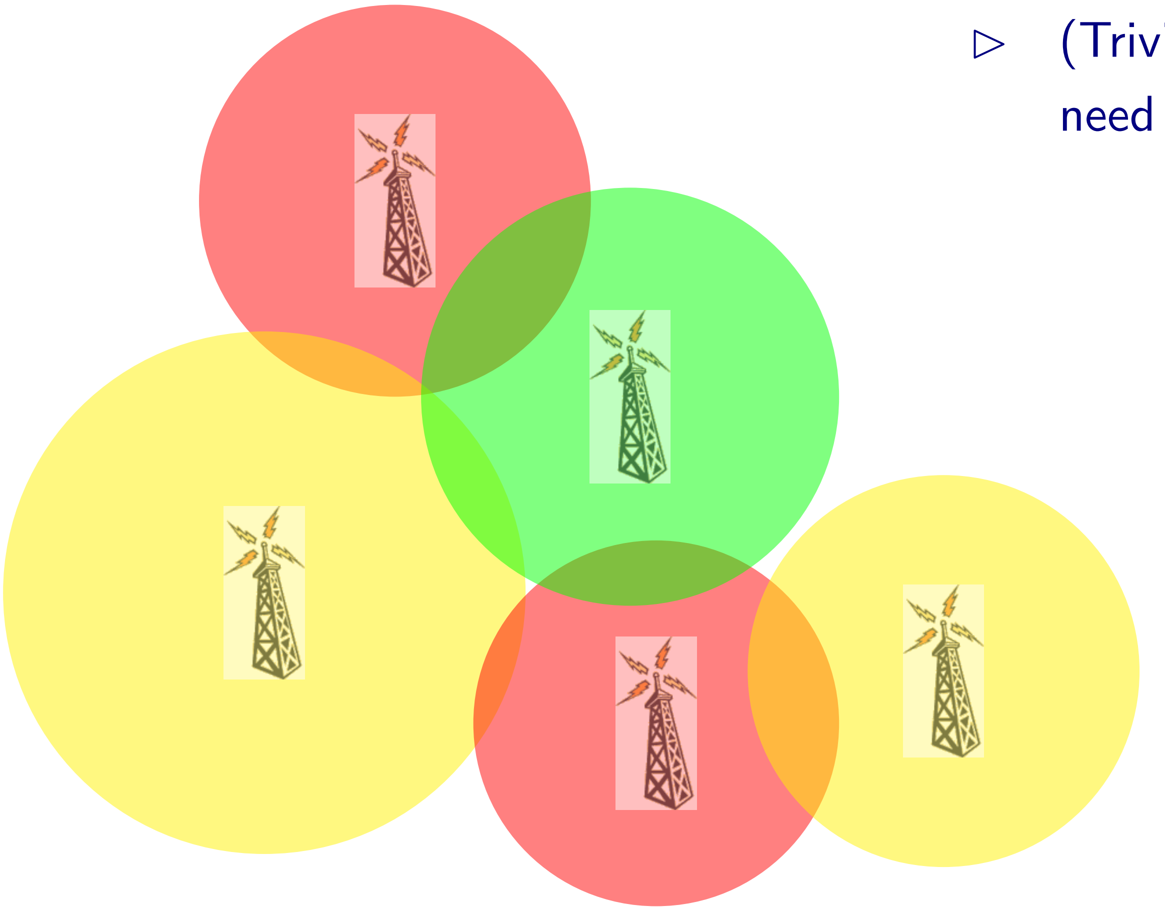
- any two antennas with overlapping regions send on different frequencies
- as few frequencies as possible are used



(Vertex Colouring Problem)

▷ Problem: assign transmission frequencies to antennas such that

- any two antennas with overlapping regions send on different frequencies
- as few frequencies as possible are used

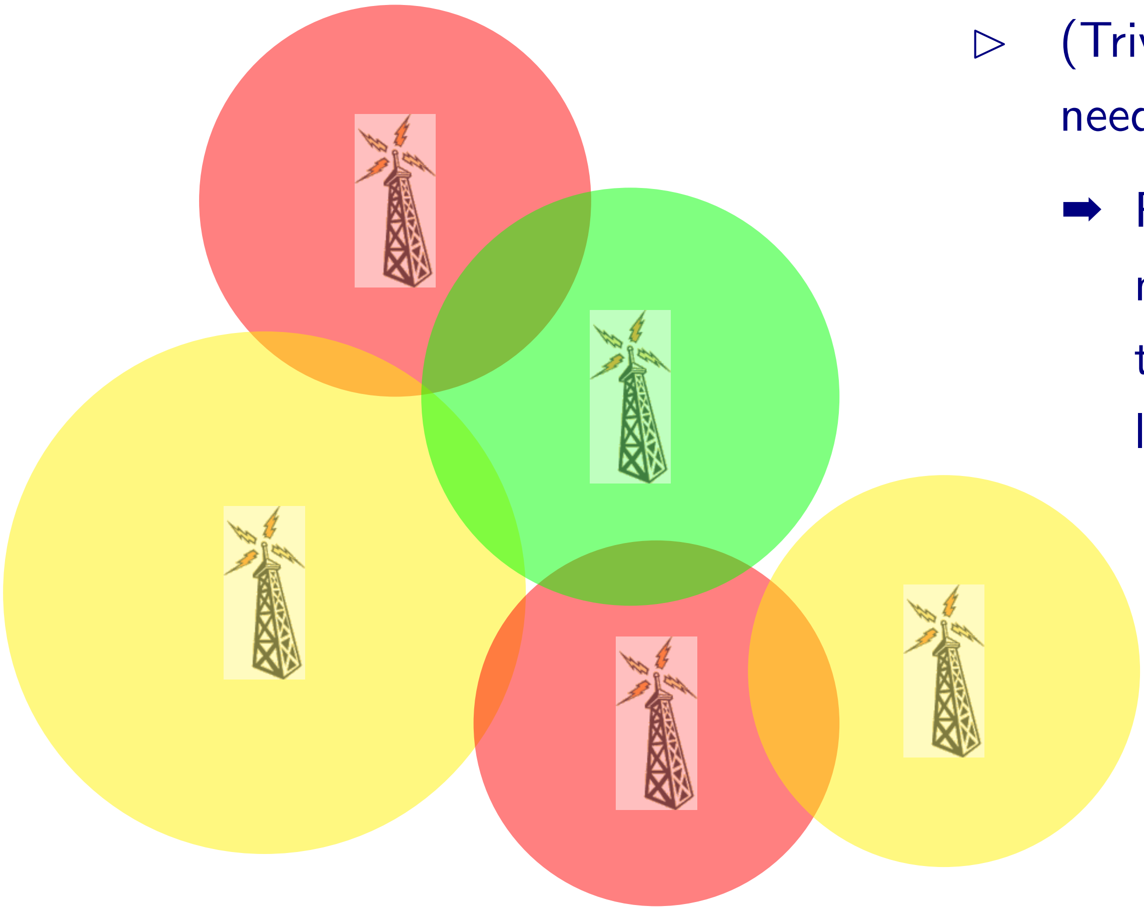


▷ (Trivial) observation: For n antennas we need at most n different frequencies

(Vertex Colouring Problem)

▷ Problem: assign transmission frequencies to antennas such that

- any two antennas with overlapping regions send on different frequencies
- as few frequencies as possible are used



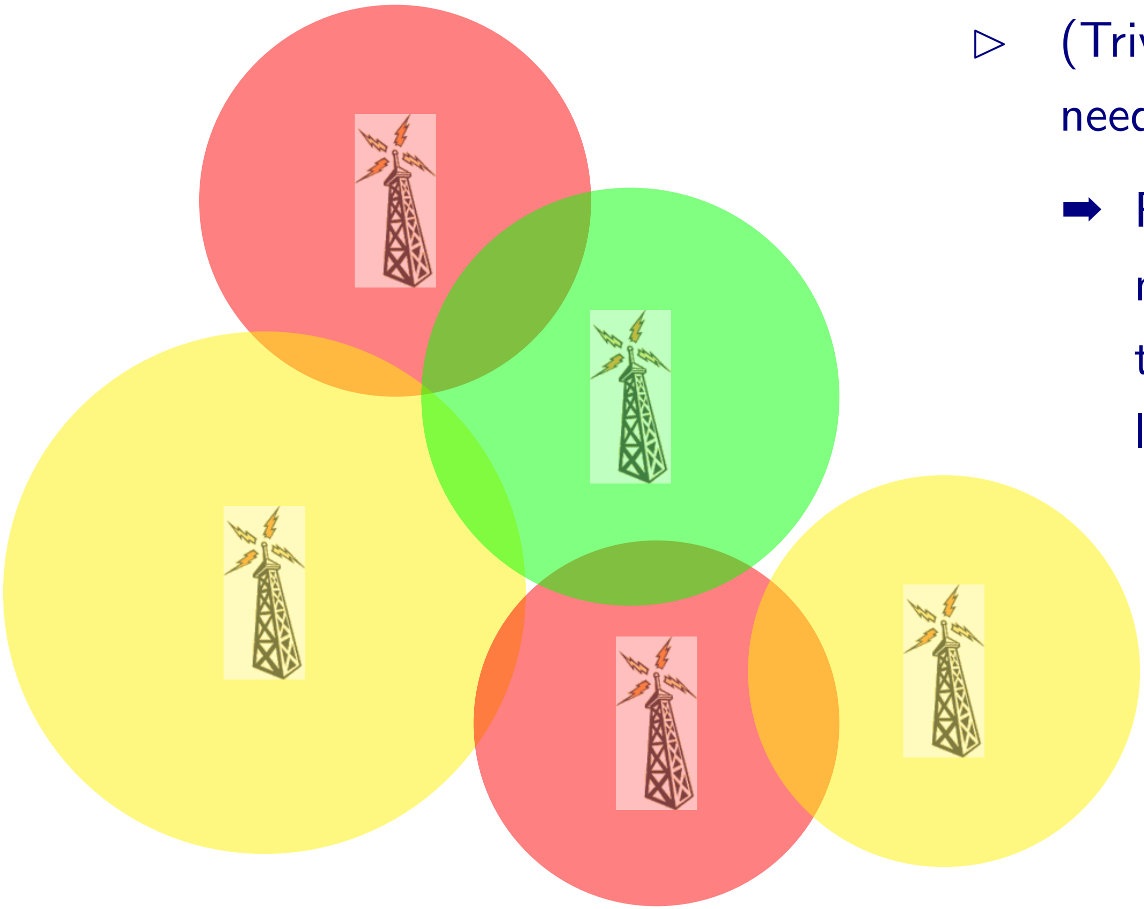
▷ (Trivial) observation: For n antennas we need at most n different frequencies

➡ First modelling step:
number the antennas $1, \ldots, n$,
the possible frequencies also $1, \ldots, n$,
let the used frequencies be $1, \ldots, k$

(Vertex Colouring Problem)

▷ Problem: assign transmission frequencies to antennas such that

- any two antennas with overlapping regions send on different frequencies
- as few frequencies as possible are used



▷ (Trivial) observation: For n antennas we need at most n different frequencies

➡ First modelling step:
number the antennas $1, \ldots, n$,
the possible frequencies also $1, \ldots, n$,
let the used frequencies be $1, \ldots, k$

➡ Problem to solve:
find an assignment of frequencies to antennas with minimal k

(Vertex Colouring Problem)

S Antennas: $A = \{1, \ldots, n\}$

V (frequencies of antennas)

$$1 \leq x_i \leq n \quad \text{for all } i \in A$$

$$x_i \quad \text{integer} \quad \text{for all } i \in A$$

S Antennas: $A = \{1, \ldots, n\}$

V (frequencies of antennas)

$$1 \leq x_i \leq n \quad \text{for all } i \in A$$

$$x_i \quad \text{integer} \quad \text{for all } i \in A$$

(highest frequency number)

$$1 \leq y \leq n$$

$$y \quad \text{integer}$$

S Antennas: $A = \{1, \ldots, n\}$

C (used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

V (frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S Antennas: $A = \{1, \ldots, n\}$

Objective: minimize (highest used frequency number) y

C (used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

V (frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S Antennas: $A = \{1, \ldots, n\}$

Objective: minimize (highest used frequency number) y

C (used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

V (frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i,j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$ $\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \neq x_j$ for all conflict pairs $(i, j) \in C$

V

(frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i, j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$
$\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \neq x_j$ for all conflict pairs $(i, j) \in C$

V

(frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i, j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$
$\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \not= x_j$ for all conflict pairs $(i, j) \in C$

$|x_i - x_j| \geq 1$

V

(frequencies of antennas)

$1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number)

$1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i, j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$

$\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \neq x_j$ for all conflict pairs $(i,j) \in C$

$|x_i - x_j| \geq 1$

V

(frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i,j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$
$\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \neq x_j$ for all conflict pairs $(i,j) \in C$

$|x_i - x_j| \geq 1$

$x_i - x_j \geq 1$ or $x_j - x_i \geq 1$

V

(frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i,j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$ $\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \neq x_j$ for all conflict pairs $(i,j) \in C$

$|x_i - x_j| \geq 1$

$x_i - x_j \geq 1$ or $x_j - x_i \geq 1$

V

(frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i,j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$
$\subseteq A \times A$

Objective: minimize (highest used frequency number) y

C

(used frequency numbers) $x_i \leq y$ for all antennas $i \in A$

(frequency conflicts) $x_i \neq x_j$ for all conflict pairs $(i, j) \in C$

$|x_i - x_j| \geq 1$

$x_i - x_j \geq 1$ or $x_j - x_i \geq 1$ ➡ ?

V

(frequencies of antennas) $1 \leq x_i \leq n$ for all $i \in A$

x_i integer for all $i \in A$

(highest frequency number) $1 \leq y \leq n$

y integer

S

Antennas: $A = \{1, \ldots, n\}$

Conflicts: $C = \{(i, j) \mid$ cell of antenna $i \in A$ overlaps with cell of antenna $j \in A\}$ $\subseteq A \times A$

▷ How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

- How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

- Idea: “switch” between the two “or” options with binary variables

V $\quad \xi_{i,j} \in \{0,1\}$ for all conflicting pairs $(i,j) \in C$

- How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

- Idea: "switch" between the two "or" options with binary variables

V

$\xi_{i,j} \in \{0,1\}$ for all conflicting pairs $(i,j) \in C$

➡ $\xi_{i,j} = 1 \quad \Leftrightarrow$ frequency assigned to i is $\leq$ frequency assigned to j

▷ How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

▷ Idea: "switch" between the two "or" options with binary variables

V

$\xi_{i,j} \in \{0,1\}$ for all conflicting pairs $(i,j) \in C$

➡ $\xi_{i,j} = 1$ ⇔ frequency assigned to i is $\leq$ frequency assigned to j

⇔ first "or" option is switched off

- How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

- Idea: "switch" between the two "or" options with binary variables

V $\quad \xi_{i,j} \in \{0,1\}$ for all conflicting pairs $(i,j) \in C$

➡ $\xi_{i,j} = 1 \quad \Leftrightarrow$ frequency assigned to i is $\leq$ frequency assigned to j

$\Leftrightarrow$ first "or" option is switched off

and additional constraints:

C

("or" option 1) $\quad x_i - x_j \geq 1 - n \cdot \xi_{i,j}$

("or" option 2) $\quad x_j - x_i \geq 1 - n \cdot (1 - \xi_{i,j})$

- How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

- Idea: "switch" between the two "or" options with binary variables

V $\xi_{i,j} \in \{0,1\}$ for all conflicting pairs $(i,j) \in C$

➡ $\xi_{i,j} = 1$ $\Leftrightarrow$ frequency assigned to i is $\leq$ frequency assigned to j

$\Leftrightarrow$ first "or" option is switched off

and additional constraints:

C

("or" option 1) $x_i - x_j \geq 1 - n \cdot \xi_{i,j}$

("or" option 2) $x_j - x_i \geq 1 - n \cdot (1 - \xi_{i,j})$

- Extension: model "or" constraints involving more than two clauses

▷ How to model the restriction

$$x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \qquad ?$$

▷ Idea: "switch" between the two "or" options with binary variables

V $\quad \xi_{i,j} \in \{0,1\}$ for all conflicting pairs $(i,j) \in C$

➡ $\xi_{i,j} = 1 \quad \Leftrightarrow$ frequency assigned to i is $\leq$ frequency assigned to j

$\Leftrightarrow$ first "or" option is switched off

and additional constraints:

C

("or" option 1) $\quad x_i - x_j \geq 1 - n \cdot \xi_{i,j}$

("or" option 2) $\quad x_j - x_i \geq 1 - n \cdot (1 - \xi_{i,j})$

▷ Extension: model "or" constraints involving more than two clauses

➡ Combination with set-partitioning constraints...

- We know how to solve linear programs efficiently

▷ We know how to solve linear programs efficiently

➡ Solving the LP relaxation gives some information on the MIP:

- a (probably) fractional solution of the LP
- an upper bound on the MIP optimum (for maximizing!)

▷ If the solution has integer values in all required entries, we are done!

- ▷ We know how to solve linear programs efficiently
  - ➡ Solving the LP relaxation gives some information on the MIP:
    - a (probably) fractional solution of the LP
    - an upper bound on the MIP optimum (for maximizing!)
- ▷ If the solution has integer values in all required entries, we are done!
- ▷ Otherwise there is a fractional entry $x_i = q \notin \mathbb{Z}$ which should be integer
  - ➡ All possible solutions to the MIP have either $x_i \leq \lfloor q \rfloor$ or $x_i \geq \lceil q \rceil$

- ▷ We know how to solve linear programs efficiently
  - ➡ Solving the LP relaxation gives some information on the MIP:
    - a (probably) fractional solution of the LP
    - an upper bound on the MIP optimum (for maximizing!)
- ▷ If the solution has integer values in all required entries, we are done!
- ▷ Otherwise there is a fractional entry $x_i = q \notin \mathbb{Z}$ which should be integer
  - ➡ All possible solutions to the MIP have either $x_i \leq \lfloor q \rfloor$ or $x_i \geq \lceil q \rceil$
- ▷ Idea: split up the MIP into two sub-problems:
  - add the constraint $x_i \leq \lfloor q \rfloor$ for the first sub-problem
  - add the constraint $x_i \geq \lceil q \rceil$ for the second sub-problem
  - ➡ Problem is solved if both sub-problems are solved
  - ➡ Solve the sub-problems recursively, in the same way

Objective function: maximize (total revenue) $r_{\text{beetle}} \cdot x_{\text{beetle}} + r_{\text{cabrio}} \cdot x_{\text{cabrio}}$

C

(total raw material available) $\rho_{\text{beetle}} x_{\text{beetle}} + \rho_{\text{cabrio}} x_{\text{cabrio}} \leq R$

(time spent in each department) $t_{\text{beetle},d} x_{\text{beetle}} + t_{\text{cabrio},d} x_{\text{cabrio}} \leq T_d$ for all $d \in D$

V

(integrality of variables)

$x_c \geq 0$ for all $c \in C$

x_c integer for all $c \in C$

Objective function: maximize (total revenue) $r_{\text{beetle}} \cdot x_{\text{beetle}} + r_{\text{cabrio}} \cdot x_{\text{cabrio}}$

C

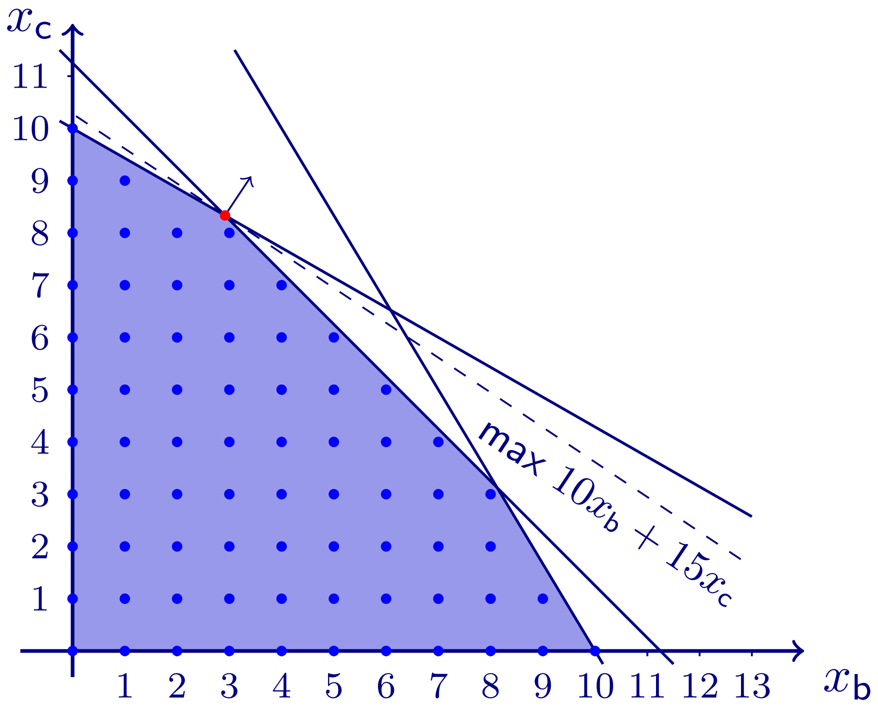
(total raw material available) $\rho_{\text{beetle}} x_{\text{beetle}} + \rho_{\text{cabrio}} x_{\text{cabrio}} \leq R$

(time spent in each department) $t_{\text{beetle},d} x_{\text{beetle}} + t_{\text{cabrio},d} x_{\text{cabrio}} \leq T_d$ for all $d \in D$

V

$x_c \geq 0$ for all $c \in C$

(integrality of variables) x_c integer for all $c \in C$



Objective function: maximize (total revenue) $r_{\text{beetle}} \cdot x_{\text{beetle}} + r_{\text{cabrio}} \cdot x_{\text{cabrio}}$

C

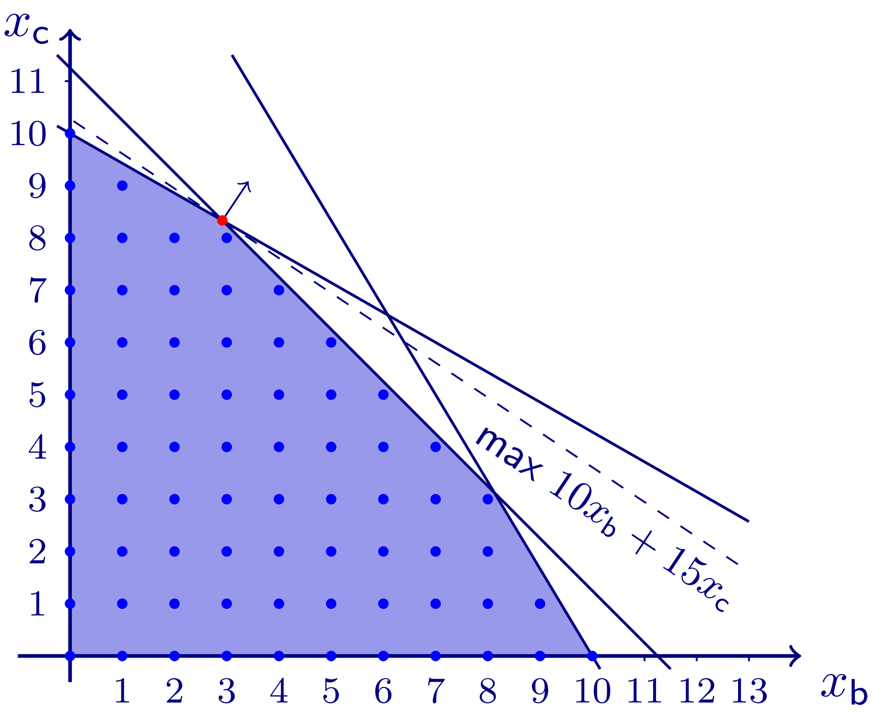
(total raw material available) $\rho_{\text{beetle}} x_{\text{beetle}} + \rho_{\text{cabrio}} x_{\text{cabrio}} \leq R$

(time spent in each department) $t_{\text{beetle},d} x_{\text{beetle}} + t_{\text{cabrio},d} x_{\text{cabrio}} \leq T_d$ for all $d \in D$

V

$x_c \geq 0$ for all $c \in C$

(integrality of variables) x_c integer for all $c \in C$



➡ optimal solution of LP relaxation:

$$(x_b, x_c) = \left(\frac{35}{12}, \frac{25}{3}\right)$$

$$\approx (2.91667, 8.33333)$$

Objective function: maximize (total revenue) $r_{\text{beetle}} \cdot x_{\text{beetle}} + r_{\text{cabrio}} \cdot x_{\text{cabrio}}$

C

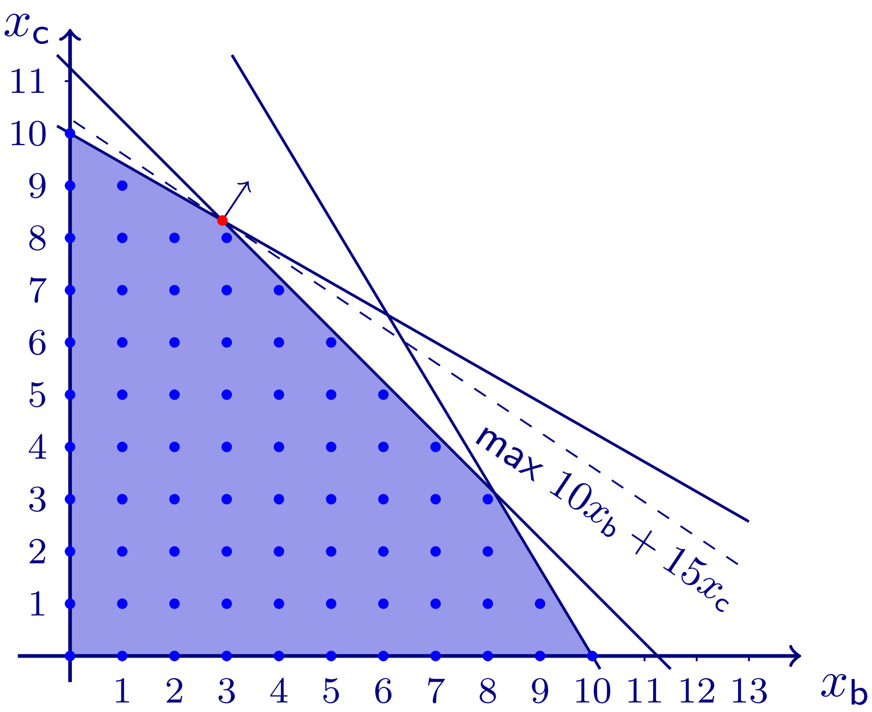
(total raw material available) $\rho_{\text{beetle}} x_{\text{beetle}} + \rho_{\text{cabrio}} x_{\text{cabrio}} \leq R$

