Section #2:
Linear and Integer Programming

Prof. Dr. Sven Seuken
8.3.2012

(with most slides borrowed from David Parkes)

Housekeeping

- Game Theory homework submitted?
- HW-00 and HW-01 returned
- Feedback on Comprehension Questions later today or tomorrow
- BitTorrent assignment → online this afternoon
- Next chapter → online this afternoon
- Questions? Concerns?

Outline

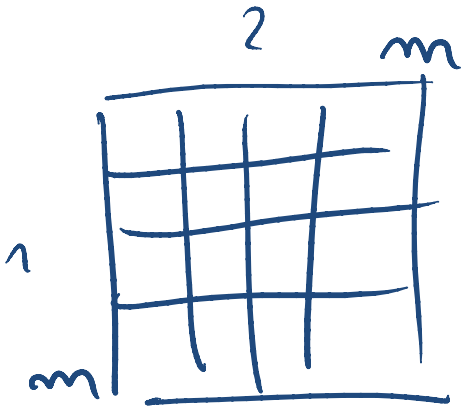
1. O()-Notation
2. NP vs. P
3. Linear Programming
4. Integer Programming

Big O()-Notation

- Analysis of algorithms
- Asymptotic running time
- Input Size…?

$O(x^2)$

$x = \#$ of bytes



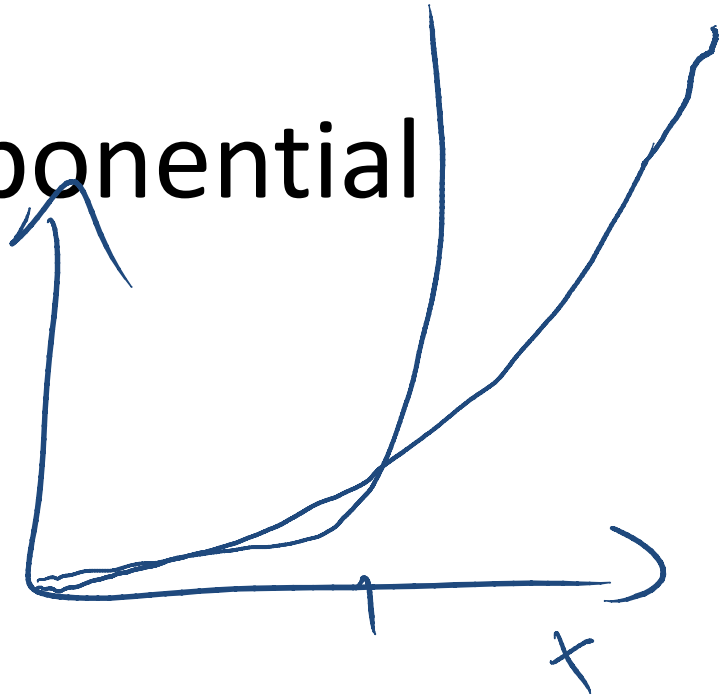
$m^2 \cdot 2 \cdot 8$

$x =$ input size $= O(x \cdot m^2)$

running time $\rightarrow O(m^4)$

$\rightarrow O(x^2)$

Polynomial vs. Exponential



$O(x^2)$

$O(x^3)$

$\vdots$

$O(x^k)$ = polynomial

$\underline{\underline{100{,}000}} \cdot x^k + \underline{\underline{1{,}000{,}000}}$

$O(2^x)$, $O(3^x)$

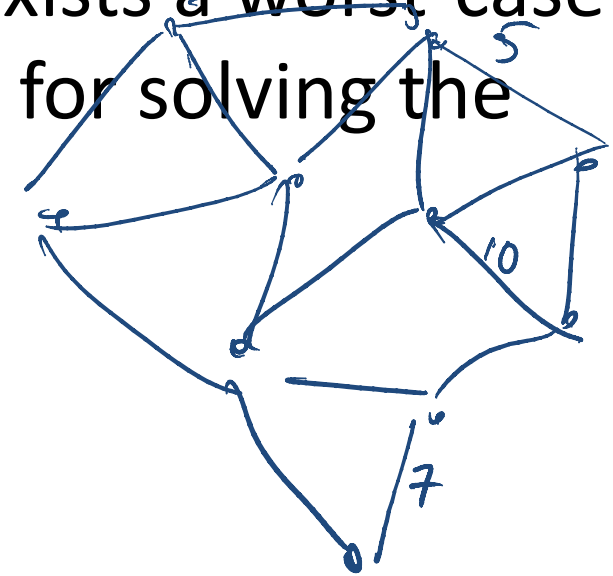
The Complexity Class “P”

- Decision Problems
- A problem is in P, if there exists a worst-case polynomial time algorithm for solving the problem.
- Efficient

Yes, No

$