(time spent in each department) $t_{\text{beetle},d} x_{\text{beetle}} + t_{\text{cabrio},d} x_{\text{cabrio}} \leq T_d$ for all $d \in D$

V

$x_c \geq 0$ for all $c \in C$

(integrality of variables) x_c integer for all $c \in C$

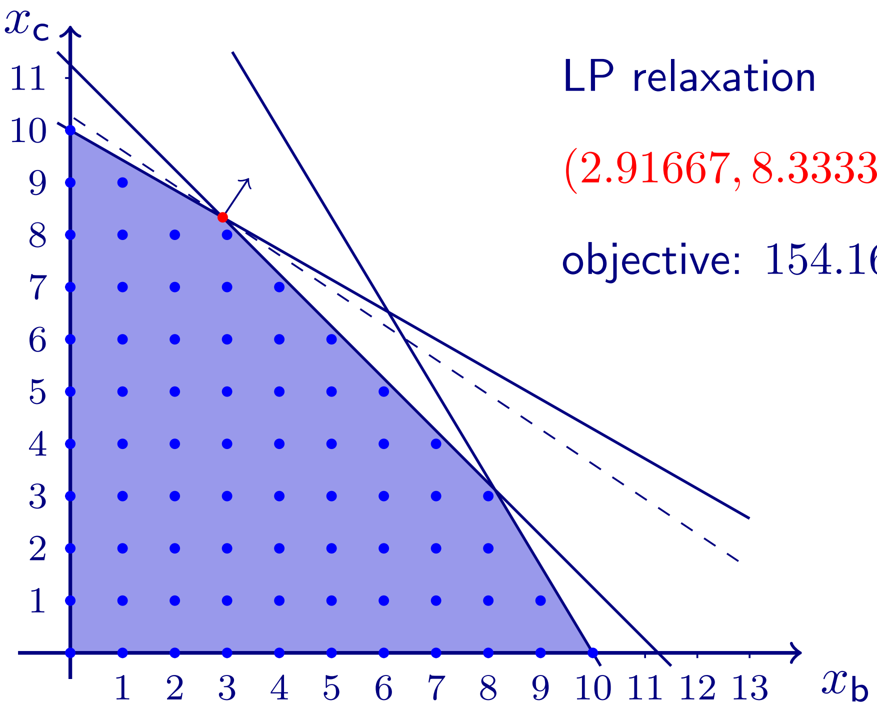


➡ optimal solution of LP relaxation:

$$(x_b, x_c) = \left(\frac{35}{12}, \frac{25}{3}\right)$$

$$\approx (2.91667, 8.33333)$$

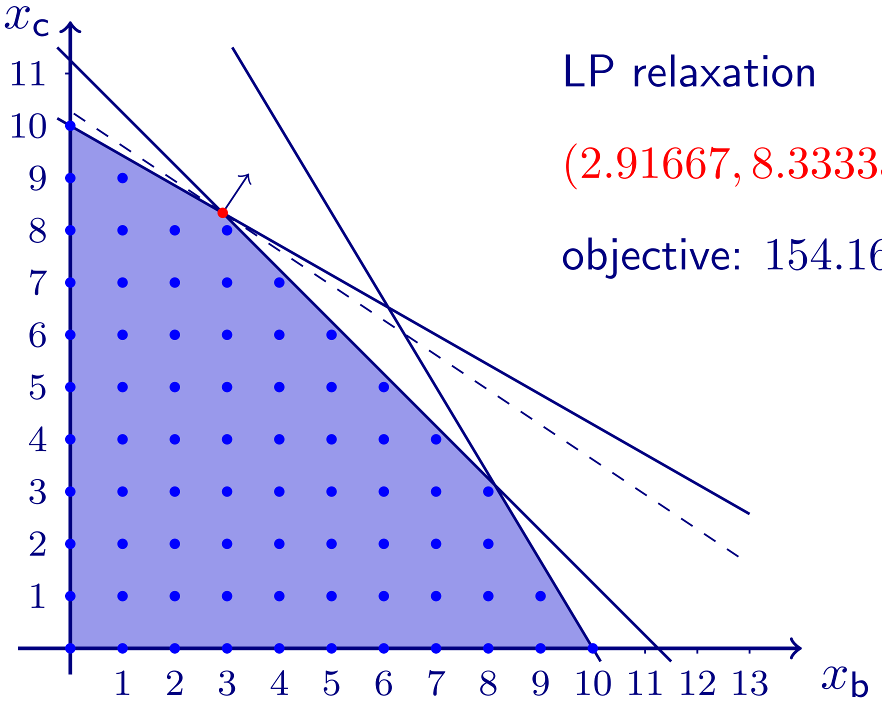
➡ fractional!



LP relaxation

$(2.91667, 8.33333)$

objective: 154.167



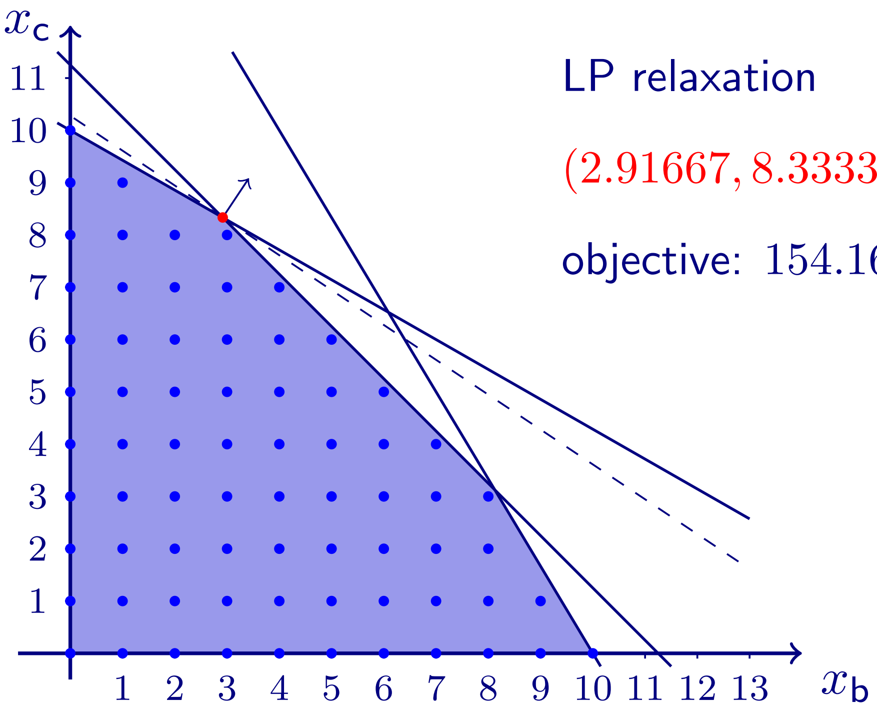
LP relaxation

$(2.91667, 8.33333)$

objective: 154.167

154.167

154.167



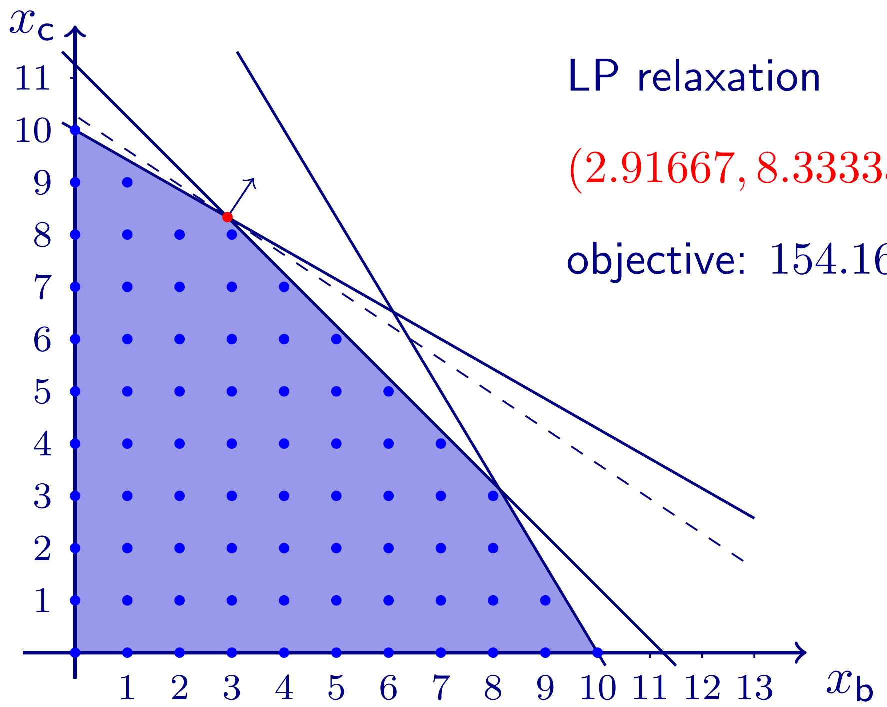
LP relaxation

$(2.91667, 8.33333)$

objective: 154.167

➡ IP optimum cannot be higher than 154.167

154.167

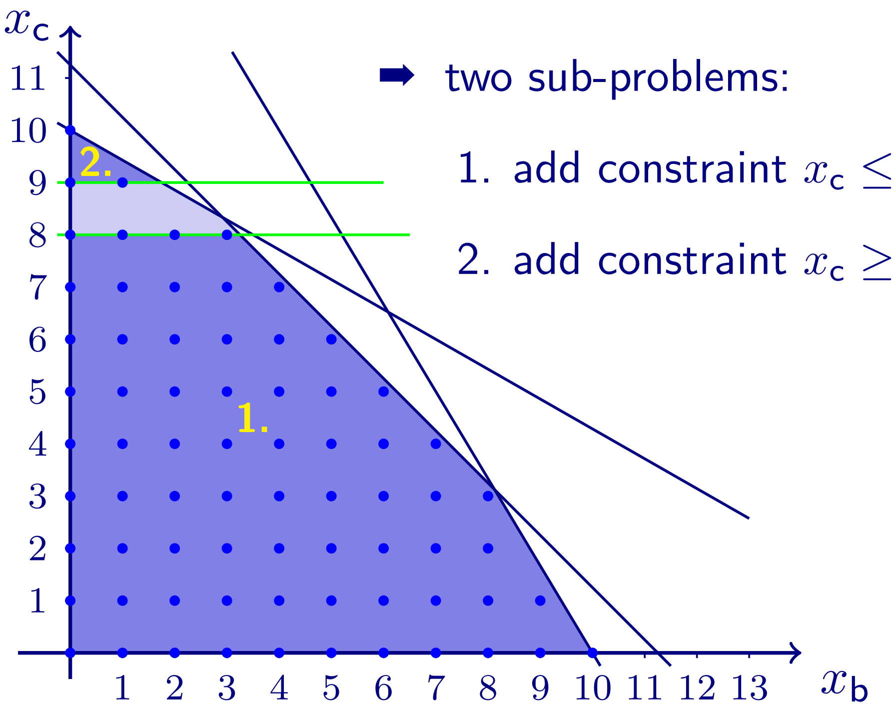


LP relaxation

$(2.91667, 8.33333)$

objective: 154.167

- ➡ IP optimum cannot be higher than 154.167
- ➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$



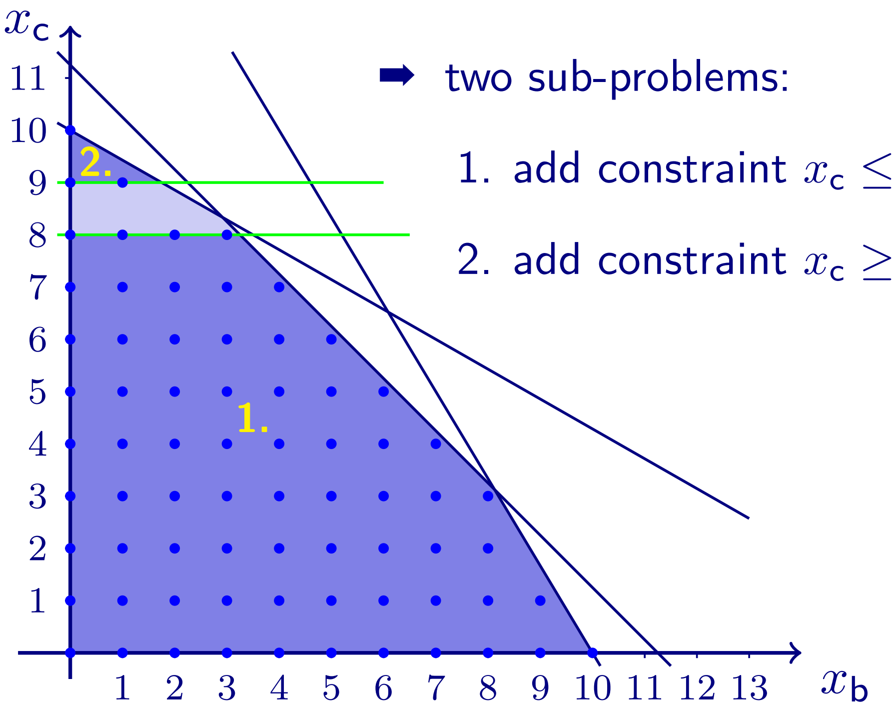
➡ two sub-problems:

1. add constraint $x_c \leq 8$
2. add constraint $x_c \geq 9$

154.167

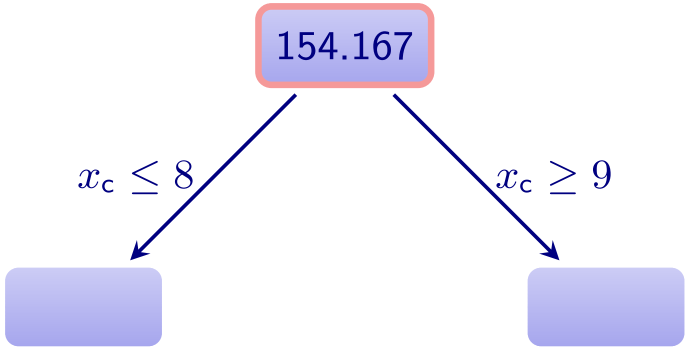
➡ IP optimum cannot be higher than 154.167

➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$



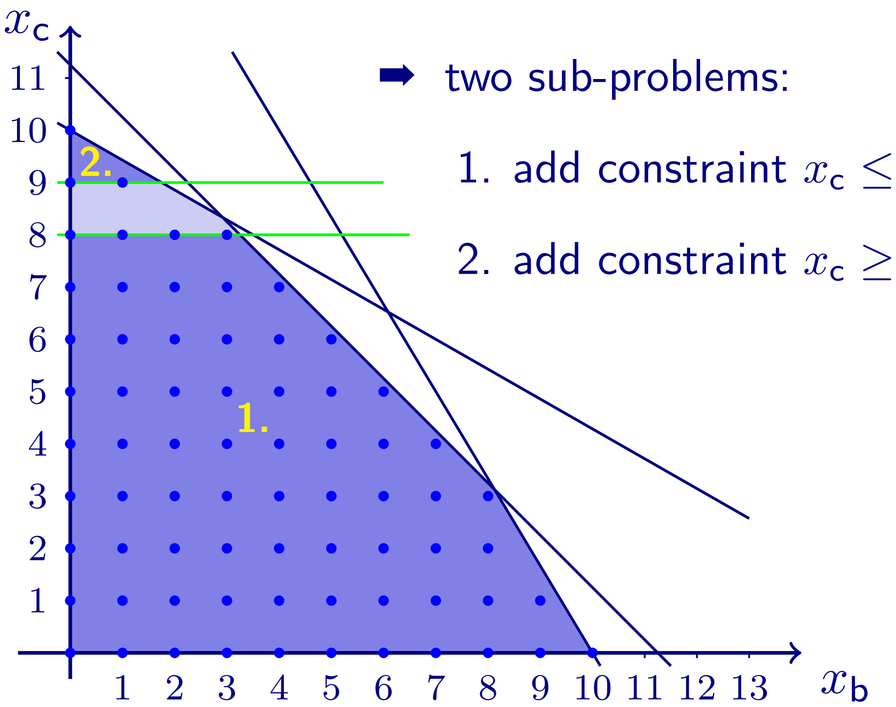
➡ two sub-problems:

1. add constraint $x_c \leq 8$
2. add constraint $x_c \geq 9$



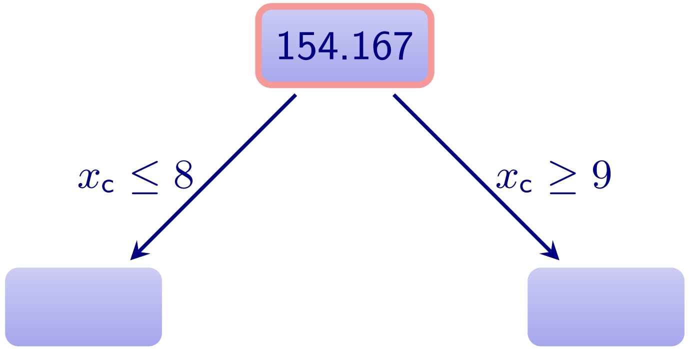
➡ IP optimum cannot be higher than 154.167

➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$



➡ two sub-problems:

1. add constraint $x_c \leq 8$
2. add constraint $x_c \geq 9$



➡ IP optimum cannot be higher than 154.167

➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$

➡ To be continued...

- We know how to solve linear programs efficiently

▷ We know how to solve linear programs efficiently

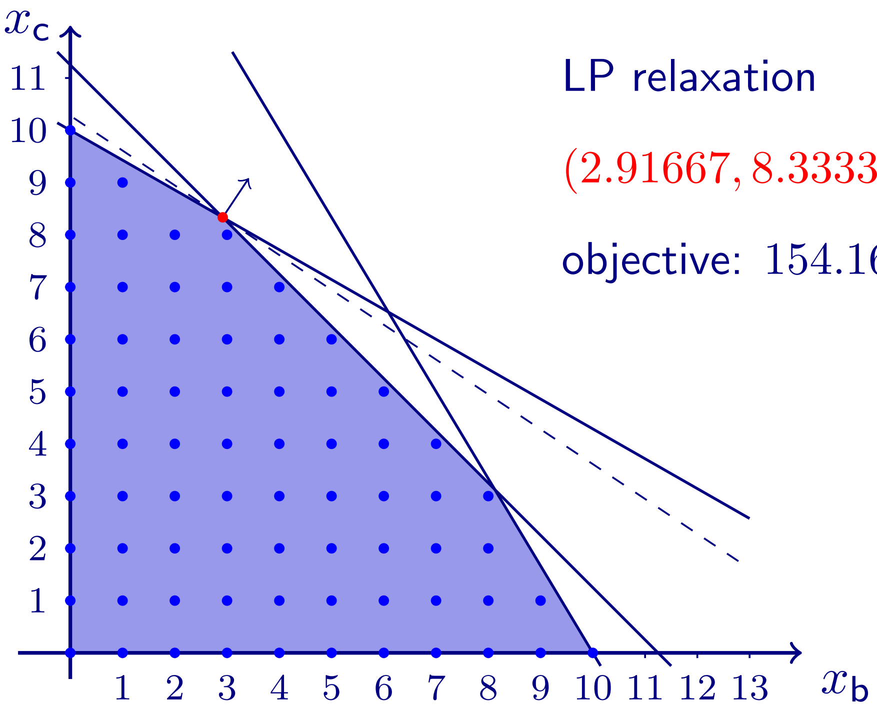
➡ Solving the LP relaxation gives some information on the MIP:

- a (probably) fractional solution of the LP
- an upper bound on the MIP optimum (for maximizing!)

▷ If the solution has integer values in all required entries, we are done!

- We know how to solve linear programs efficiently
  - ➡ Solving the LP relaxation gives some information on the MIP:
    - a (probably) fractional solution of the LP
    - an upper bound on the MIP optimum (for maximizing!)
- If the solution has integer values in all required entries, we are done!
- Otherwise there is a fractional entry $x_i = q \notin \mathbb{Z}$ which should be integer
  - ➡ All possible solutions to the MIP have either $x_i \leq \lfloor q \rfloor$ or $x_i \geq \lceil q \rceil$

- We know how to solve linear programs efficiently
  - ➡ Solving the LP relaxation gives some information on the MIP:
    - a (probably) fractional solution of the LP
    - an upper bound on the MIP optimum (for maximizing!)
- If the solution has integer values in all required entries, we are done!
- Otherwise there is a fractional entry $x_i = q \notin \mathbb{Z}$ which should be integer
  - ➡ All possible solutions to the MIP have either $x_i \leq \lfloor q \rfloor$ or $x_i \geq \lceil q \rceil$
- Idea: split up the MIP into two sub-problems:
  - add the constraint $x_i \leq \lfloor q \rfloor$ for the first sub-problem
  - add the constraint $x_i \geq \lceil q \rceil$ for the second sub-problem
  - ➡ Problem is solved if both sub-problems are solved
  - ➡ Solve the sub-problems recursively, in the same way

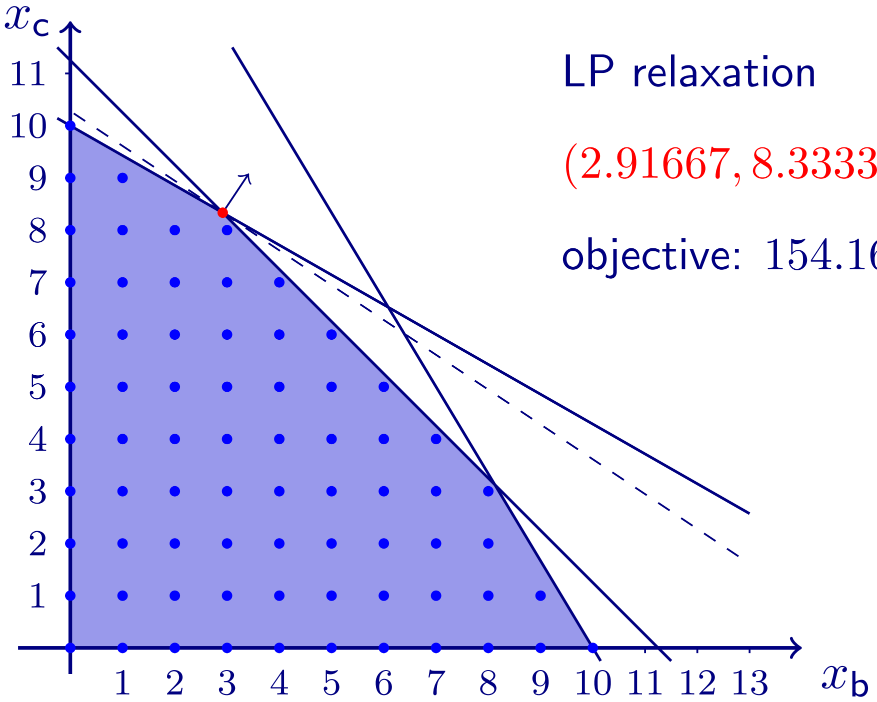


LP relaxation

$(2.91667, 8.33333)$

objective: 154.167

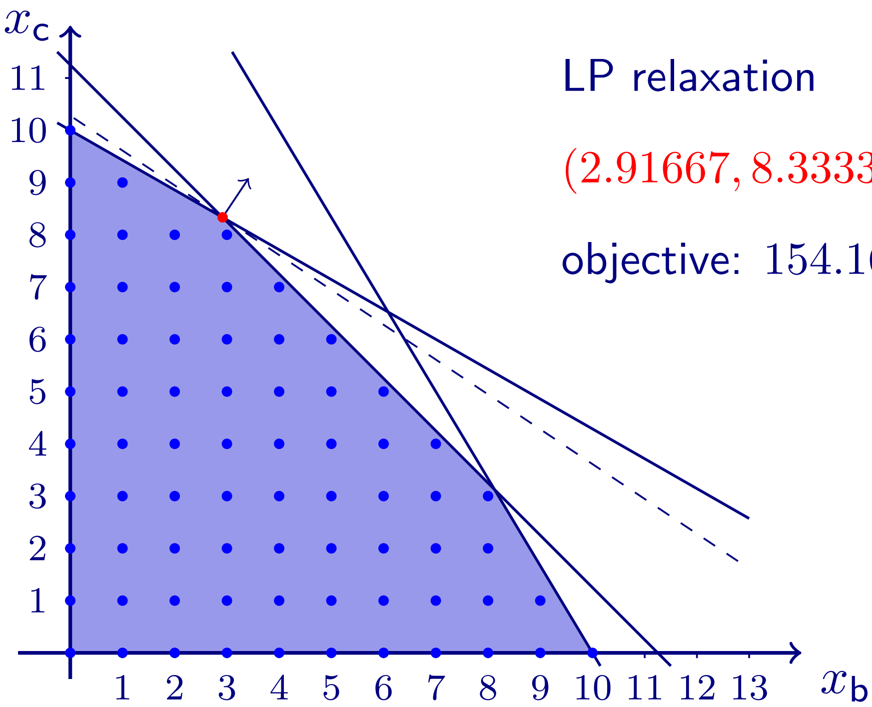
154.167



LP relaxation

$(2.91667, 8.33333)$

objective: 154.167



LP relaxation

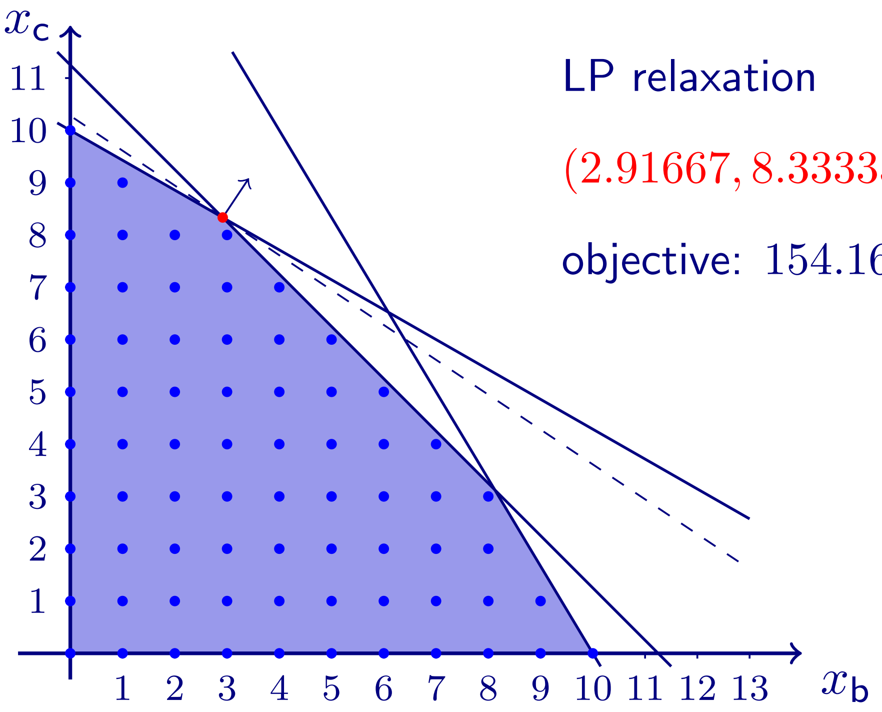
$(2.91667, 8.33333)$

objective: 154.167

154.167

➡ IP optimum cannot be higher than 154.167

154.167



LP relaxation

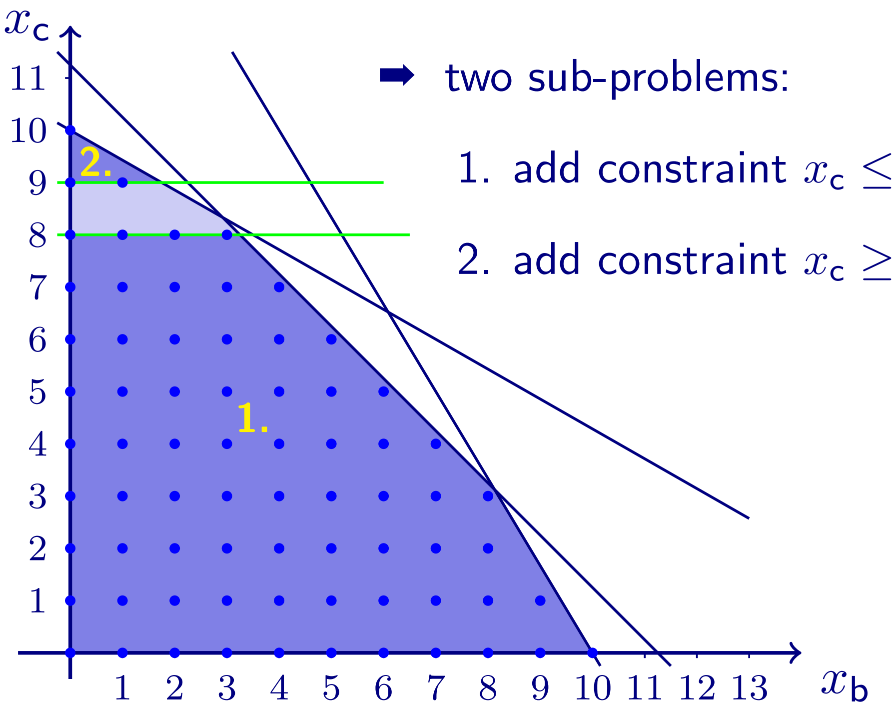
$(2.91667, 8.33333)$

objective: 154.167

➡ IP optimum cannot be higher than 154.167

➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$

154.167

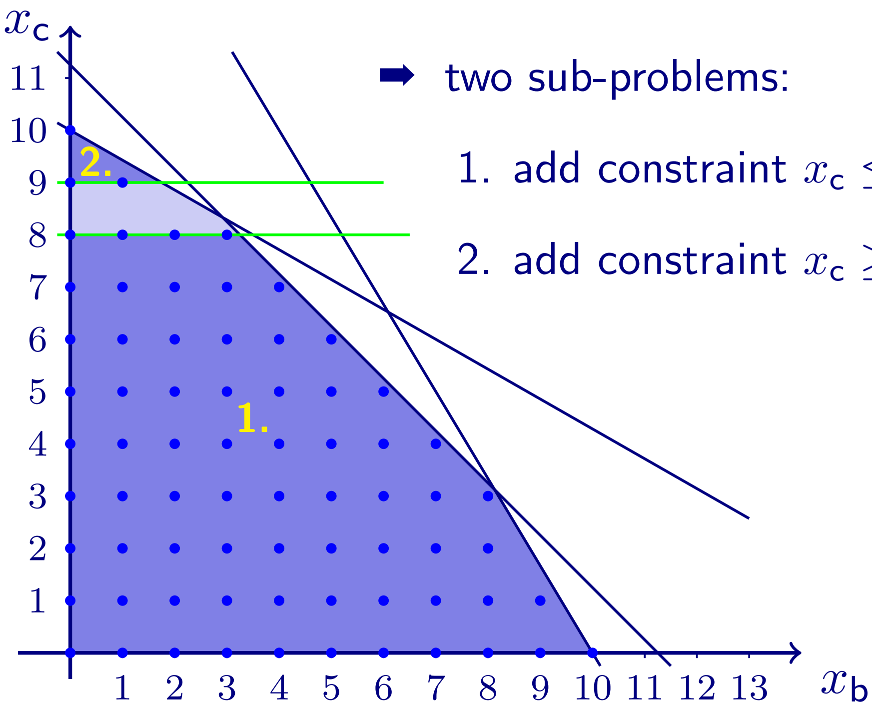


➡ two sub-problems:

1. add constraint $x_c \leq 8$
2. add constraint $x_c \geq 9$

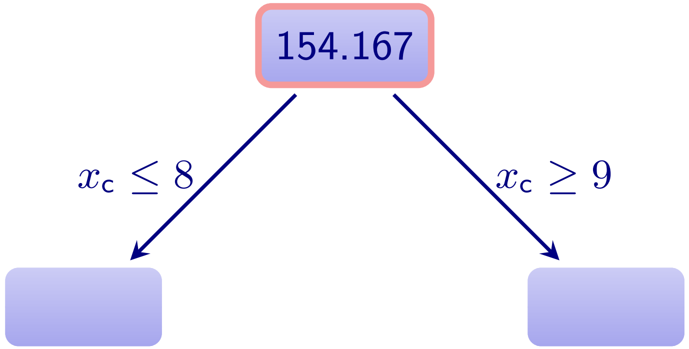
➡ IP optimum cannot be higher than 154.167

➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$



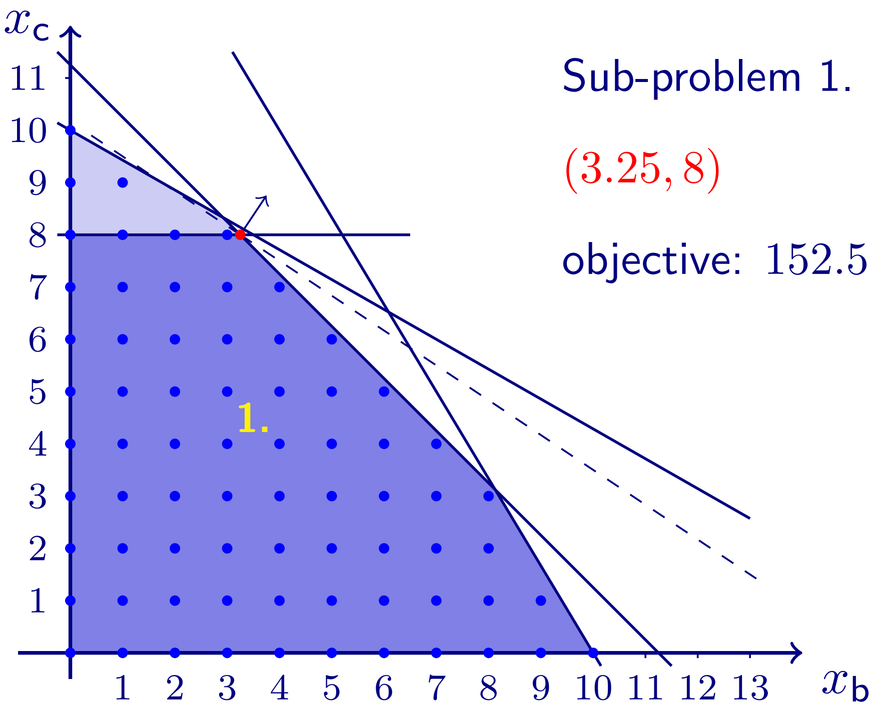
➡ two sub-problems:

1. add constraint $x_c \leq 8$
2. add constraint $x_c \geq 9$



➡ IP optimum cannot be higher than 154.167

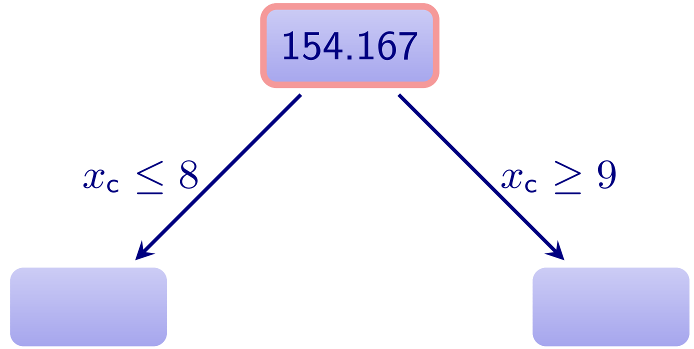
➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$



Sub-problem 1.

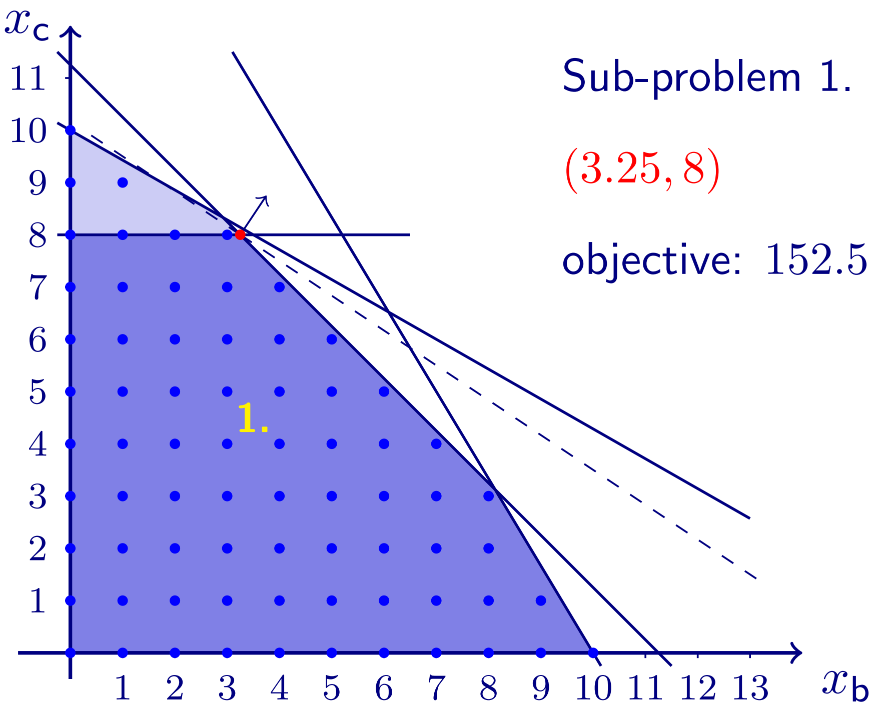
$(3.25, 8)$

objective: 152.5



➡ IP optimum cannot be higher than 154.167

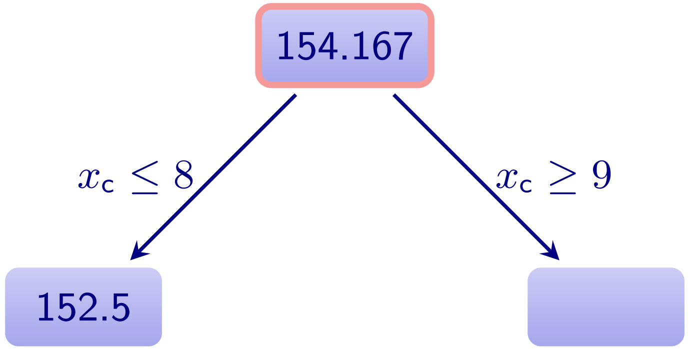
➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$



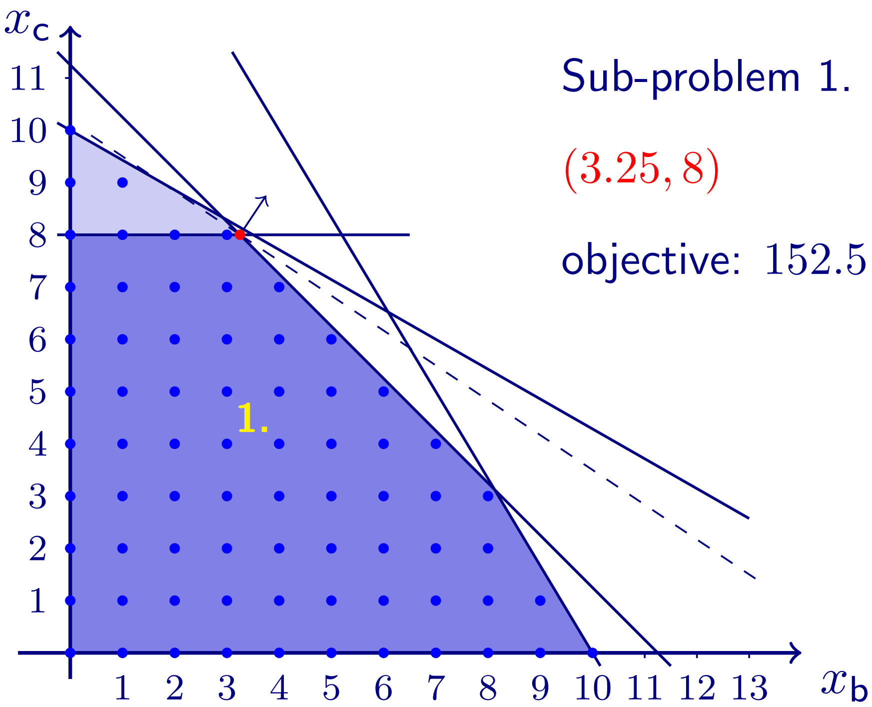
Sub-problem 1.

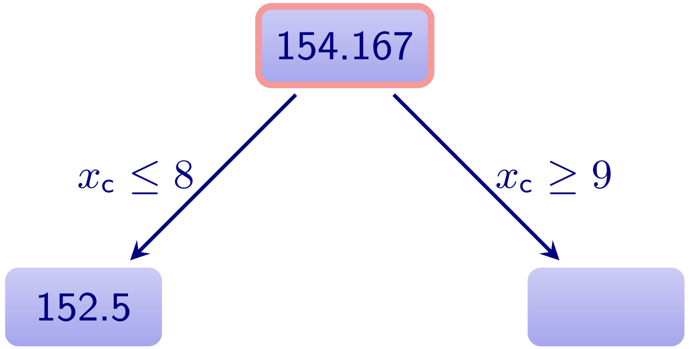
$(3.25, 8)$

objective: 152.5



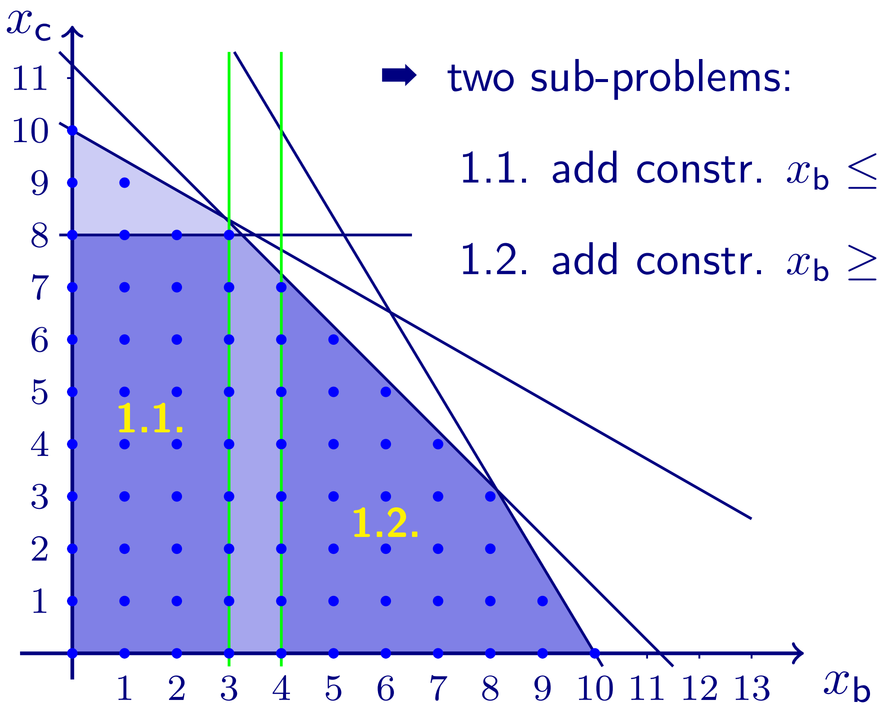
- ➡ IP optimum cannot be higher than 154.167
- ➡ Optimal IP solution has either $x_c \leq 8$ or $x_c \geq 9$





➡ IP optimum cannot be higher than 154.167

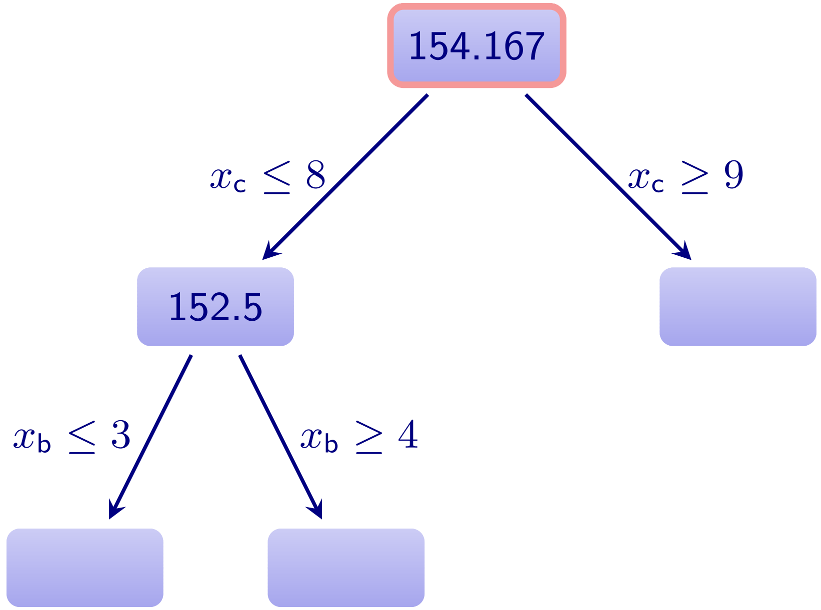
➡ Optimal IP solution of sub-problem 1 has either $x_b \leq 3$ or $x_b \geq 4$



➡ two sub-problems:

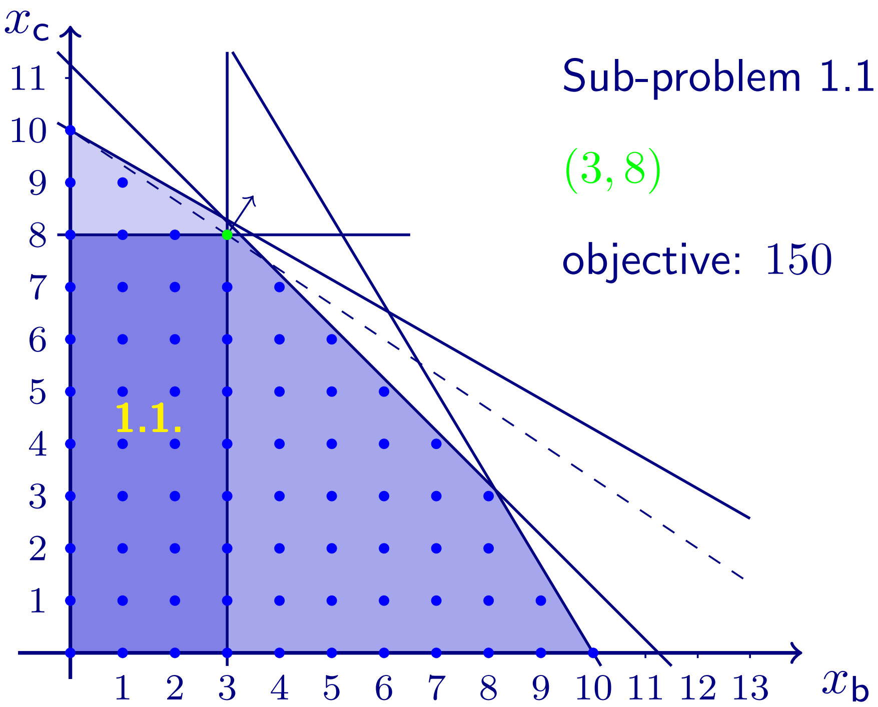
1.1. add constr. $x_b \leq 3$

1.2. add constr. $x_b \geq 4$



➡ IP optimum cannot be higher than 154.167

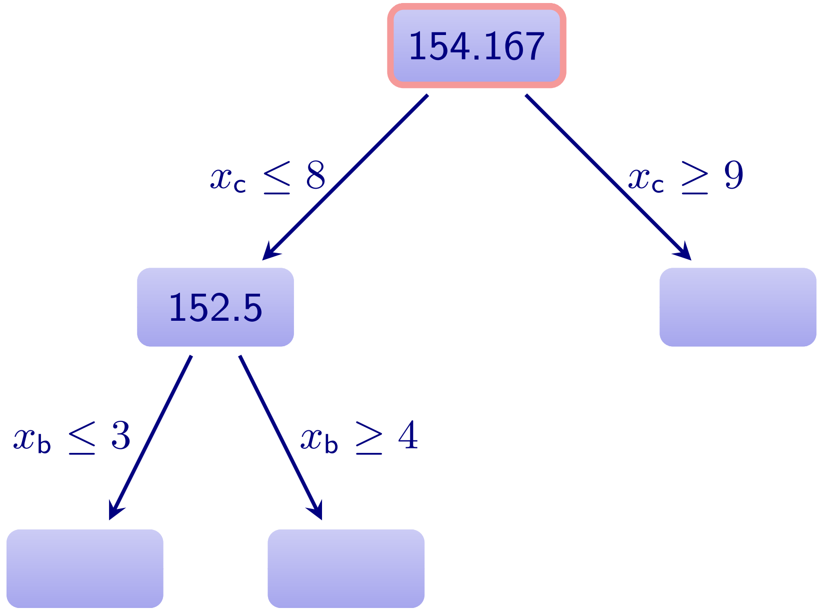
➡ Optimal IP solution of sub-problem 1 has either $x_b \leq 3$ or $x_b \geq 4$



Sub-problem 1.1

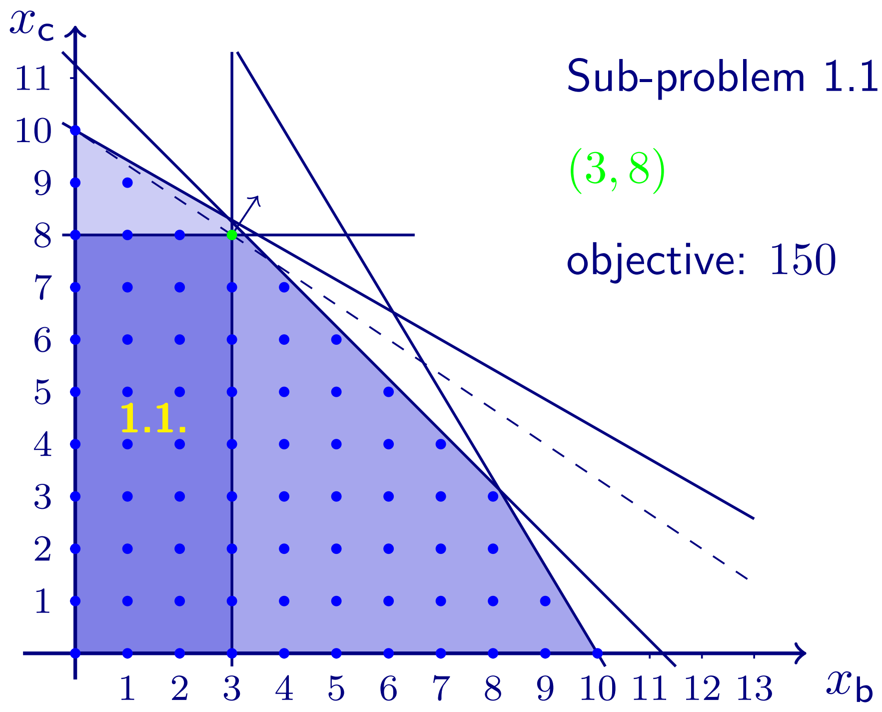
$(3, 8)$

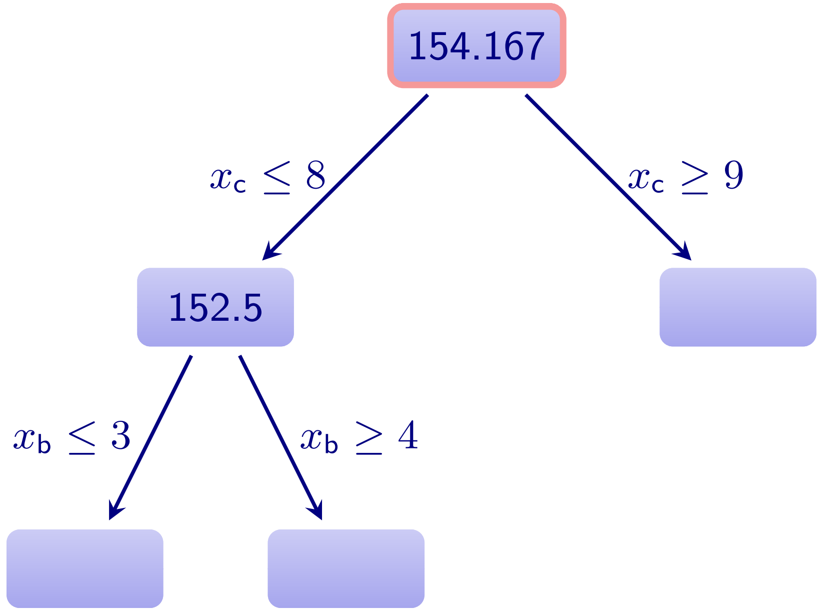
objective: 150



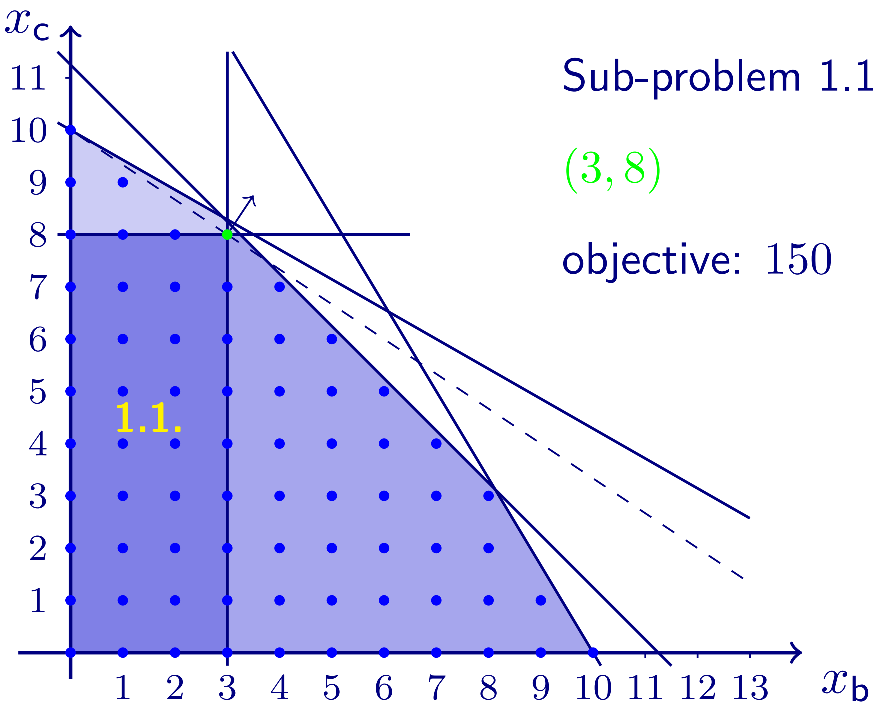
➡ IP optimum cannot be higher than 154.167

➡ Optimal IP solution of sub-problem 1 has either $x_b \leq 3$ or $x_b \geq 4$





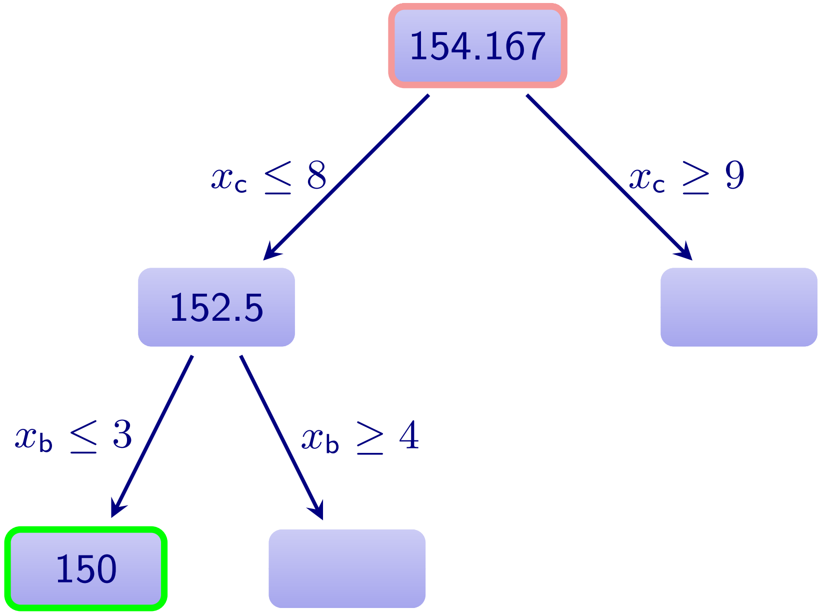
- ➡ IP optimum cannot be higher than 154.167
- ➡ IP optimum is at least 150 (objective value of the current incumbent)



Sub-problem 1.1

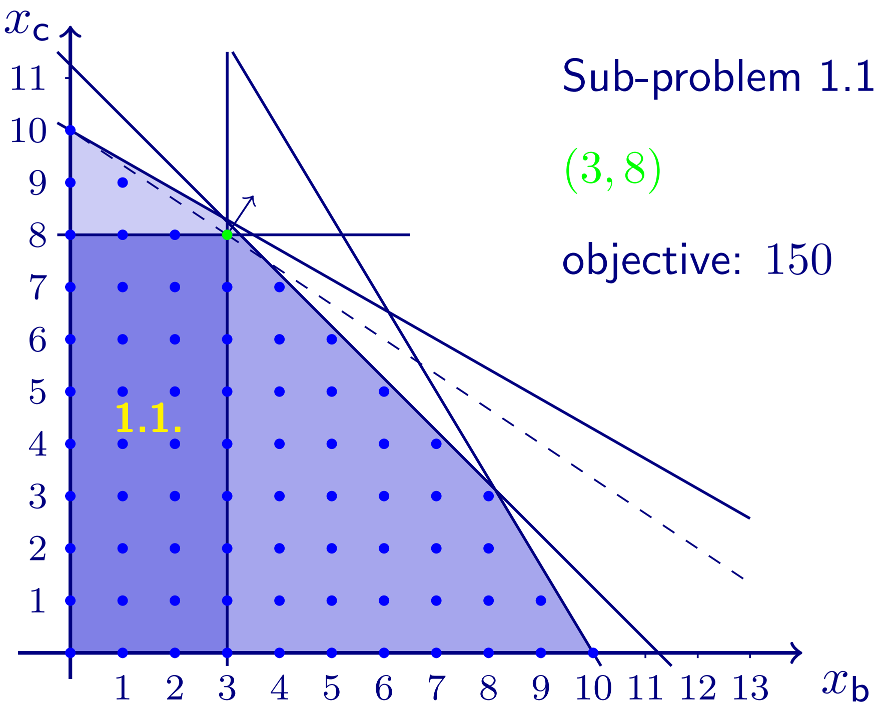
$(3, 8)$

objective: 150



➡ IP optimum cannot be higher than 154.167

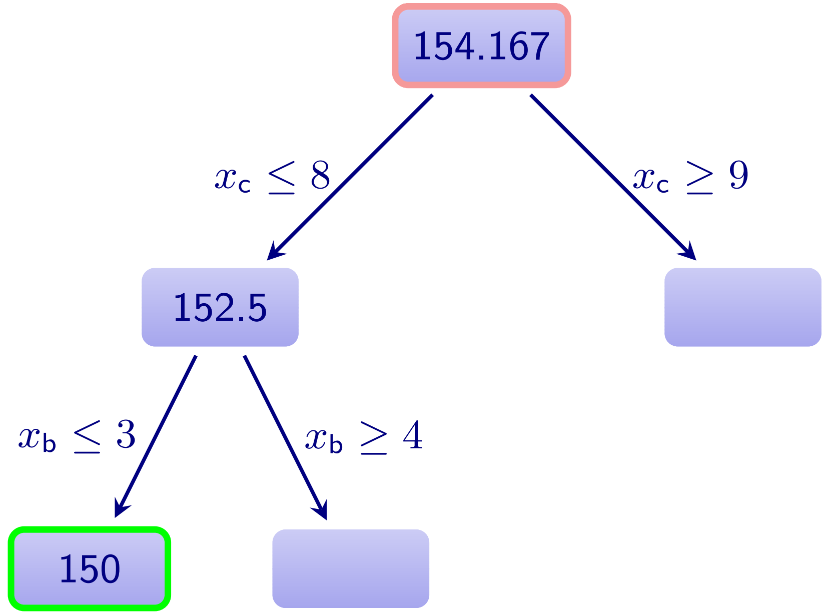
➡ IP optimum is at least 150 (objective value of the current incumbent)



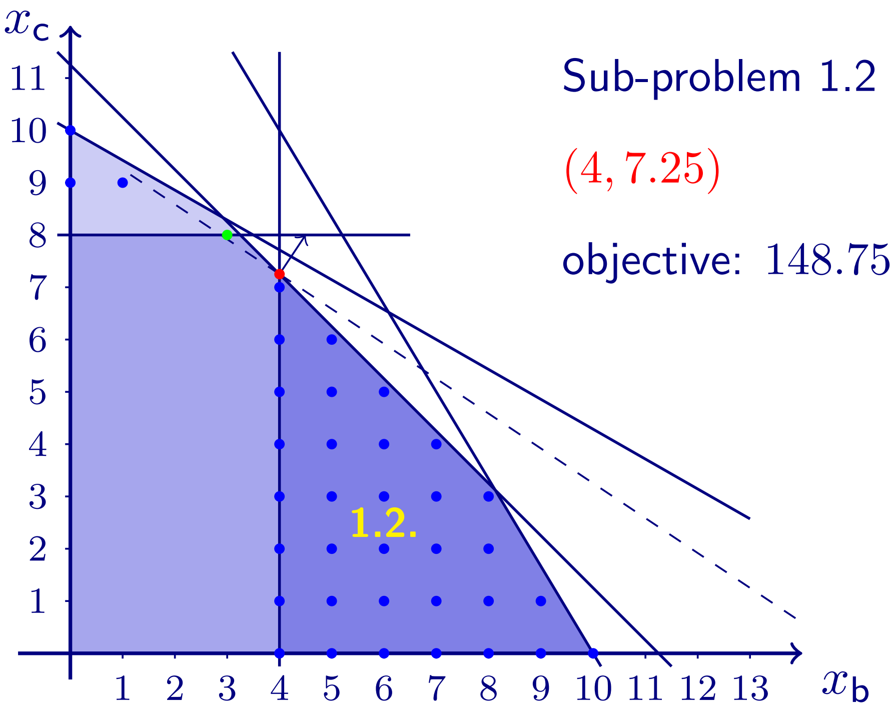
Sub-problem 1.1

$(3, 8)$

objective: 150



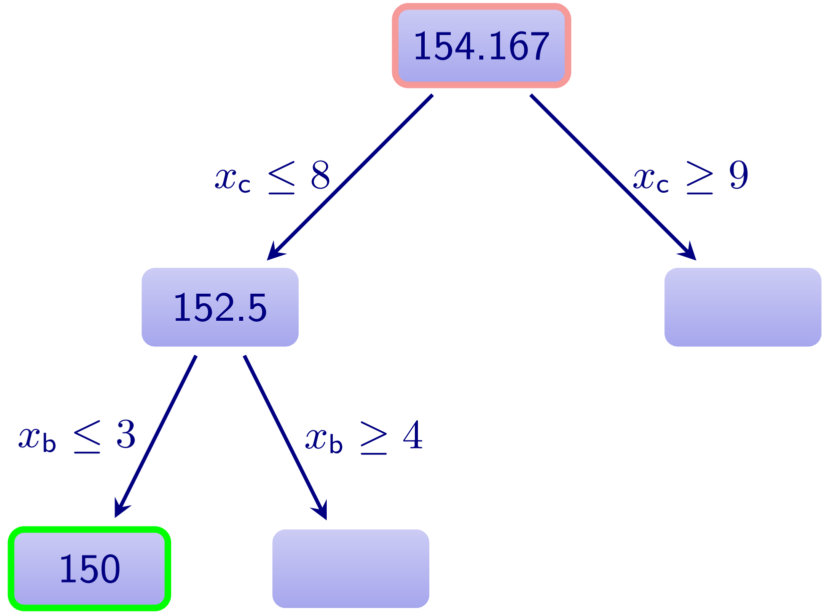
- ➡ IP optimum cannot be higher than 154.167
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All remaining integer points in feasible region 1.1. can be ignored



Sub-problem 1.2

$(4, 7.25)$

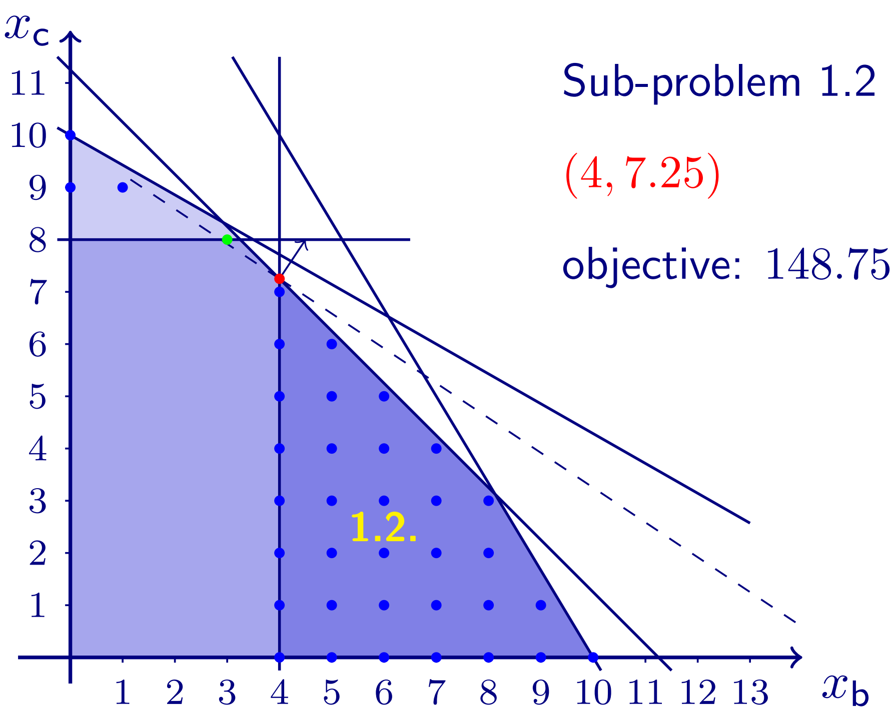
objective: 148.75



➡ IP optimum cannot be higher than 154.167

➡ IP optimum is at least 150 (objective value of the current incumbent)

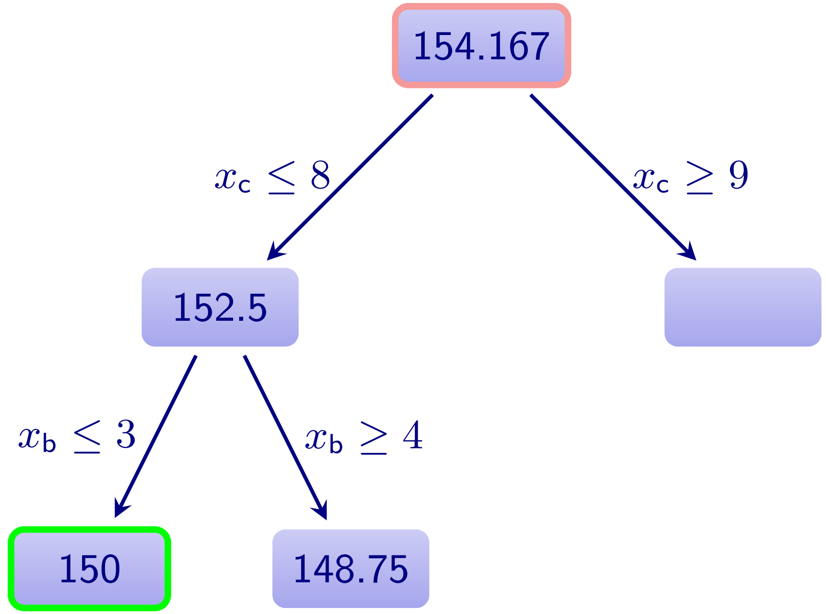
➡ All remaining integer points in feasible region 1.1. can be ignored



Sub-problem 1.2

$(4, 7.25)$

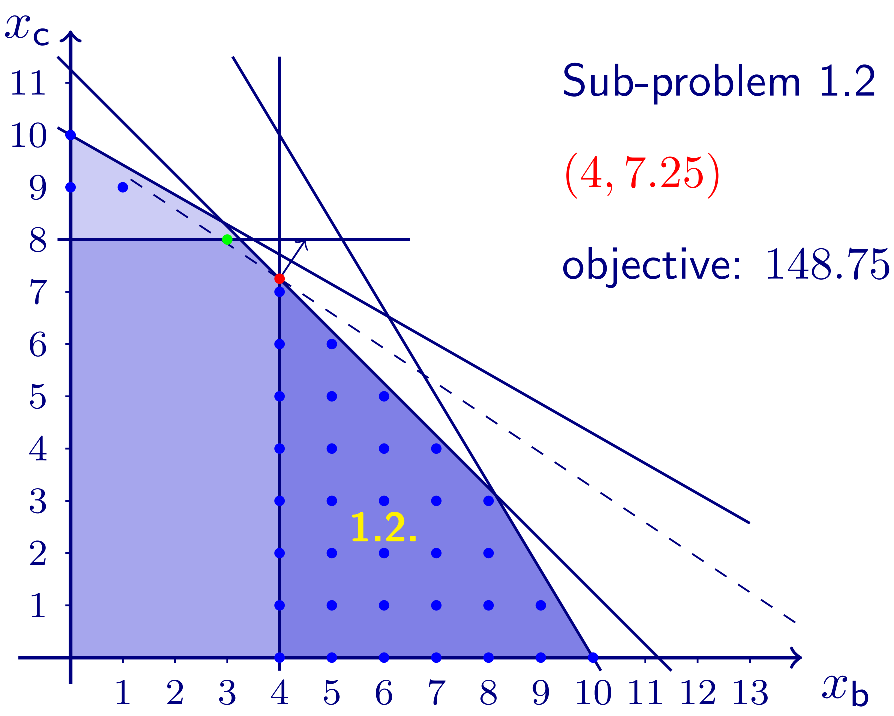
objective: 148.75



➡ IP optimum cannot be higher than 154.167

➡ IP optimum is at least 150 (objective value of the current incumbent)

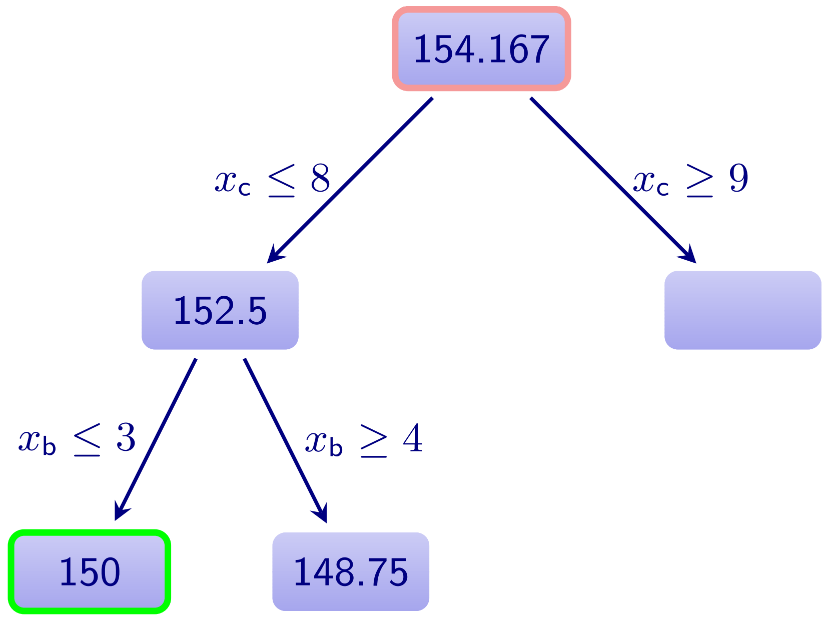
➡ All remaining integer points in feasible region 1.1. can be ignored



Sub-problem 1.2

$(4, 7.25)$

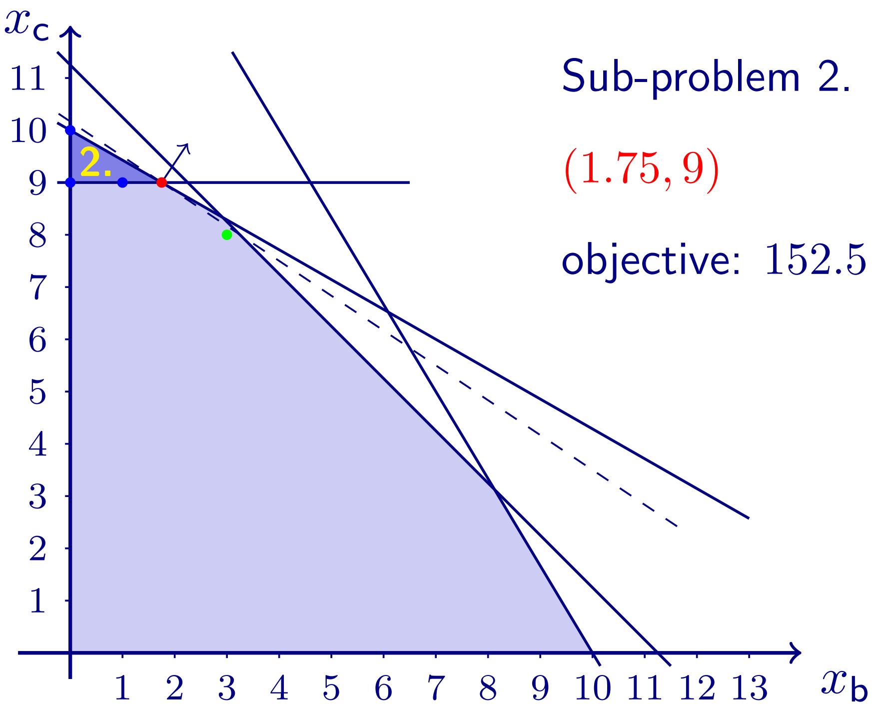
objective: 148.75



➡ IP optimum cannot be higher than 154.167

➡ IP optimum is at least 150 (objective value of the current incumbent)

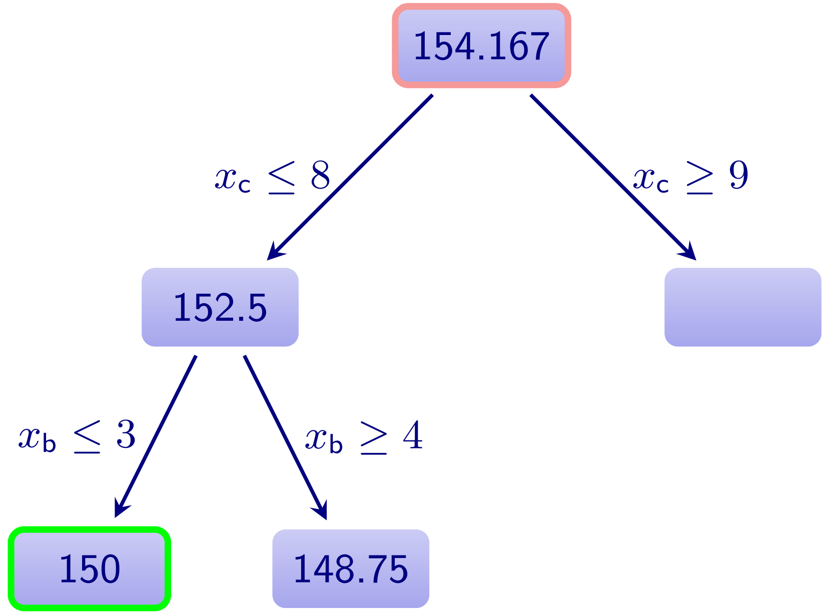
➡ All integer points in feasible region 1.2. can be ignored



Sub-problem 2.

$(1.75, 9)$

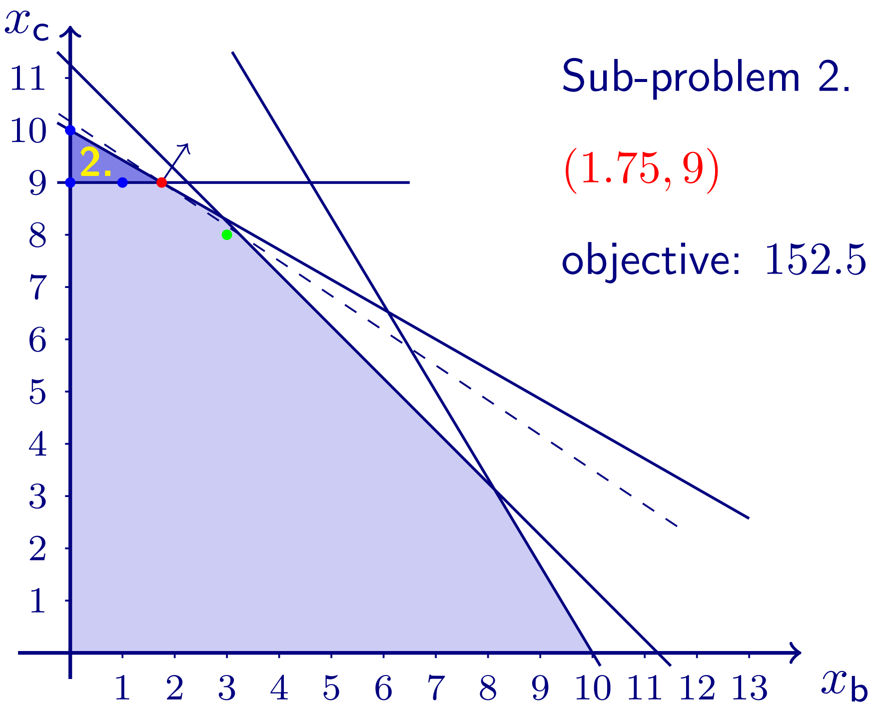
objective: 152.5



➡ IP optimum cannot be higher than 154.167

➡ IP optimum is at least 150 (objective value of the current incumbent)

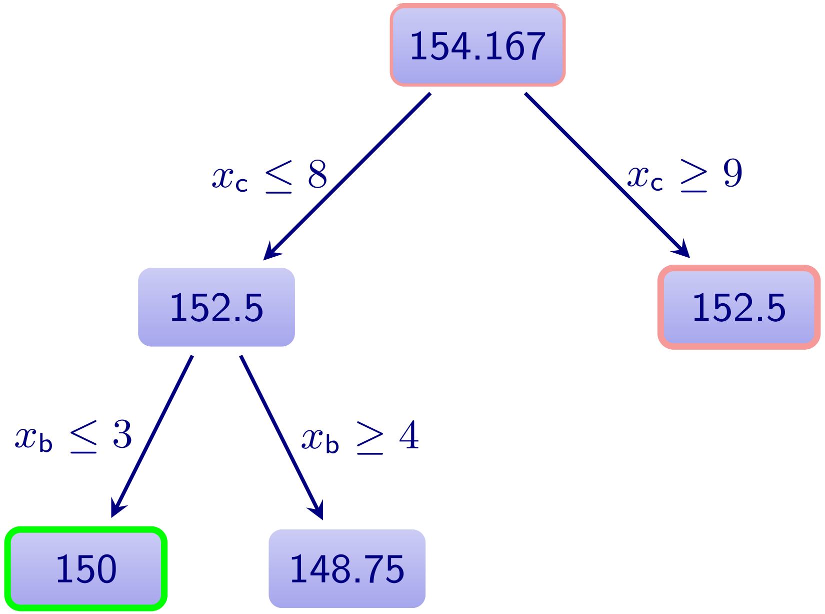
➡ All integer points in feasible region 1.2. can be ignored



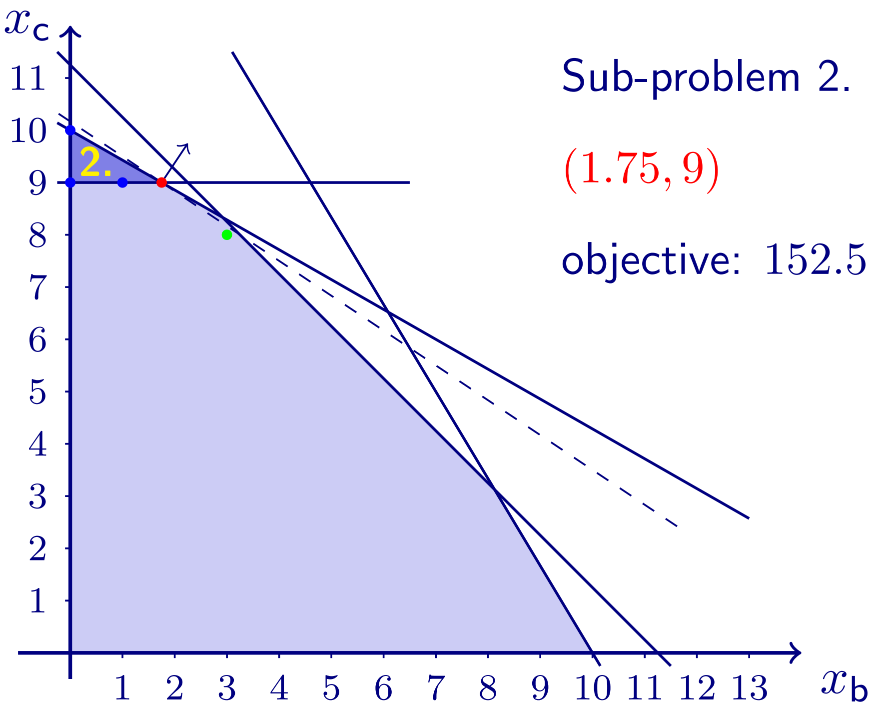
Sub-problem 2.

$(1.75, 9)$

objective: 152.5



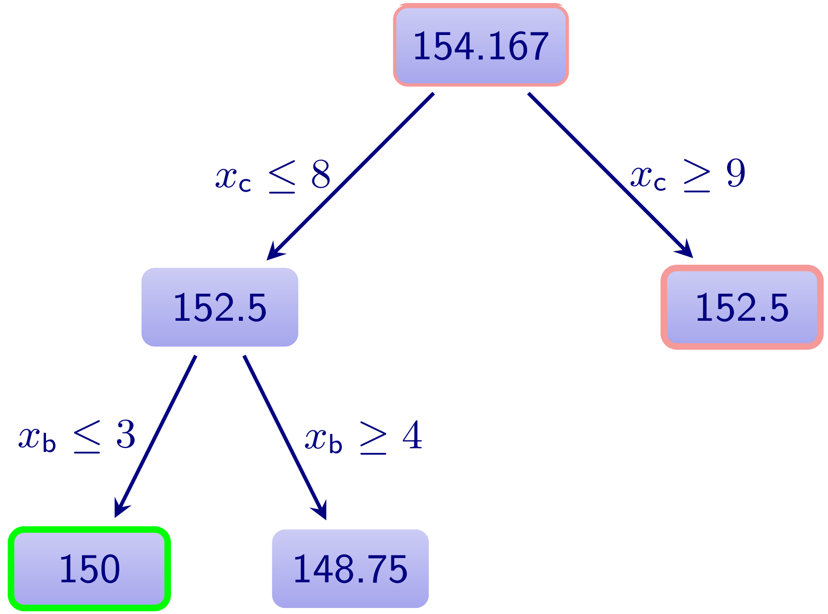
- ➡ IP optimum cannot be higher than 154.167
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored



Sub-problem 2.

$(1.75, 9)$

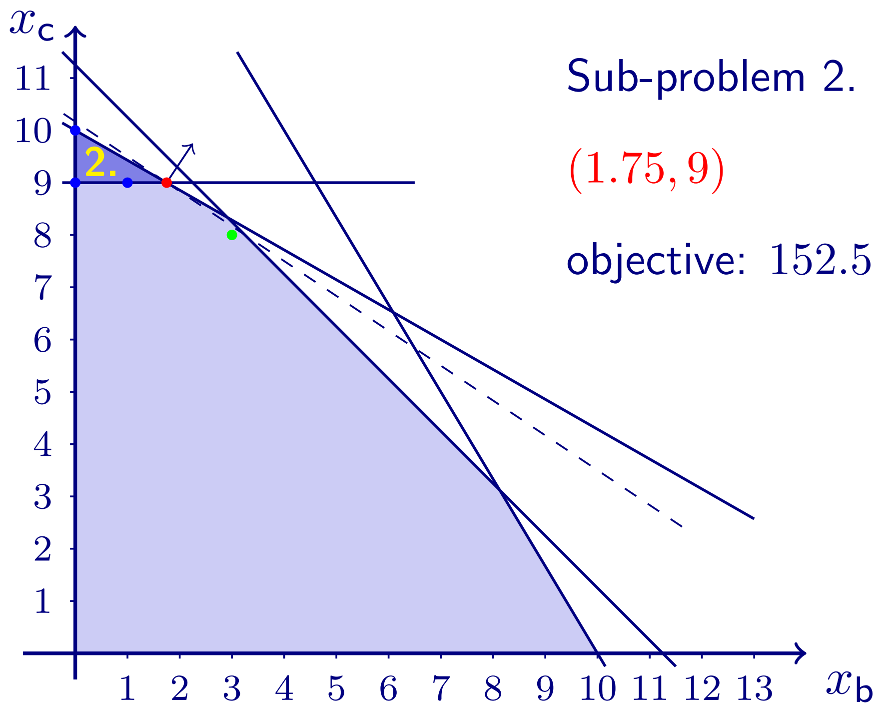
objective: 152.5



➡ IP optimum cannot be higher than 152.5

➡ IP optimum is at least 150 (objective value of the current incumbent)

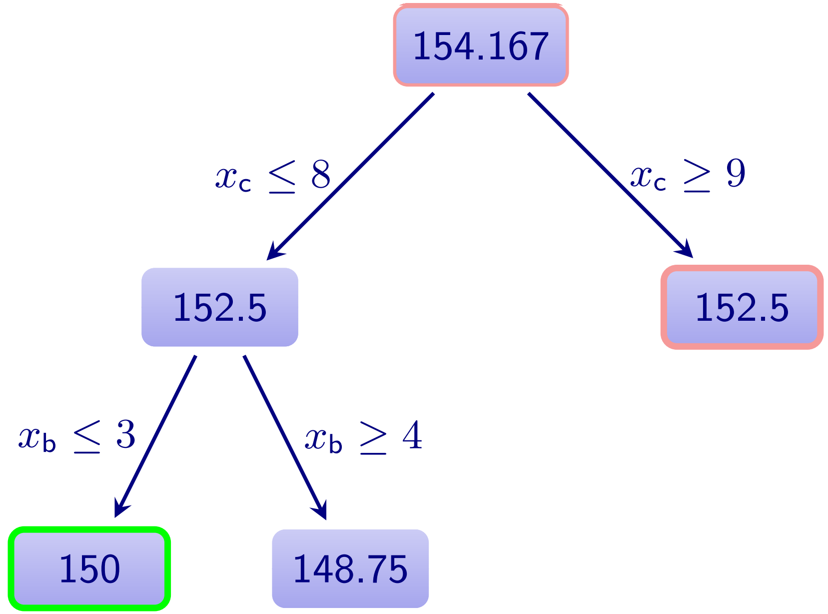
➡ All integer points in feasible region 1.2. can be ignored



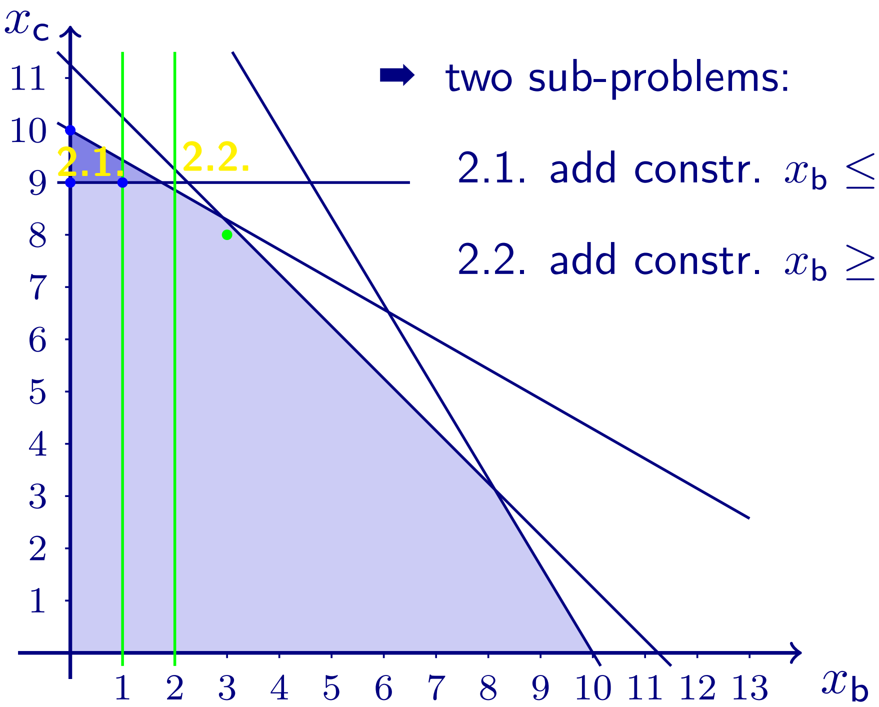
Sub-problem 2.

$(1.75, 9)$

objective: 152.5



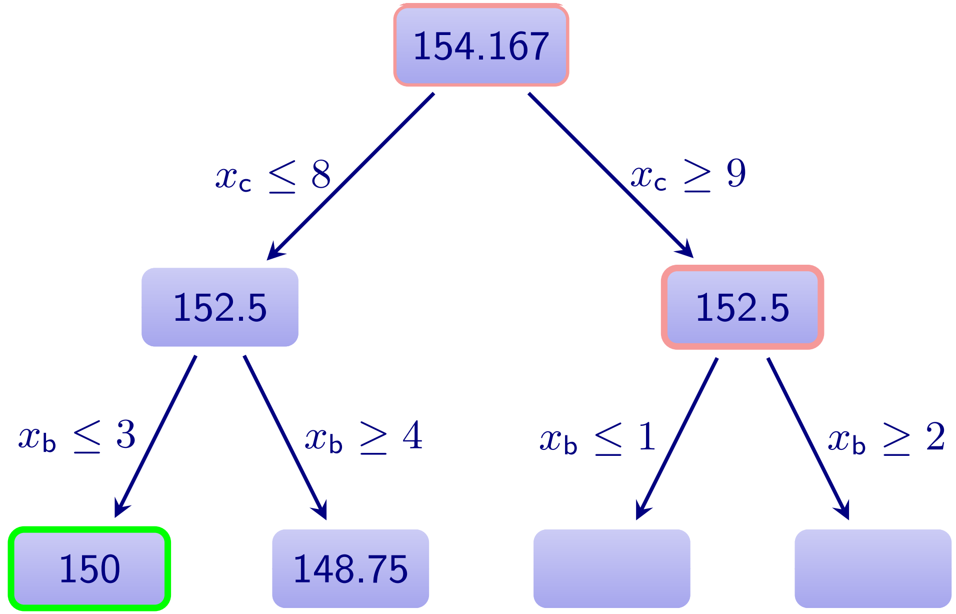
- IP optimum cannot be higher than 152.5
- IP optimum is at least 150 (objective value of the current incumbent)
- All integer points in feasible region 1.2. can be ignored
- Optimal IP solution of sub-problem 2. has either $x_b \leq 1$ or $x_b \geq 2$



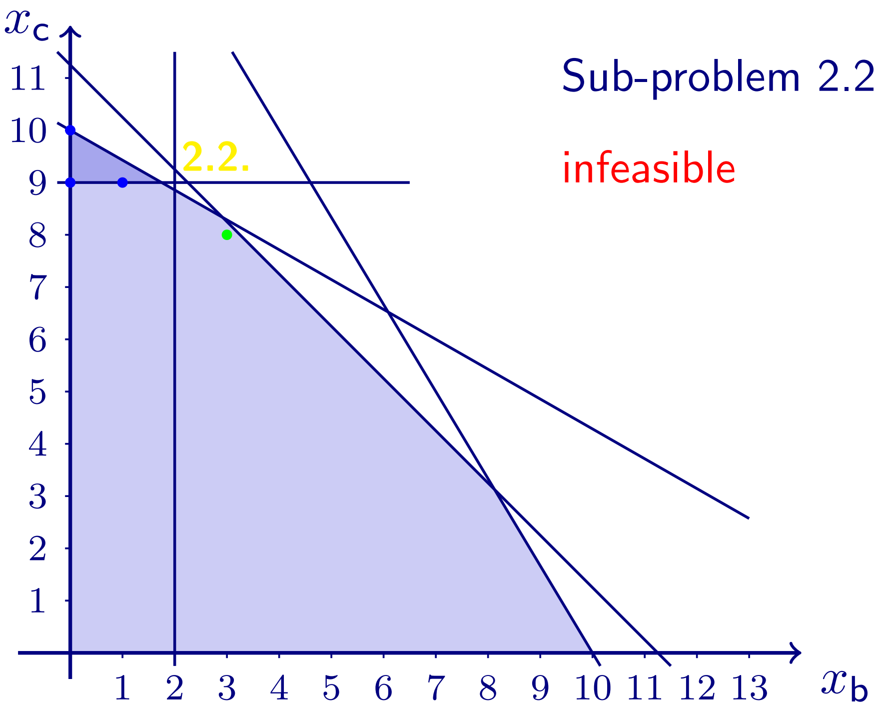
➡ two sub-problems:

2.1. add constr. $x_b \leq 1$

2.2. add constr. $x_b \geq 2$

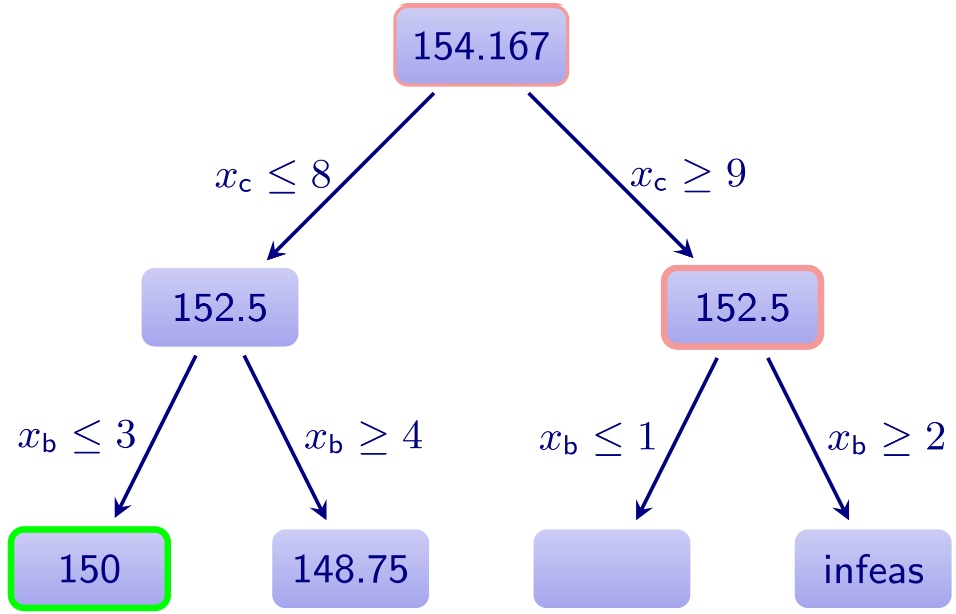


- ➡ IP optimum cannot be higher than 152.5
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored
- ➡ Optimal IP solution of sub-problem 2. has either $x_b \leq 1$ or $x_b \geq 2$

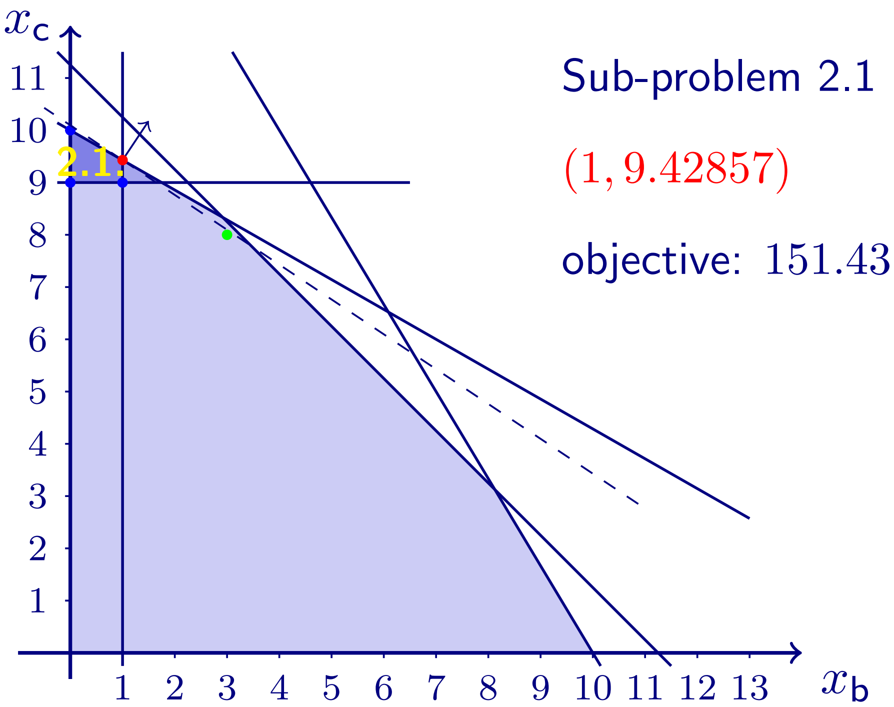


Sub-problem 2.2

infeasible



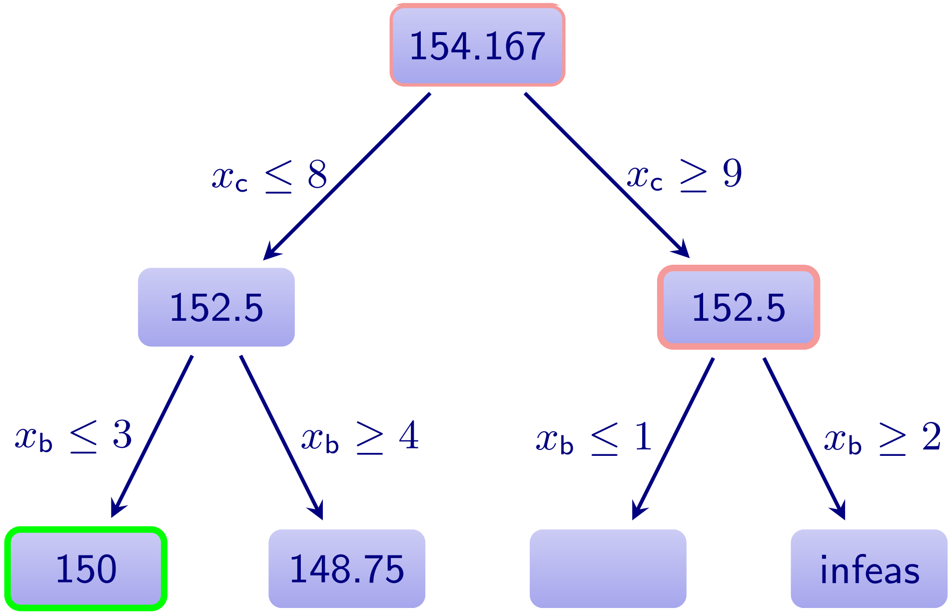
- IP optimum cannot be higher than 152.5
- IP optimum is at least 150 (objective value of the current incumbent)
- All integer points in feasible region 1.2. can be ignored
- Optimal IP solution of sub-problem 2. has either $x_b \leq 1$ or $x_b \geq 2$



Sub-problem 2.1

$(1, 9.42857)$

objective: 151.43

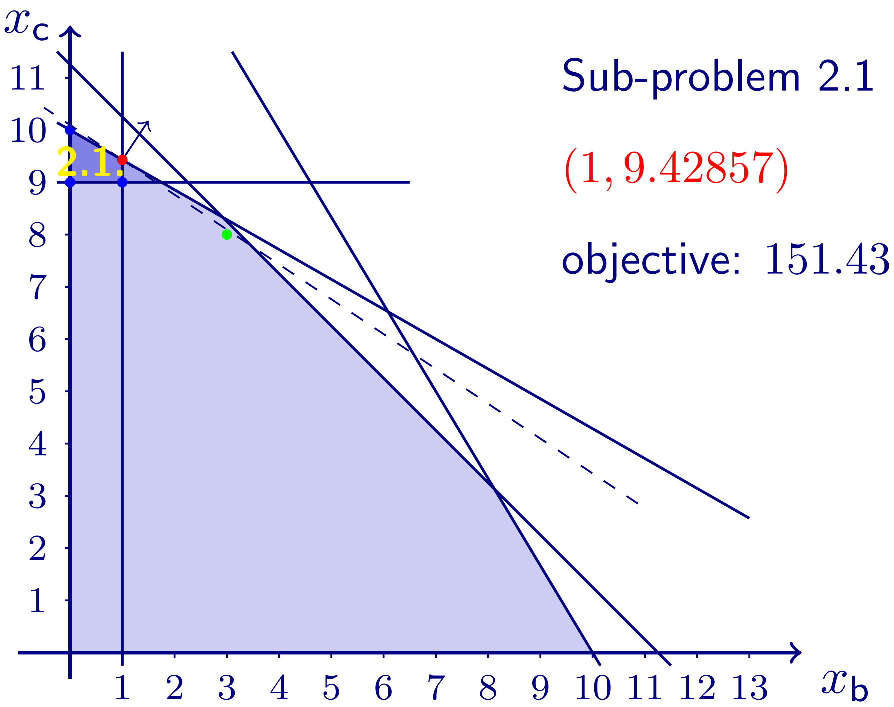


➡ IP optimum cannot be higher than 152.5

➡ IP optimum is at least 150 (objective value of the current incumbent)

➡ All integer points in feasible region 1.2. can be ignored

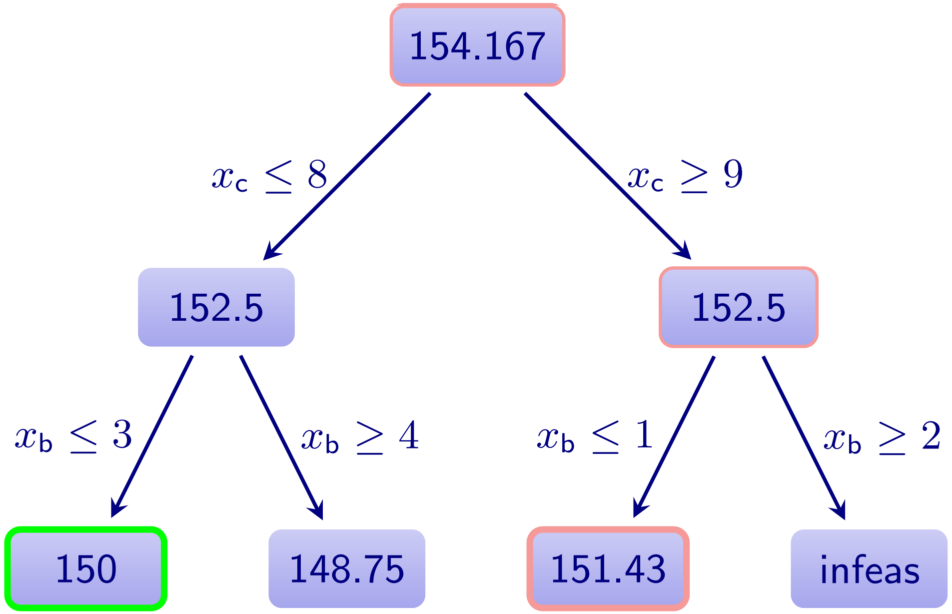
➡ Optimal IP solution of sub-problem 2. has either $x_b \leq 1$ or $x_b \geq 2$



Sub-problem 2.1

$(1, 9.42857)$

objective: 151.43

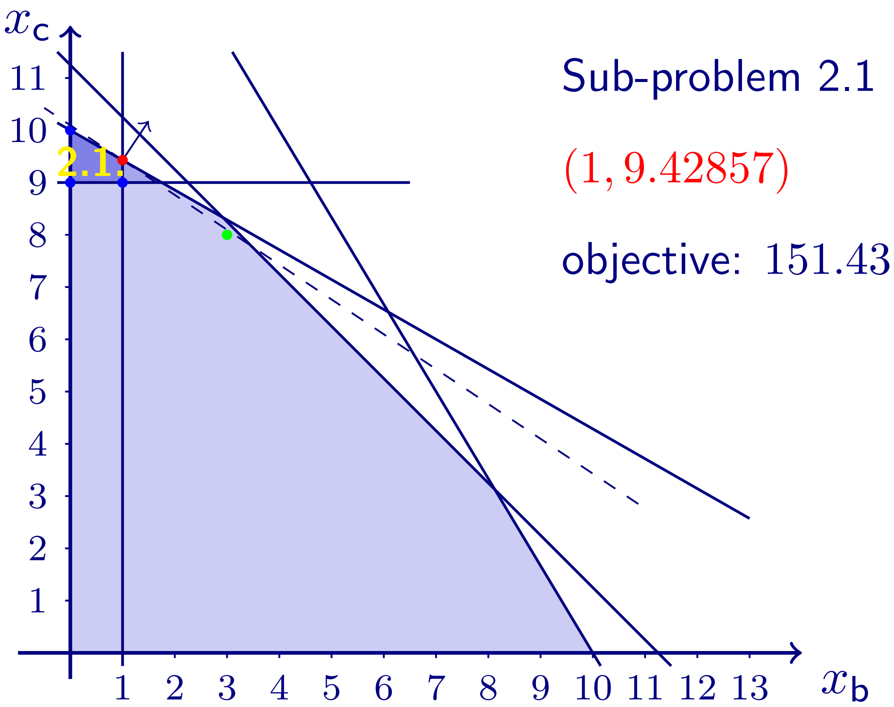


➡ IP optimum cannot be higher than 152.5

➡ IP optimum is at least 150 (objective value of the current incumbent)

➡ All integer points in feasible region 1.2. can be ignored

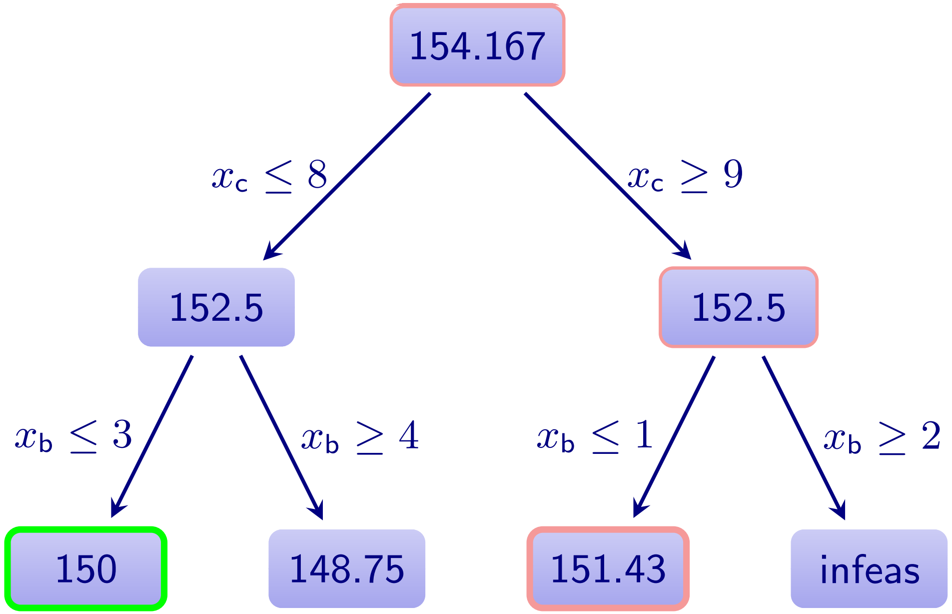
➡ Optimal IP solution of sub-problem 2. has either $x_b \leq 1$ or $x_b \geq 2$



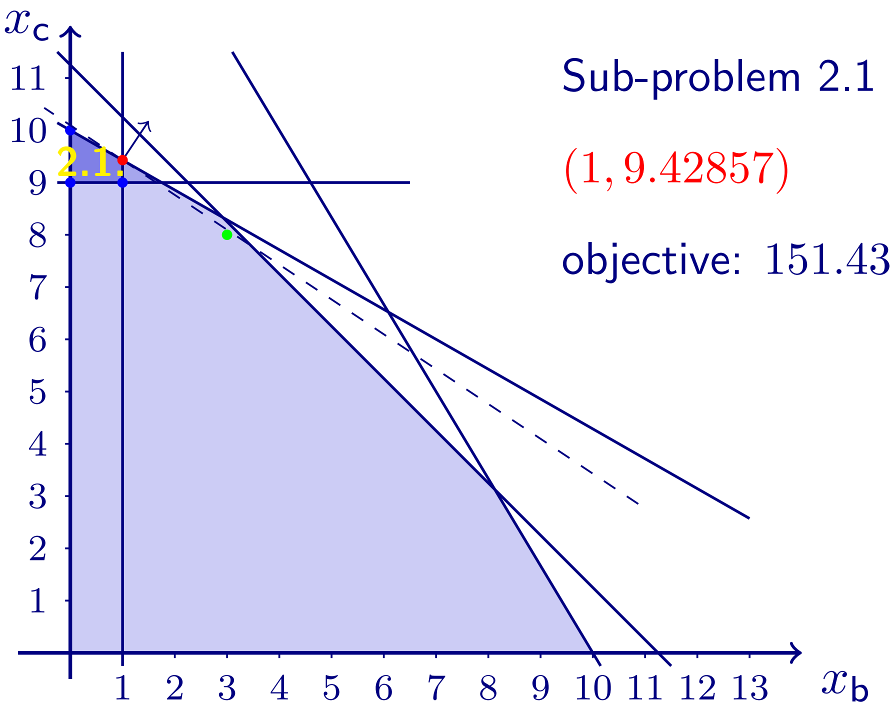
Sub-problem 2.1

$(1, 9.42857)$

objective: 151.43



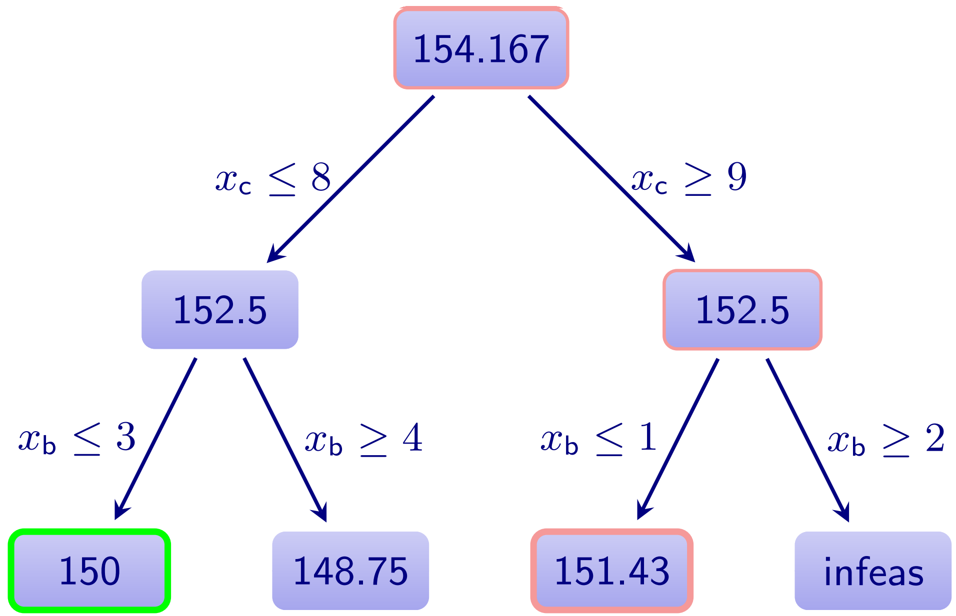
- ➡ IP optimum cannot be higher than 151.43
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored
- ➡ Optimal IP solution of sub-problem 2. has either $x_b \leq 1$ or $x_b \geq 2$



Sub-problem 2.1

$(1, 9.42857)$

objective: 151.43

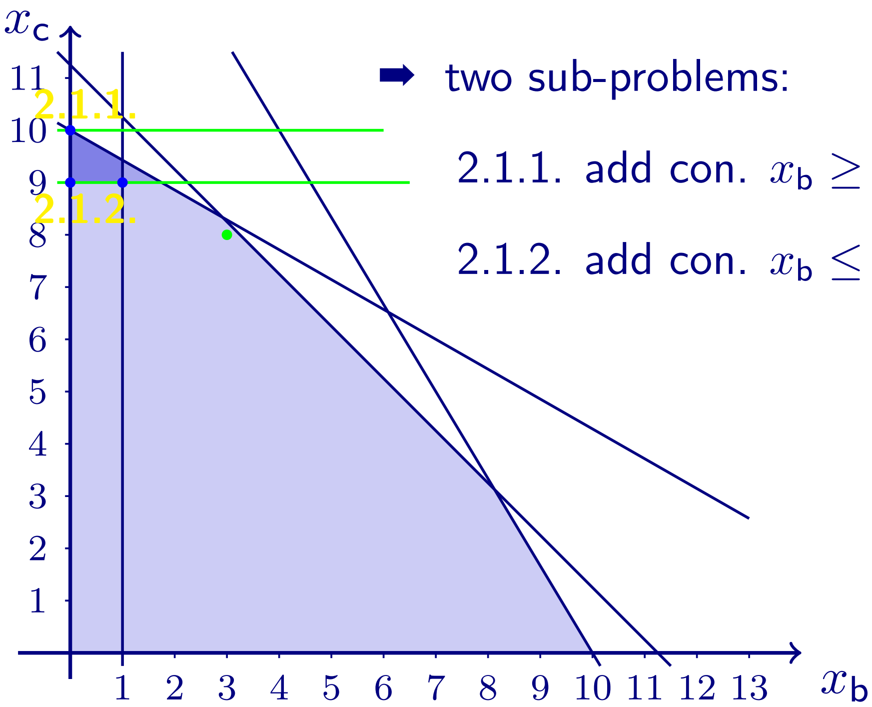


➡ IP optimum cannot be higher than 151.43

➡ IP optimum is at least 150 (objective value of the current incumbent)

➡ All integer points in feasible region 1.2. can be ignored

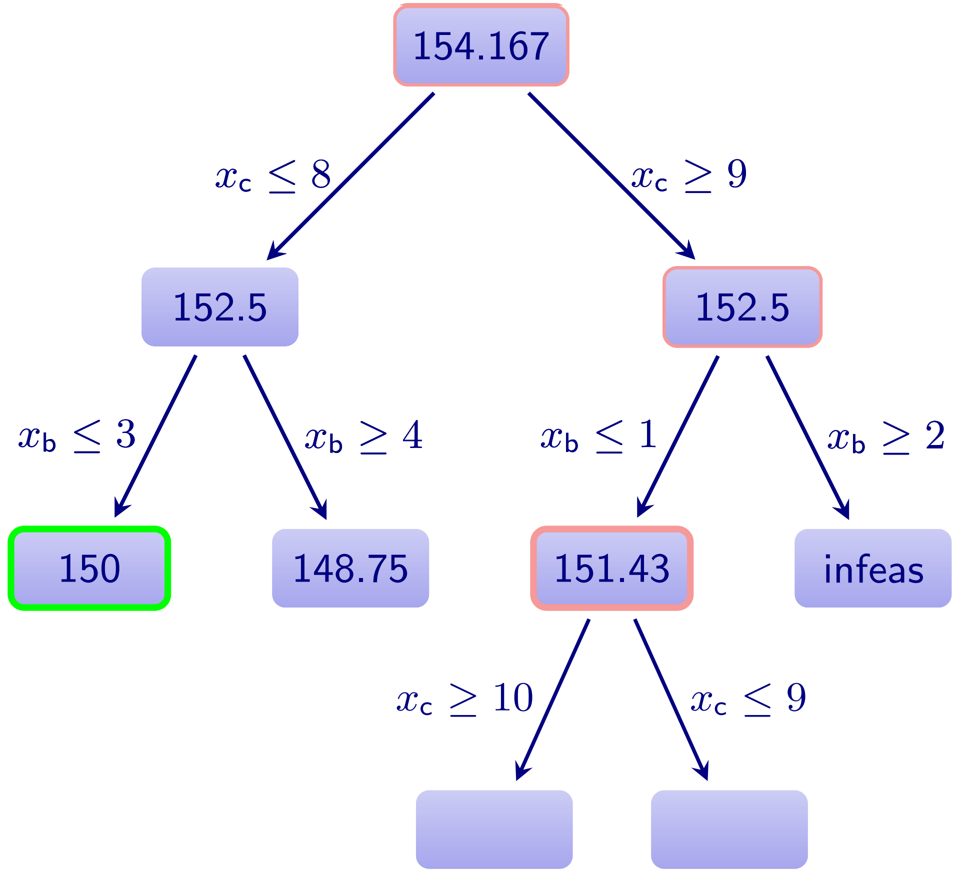
➡ Optimal IP solution of sub-problem 2.1. has either $x_c \geq 10$ or $x_c \leq 9$



➡ two sub-problems:

2.1.1. add con. $x_b \geq 10$

2.1.2. add con. $x_b \leq 9$

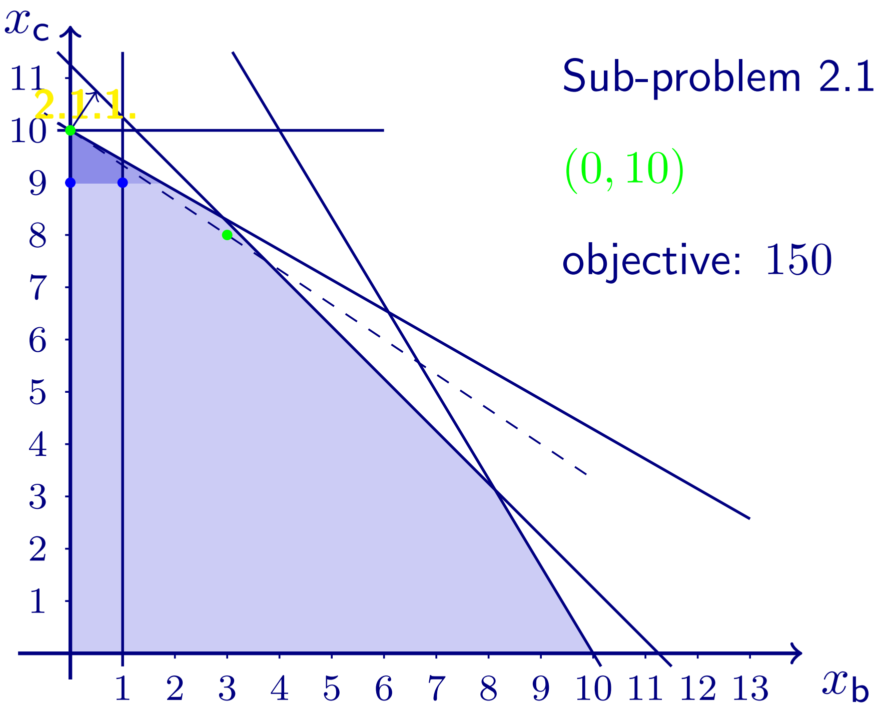


➡ IP optimum cannot be higher than 151.43

➡ IP optimum is at least 150 (objective value of the current incumbent)

➡ All integer points in feasible region 1.2. can be ignored

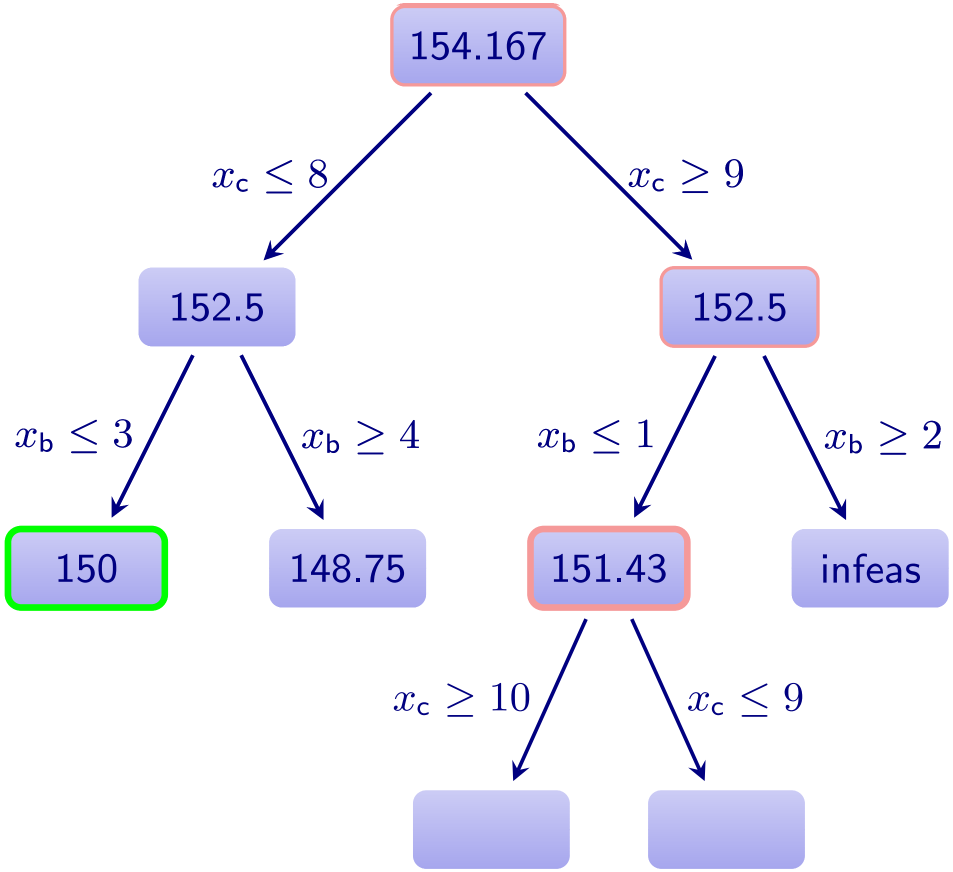
➡ Optimal IP solution of sub-problem 2.1. has either $x_c \geq 10$ or $x_c \leq 9$



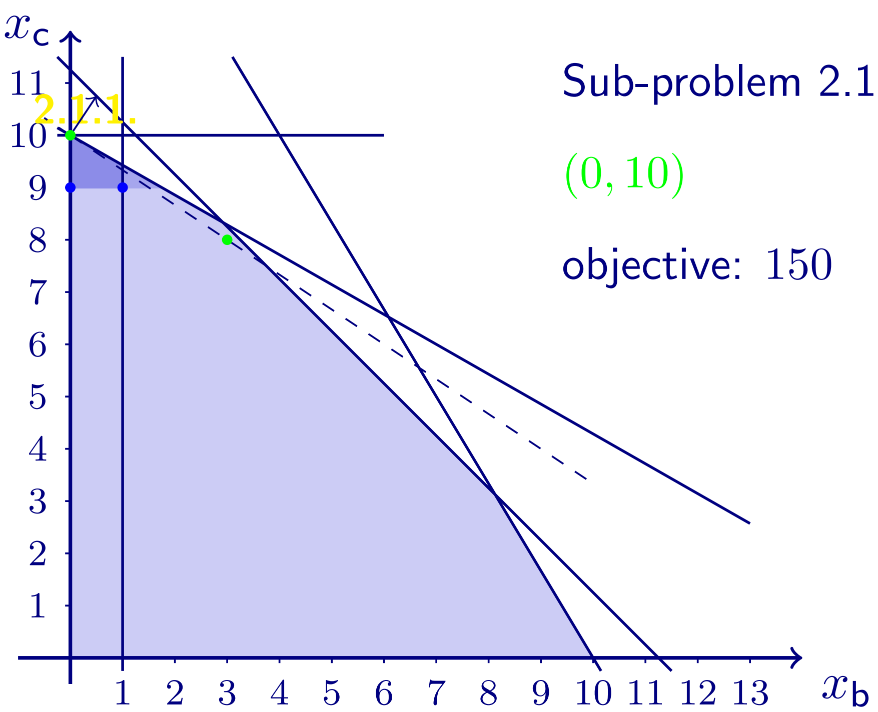
Sub-problem 2.1.1

$(0, 10)$

objective: 150



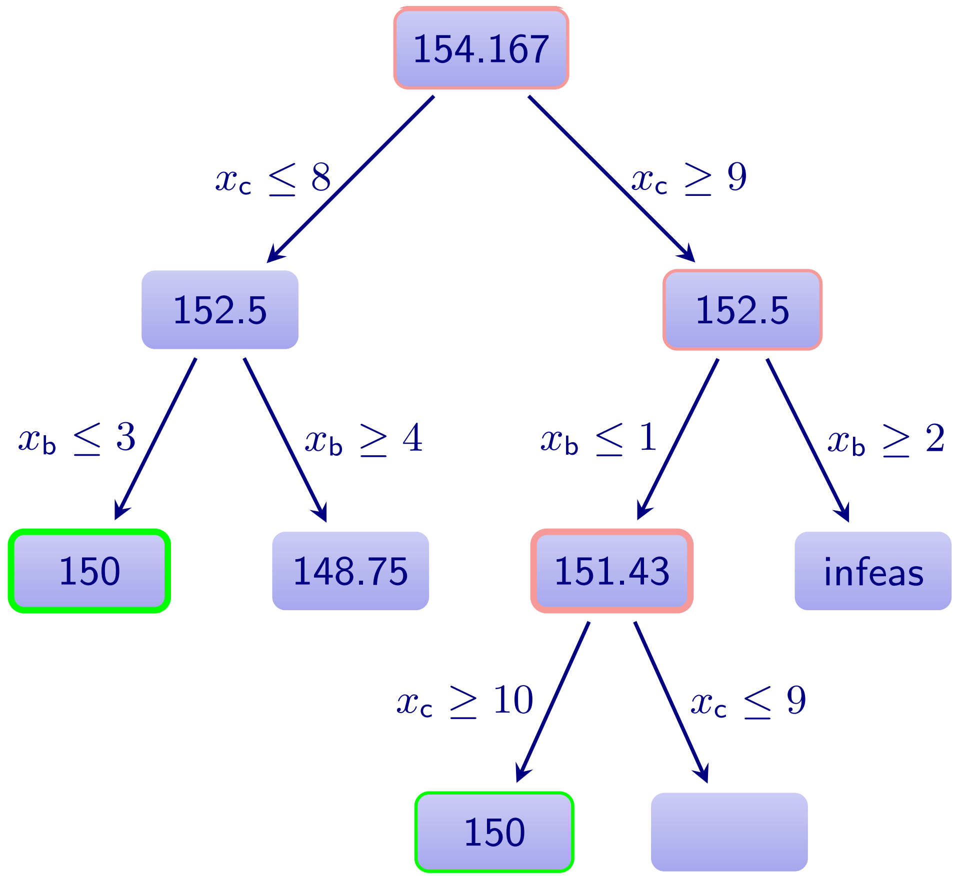
- ➡ IP optimum cannot be higher than 151.43
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored
- ➡ Optimal IP solution of sub-problem 2.1. has either $x_c \geq 10$ or $x_c \leq 9$



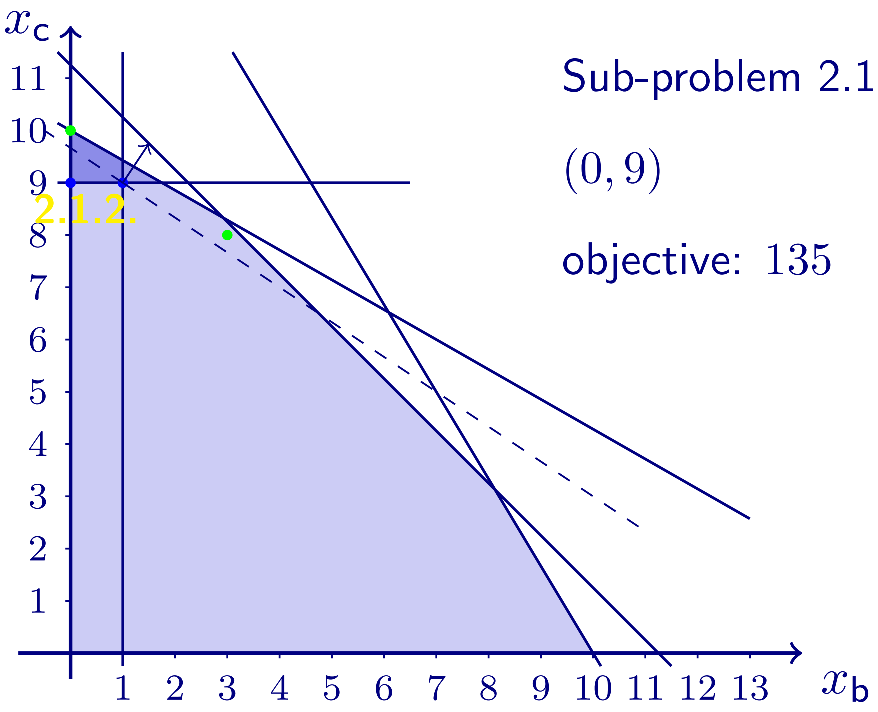
Sub-problem 2.1.1

$(0, 10)$

objective: 150



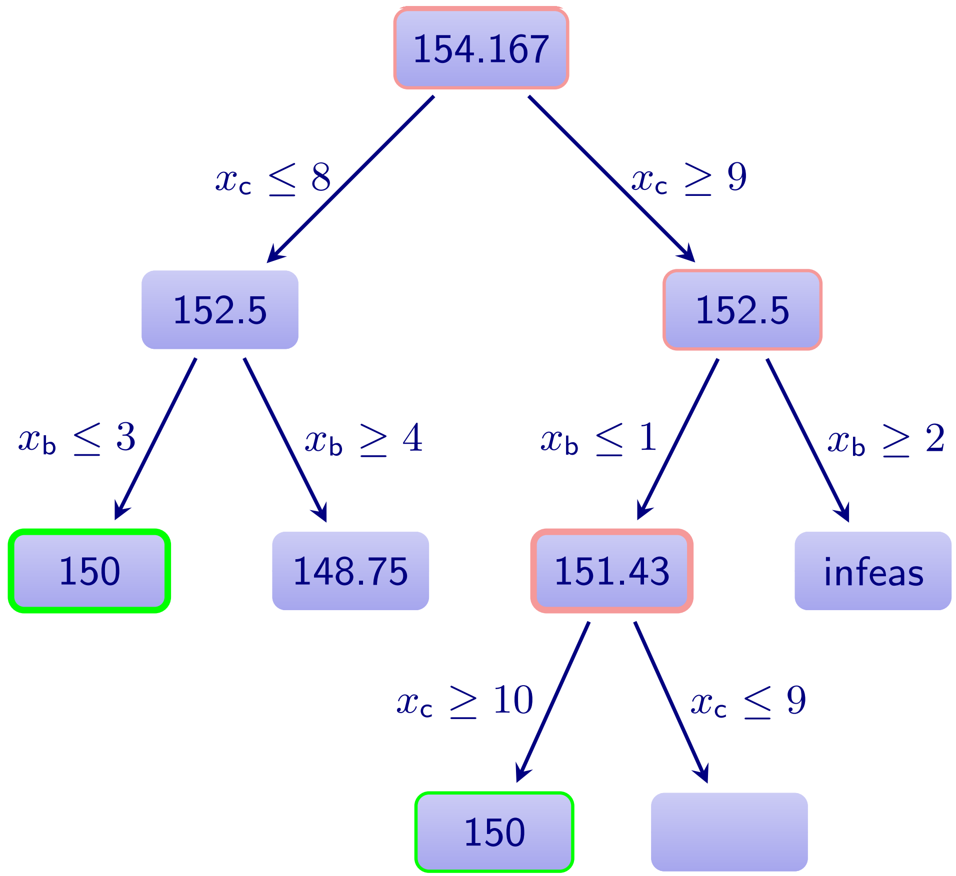
- ➡ IP optimum cannot be higher than 151.43
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored
- ➡ Optimal IP solution of sub-problem 2.1. has either $x_c \geq 10$ or $x_c \leq 9$



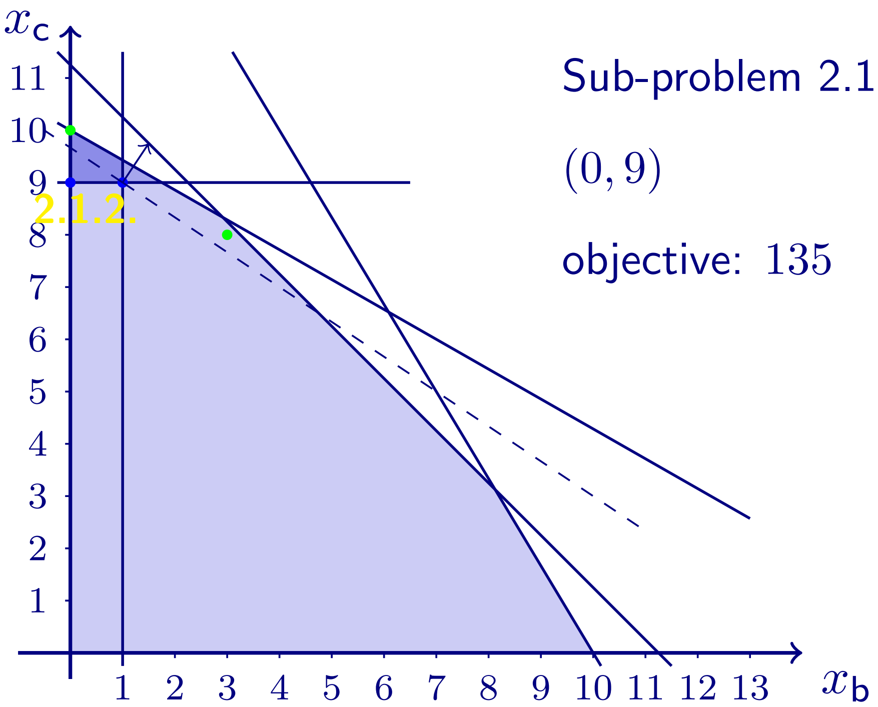
Sub-problem 2.1.2

$(0, 9)$

objective: 135



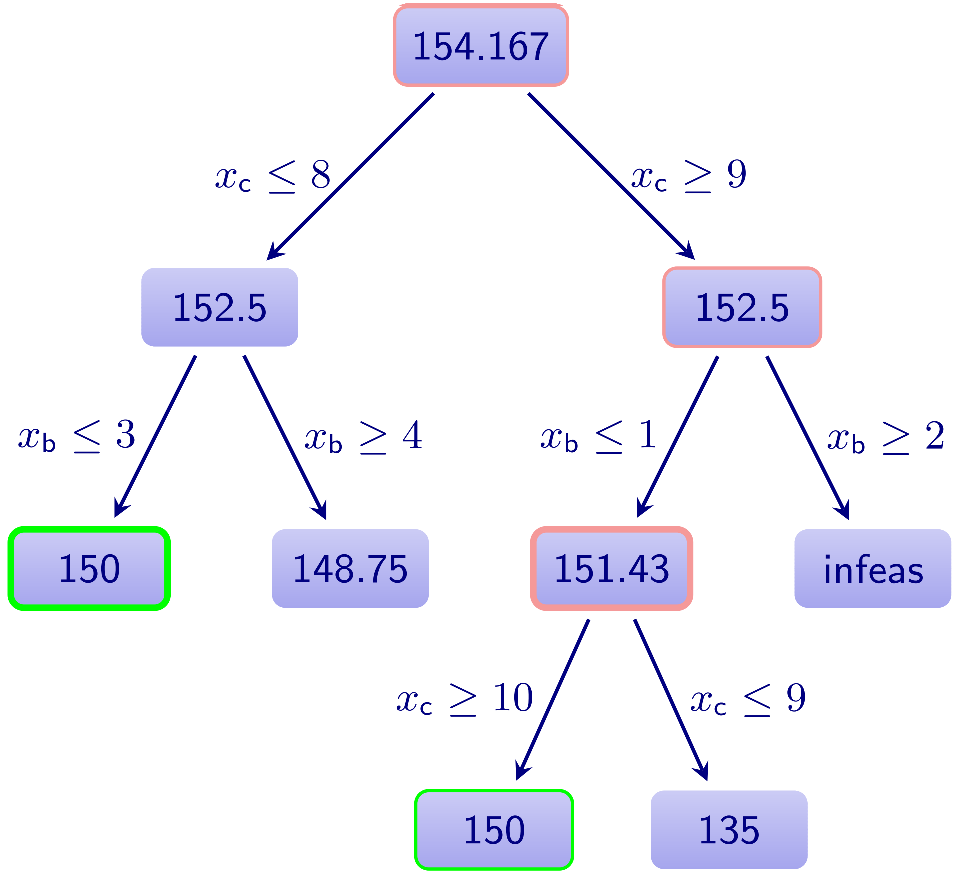
- ➡ IP optimum cannot be higher than 151.43
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored
- ➡ Optimal IP solution of sub-problem 2.1. has either $x_c \geq 10$ or $x_c \leq 9$



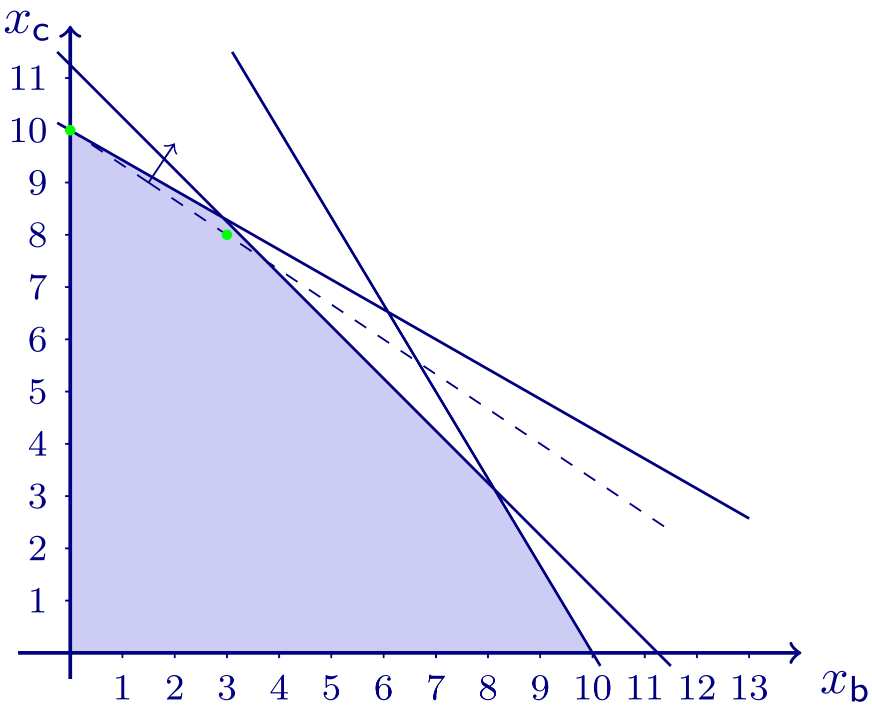
Sub-problem 2.1.2

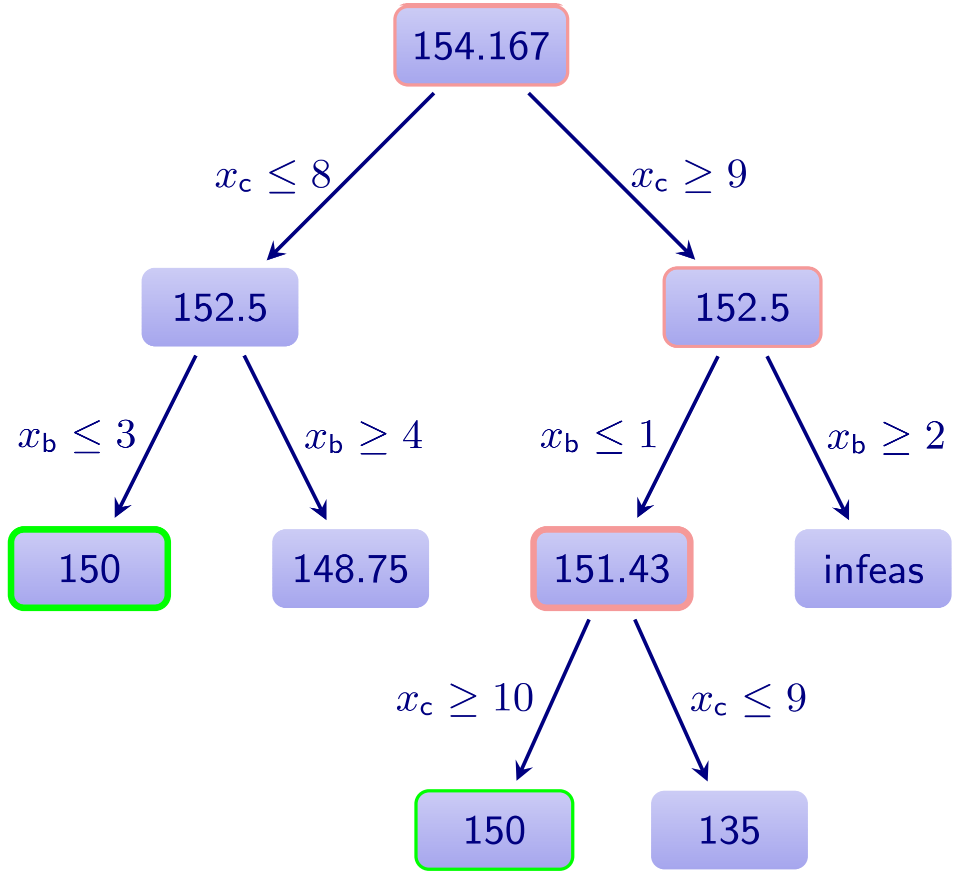
$(0, 9)$

objective: 135



- ➡ IP optimum cannot be higher than 151.43
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All integer points in feasible region 1.2. can be ignored
- ➡ Optimal IP solution of sub-problem 2.1. has either $x_c \geq 10$ or $x_c \leq 9$

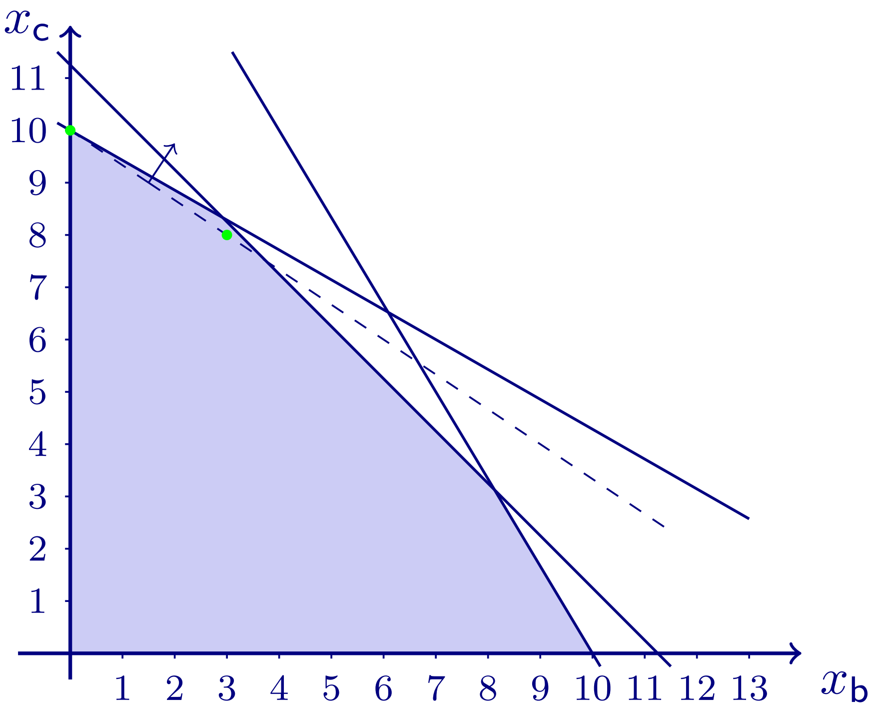


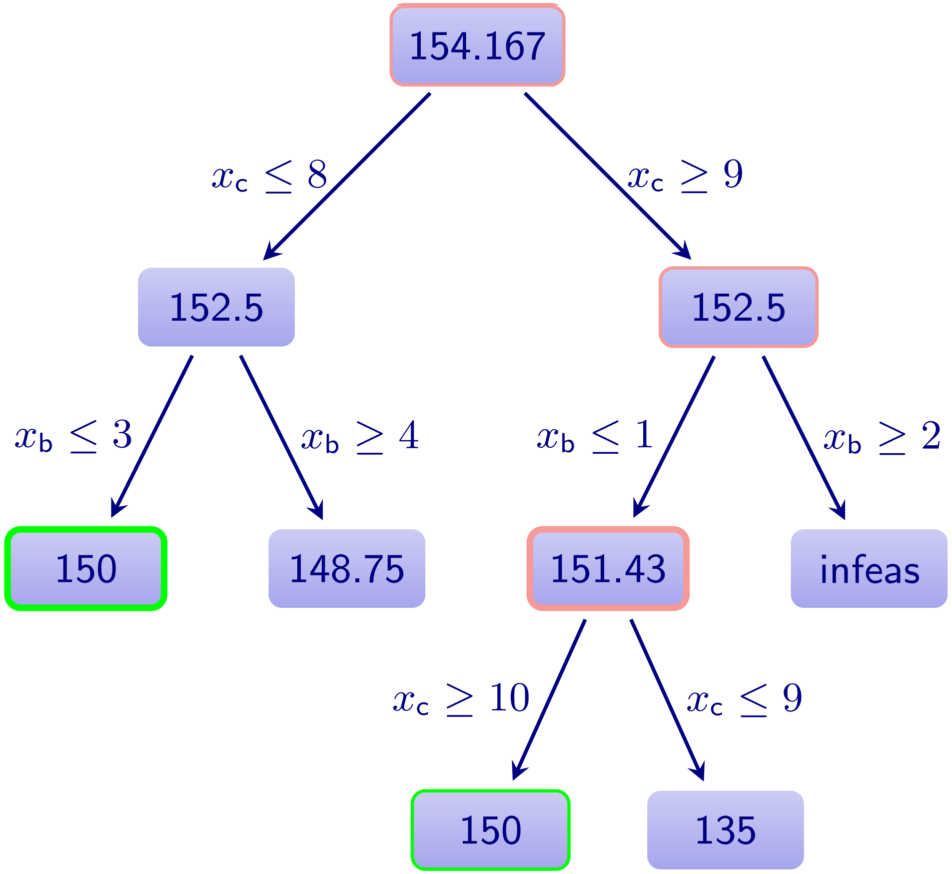


➡ IP optimum cannot be higher than 151.43

➡ IP optimum is at least 150 (objective value of the current incumbent)

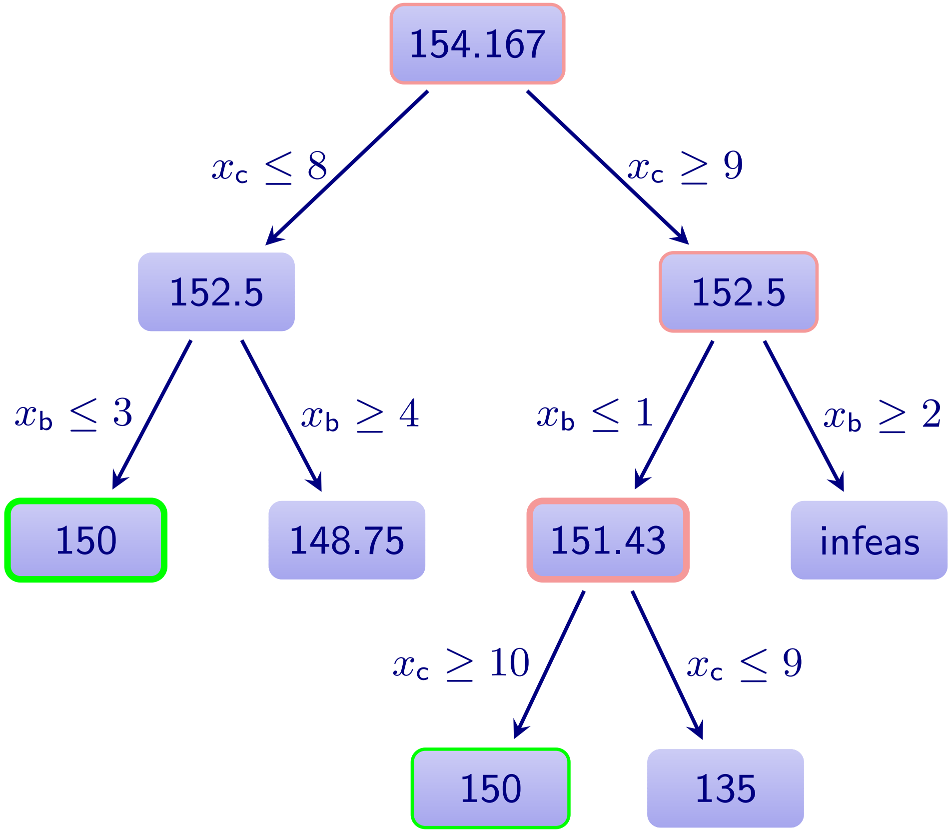
➡ All sub-problems have been solved

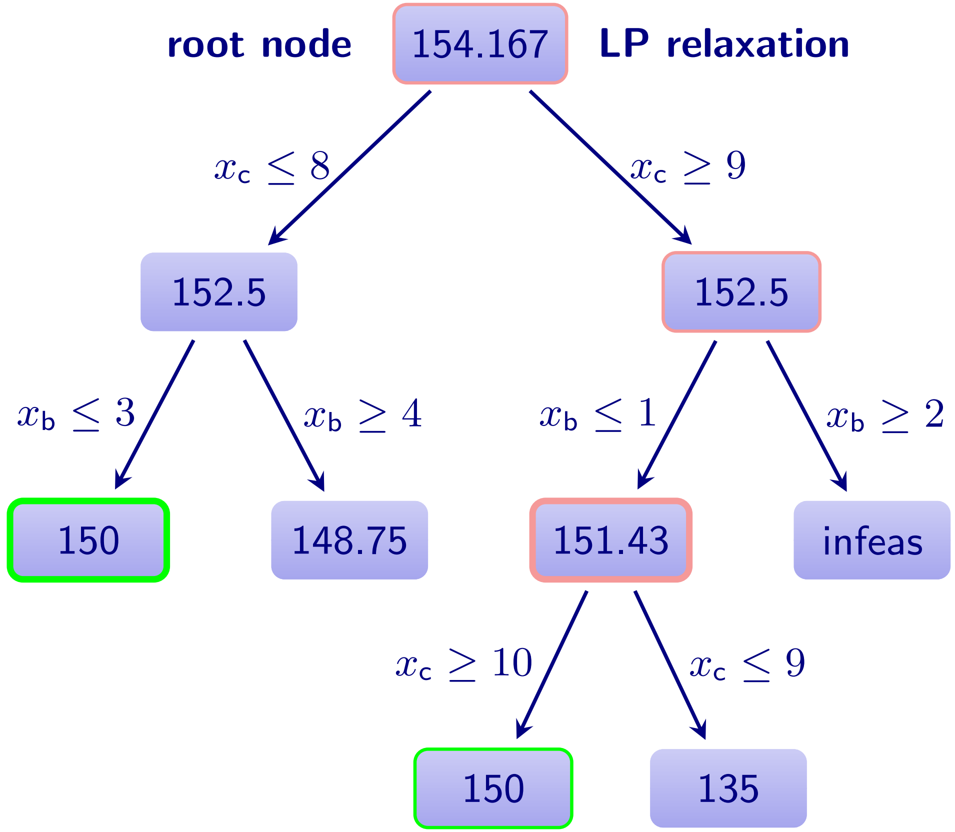


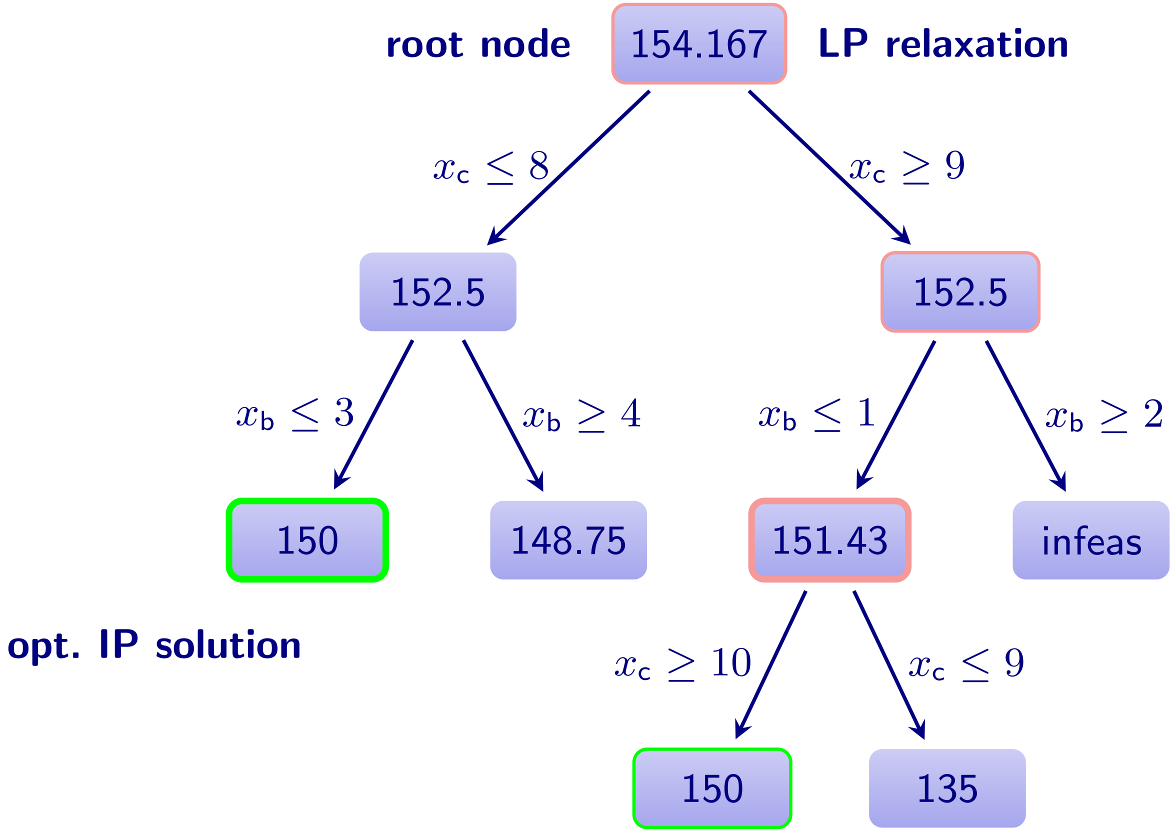


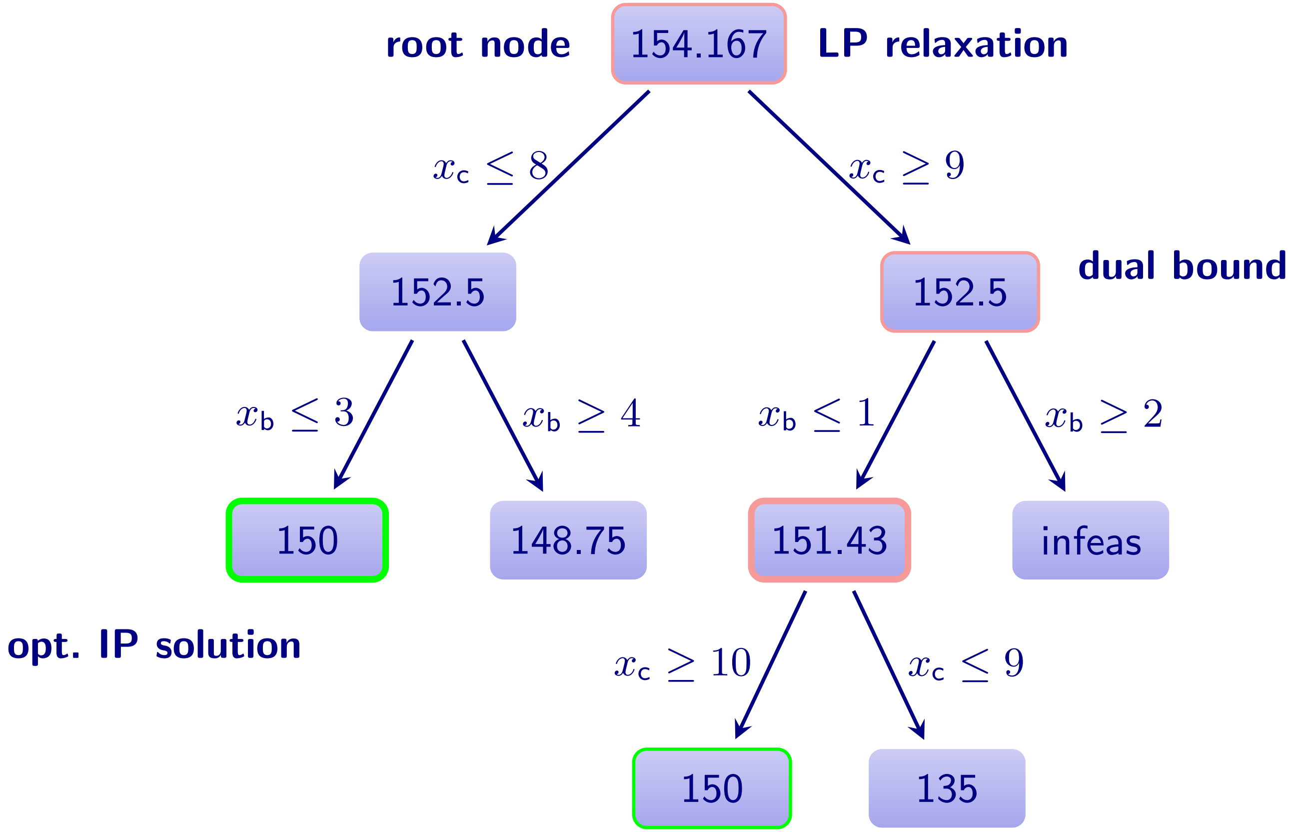
- ➡ IP optimum cannot be higher than 151.43
- ➡ IP optimum is at least 150 (objective value of the current incumbent)
- ➡ All sub-problems have been solved
- ➡ There are two optimal IP solutions: $(3, 8)$ and $(0, 10)$, both with objective value 150

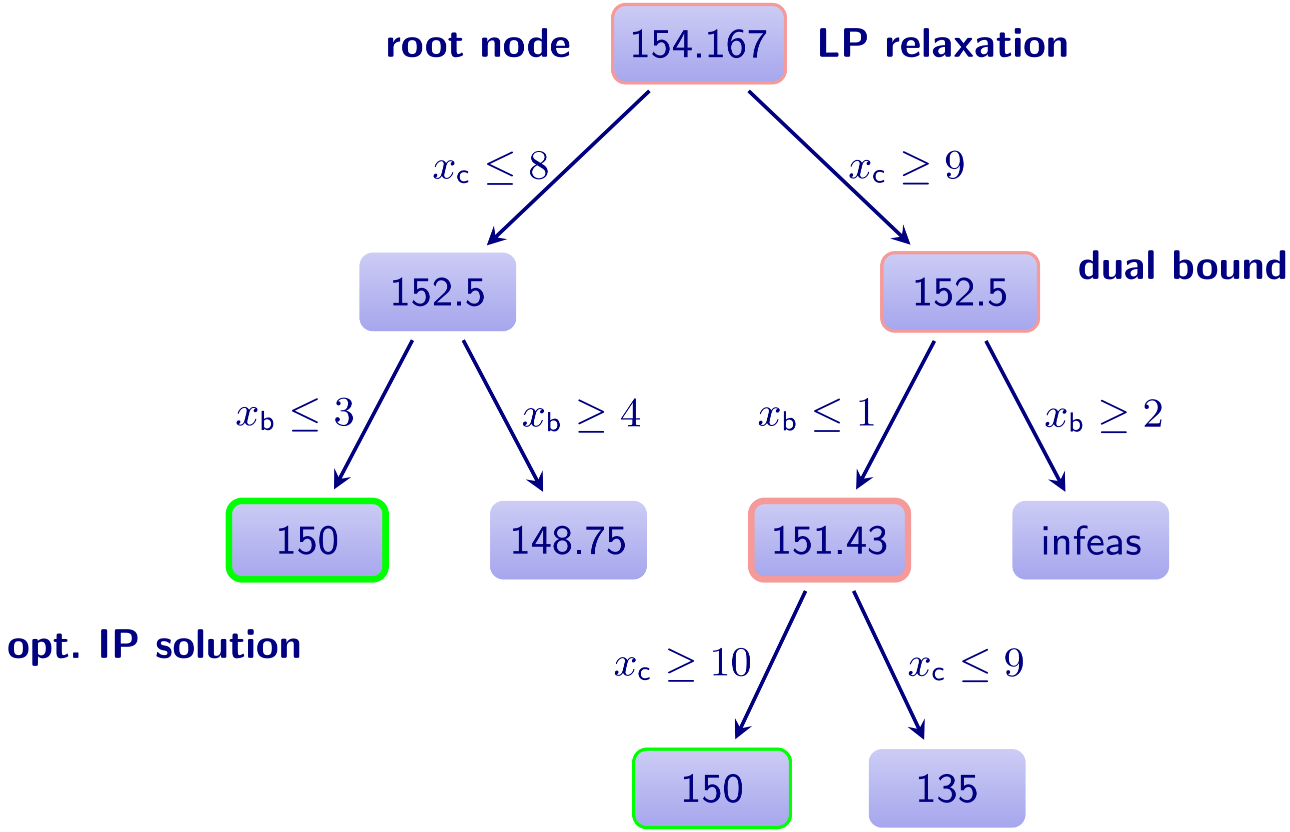
Branch-and-bound tree



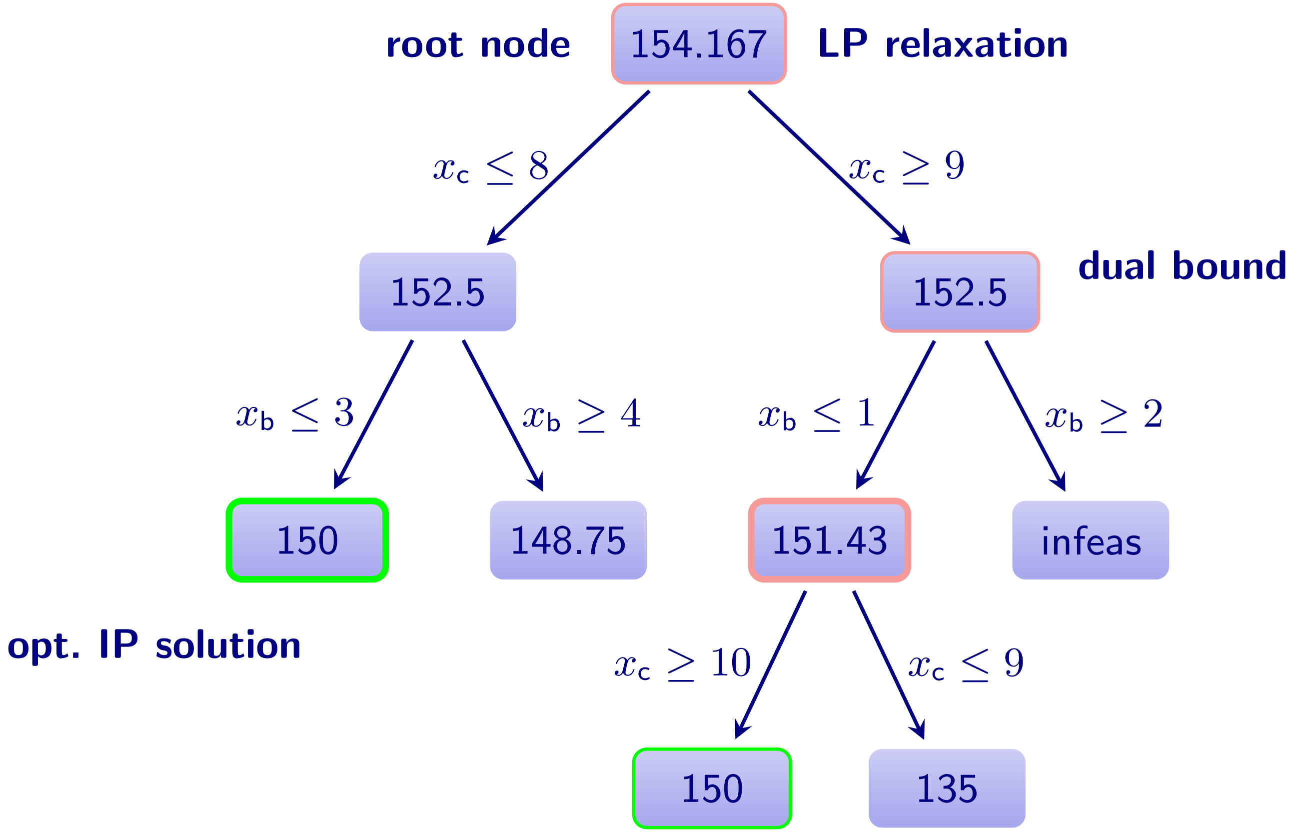

root node
154.167
LP relaxation
$x_c \leq 8$
$x_c \geq 9$
152.5
152.5
$x_b \leq 3$
$x_b \geq 4$
$x_b \leq 1$
$x_b \geq 2$
150
148.75
151.43
infeas
$x_c \geq 10$
$x_c \leq 9$
150
135


root node
154.167
LP relaxation
$x_c \leq 8$
$x_c \geq 9$
152.5
152.5
$x_b \leq 3$
$x_b \geq 4$
$x_b \leq 1$
$x_b \geq 2$
150
148.75
151.43
infeas
opt. IP solution
$x_c \geq 10$
$x_c \leq 9$
150
135


root node
154.167
LP relaxation
$x_c \leq 8$
$x_c \geq 9$
152.5
152.5
dual bound
$x_b \leq 3$
$x_b \geq 4$
$x_b \leq 1$
$x_b \geq 2$
150
148.75
151.43
infeas
opt. IP solution
$x_c \geq 10$
$x_c \leq 9$
150
135



- Solving the LP in every step might result in:
  - a new integer solution with greater objective than the current optimum
    ➡ new incumbent, primal optimum increases

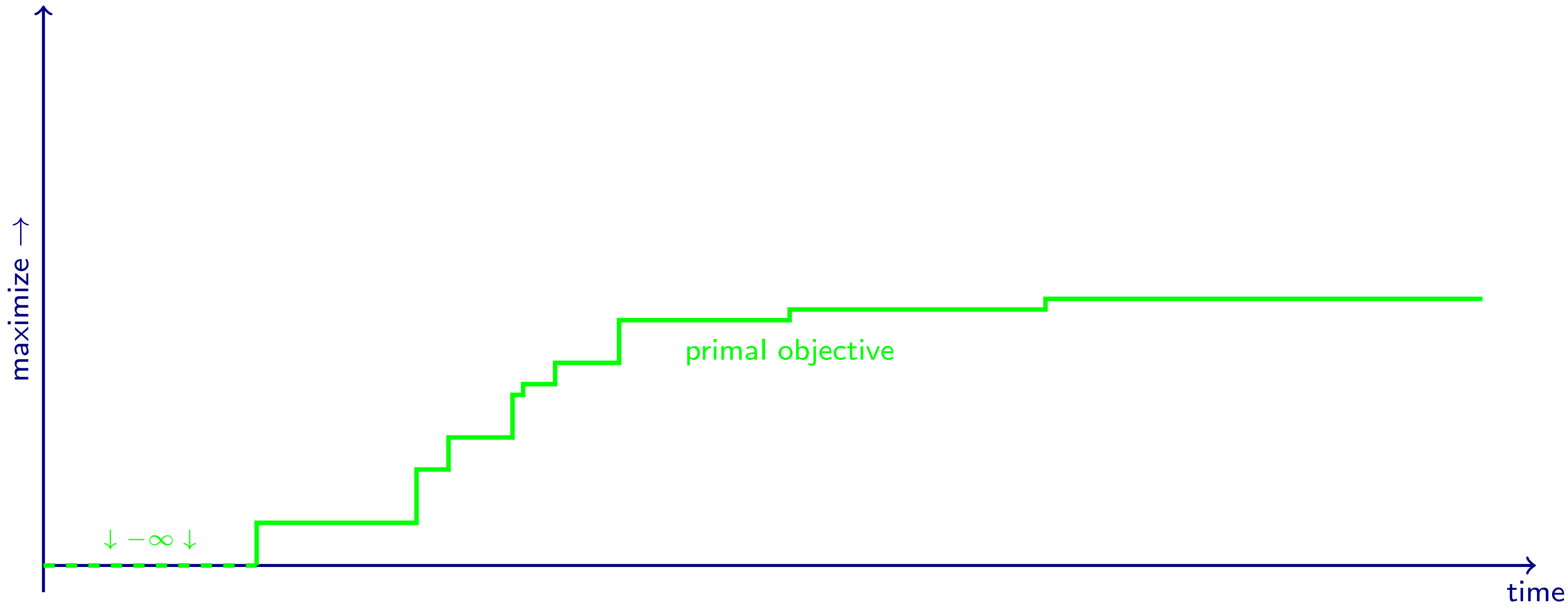


▷ Solving the LP in every step might result in:

- a new integer solution with greater objective than the current optimum
  ➡ new incumbent, primal optimum increases
- a fractional solution with lower objective than the current dual bound
  ➡ (possibly) new dual bound, upper bound decreases

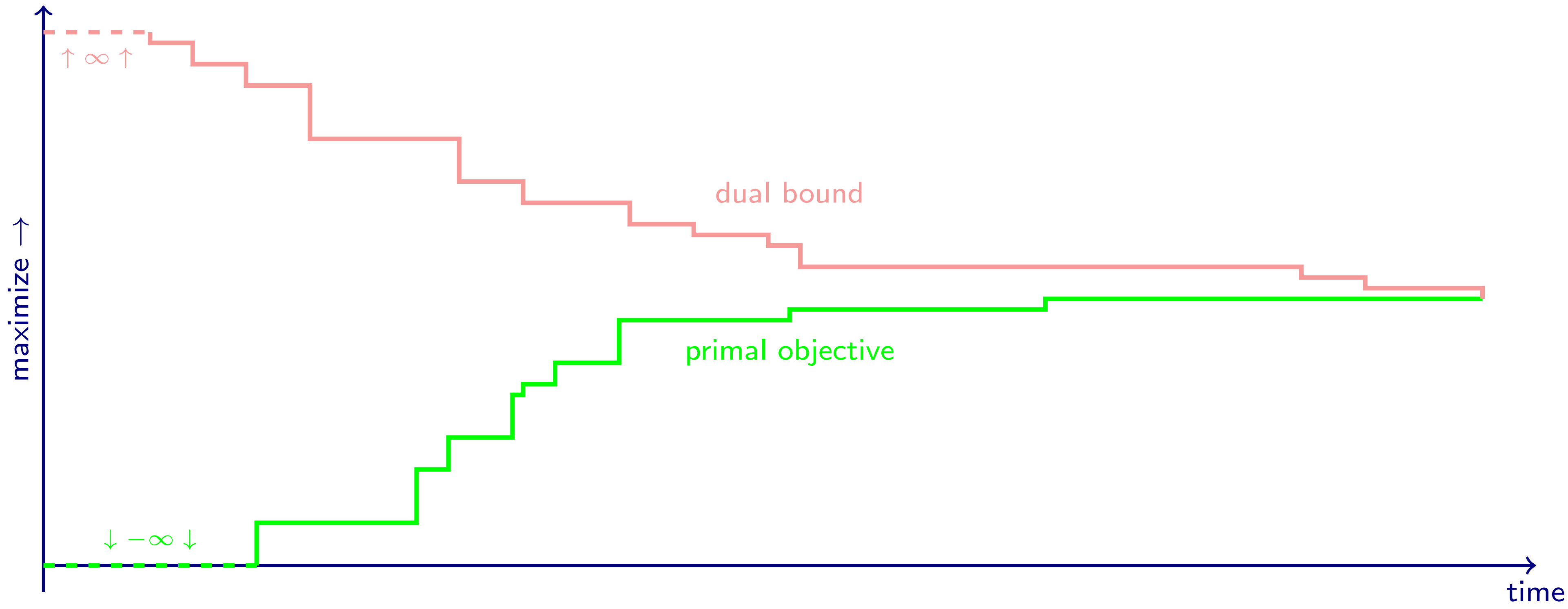
▷ The (absolute) gap during branch and bound: gap := bestdual − bestprimal

▷ The (absolute) gap during branch and bound: gap := bestdual − bestprimal



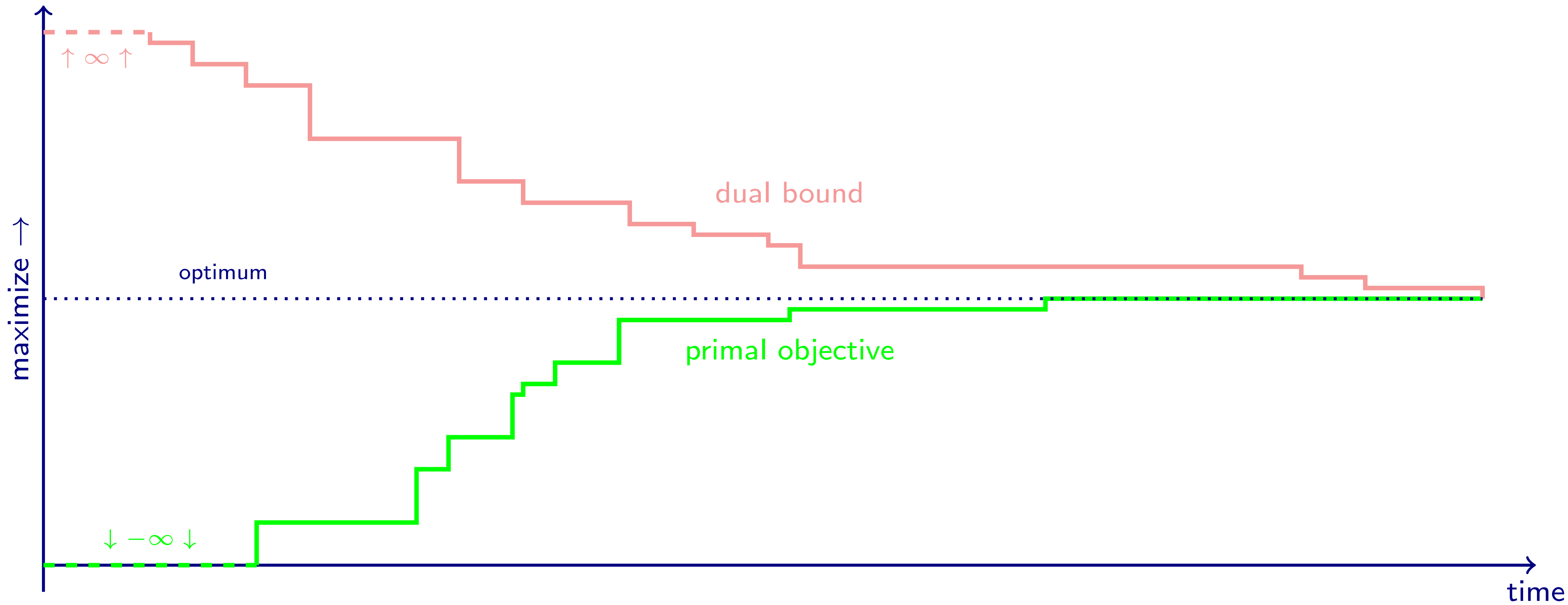
Gap

- The (absolute) gap during branch and bound: gap := bestdual − bestprimal

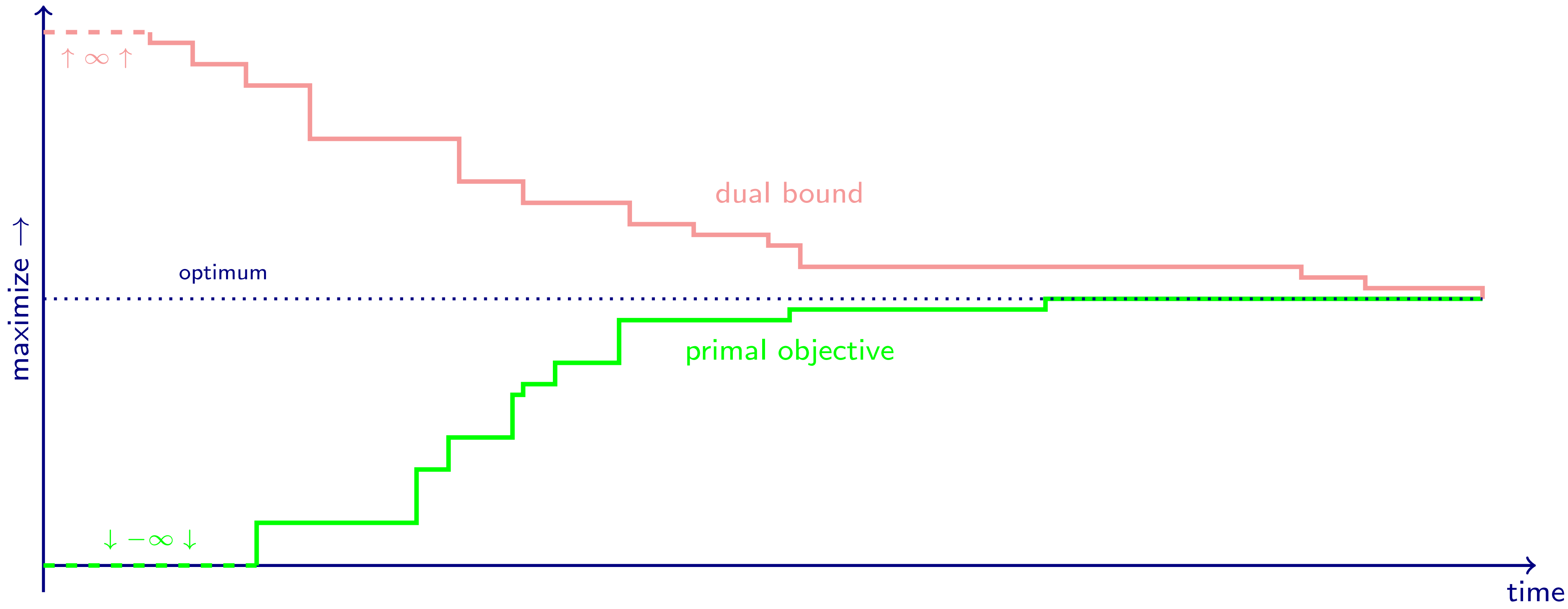


Gap

▷ The (absolute) gap during branch and bound: gap := bestdual − bestprimal

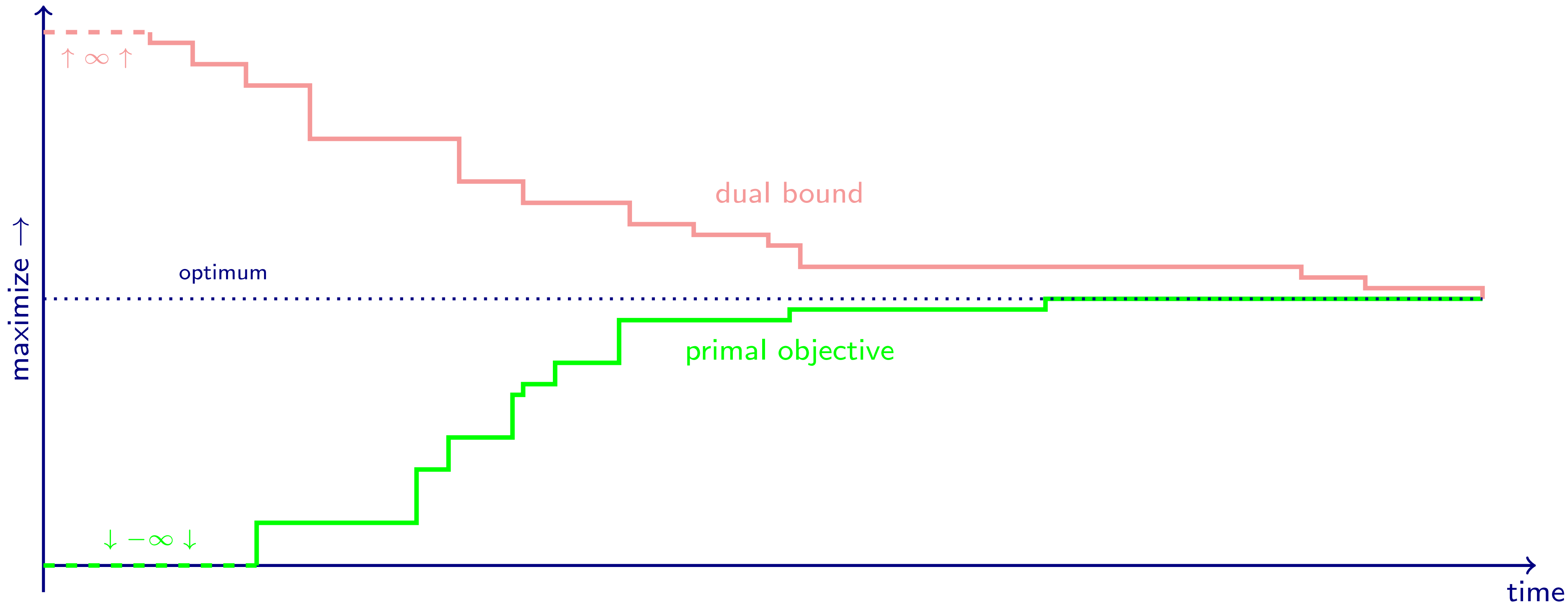


- The (absolute) gap during branch and bound: gap := bestdual − bestprimal



- Stop traversing the tree, if the gap is 0, i.e. the value of the best incumbent and the dual bound coincide

- The (absolute) gap during branch and bound: gap := bestdual − bestprimal



- Stop traversing the tree, if the gap is 0, i.e. the value of the best incumbent and the dual bound coincide
- In practise (and for large-scale MIPs): stop traversing the tree already if the relative gap $\frac{\text{gap}}{|\,\text{bestdual}\,|}$ is below a certain target (e.g. 5%, 1%, 0.5%, ...)

- Even if an optimal solution is not found, branch and bound delivers an optimality certificate, that gives valuable information on the solution quality!

▷ Even if an optimal solution is not found, branch and bound delivers an optimality certificate, that gives valuable information on the solution quality!

	modelling flexibility	solvability	optimality certificate
LPs	−	++	+++
MIPs	+	o	++
heuristics	++	+	−−
approximation algorithms	++	+	+
non-linear (convex) models	+	o	+
non-linear (general) models	++	−	o

- Even if an optimal solution is not found, branch and bound delivers an optimality certificate, that gives valuable information on the solution quality!

	modelling flexibility	solvability	optimality certificate
LPs	–	++	+++
MIPs	+	o	++
heuristics	++	+	– –
approximation algorithms	++	+	+
non-linear (convex) models	+	o	+
non-linear (general) models	++	–	o

- Tuning branch & bound:

▷ Even if an optimal solution is not found, branch and bound delivers an optimality certificate, that gives valuable information on the solution quality!

	modelling flexibility	solvability	optimality certificate
LPs	–	++	+++
MIPs	+	o	++
heuristics	++	+	– –
approximation algorithms	++	+	+
non-linear (convex) models	+	o	+
non-linear (general) models	++	–	o

▷ Tuning branch & bound:

- Order of processing the nodes (sub-problems)

▷ Even if an optimal solution is not found, branch and bound delivers an optimality certificate, that gives valuable information on the solution quality!

	modelling flexibility	solvability	optimality certificate
LPs	–	++	+++
MIPs	+	o	++
heuristics	++	+	– –
approximation algorithms	++	+	+
non-linear (convex) models	+	o	+
non-linear (general) models	++	–	o

▷ Tuning branch & bound:

- Order of processing the nodes (sub-problems)
- Defining sub-problems (branching)

▷ Even if an optimal solution is not found, branch and bound delivers an optimality certificate, that gives valuable information on the solution quality!

	modelling flexibility	solvability	optimality certificate
LPs	–	++	+++
MIPs	+	o	++
heuristics	++	+	– –
approximation algorithms	++	+	+
non-linear (convex) models	+	o	+
non-linear (general) models	++	–	o

▷ Tuning branch & bound:

- Order of processing the nodes (sub-problems)
- Defining sub-problems (branching)
- Cutting planes ➡ branch & cut

- Models, Data and Algorithms
- Linear Optimization
- Mathematical Background: Polyhedra, Simplex-Algorithm
- Sensitivity Analysis; (Mixed) Integer Programming
- MIP Modelling
- MIP Modelling: More Examples; Branch & Bound
- Cutting Planes; Combinatorial Optimization: Examples, Graphs, Algorithms
- Complexity Theory
- Nonlinear Optimization
- Scheduling
- Lot Sizing
- Multicriteria Optimization
- Oral exam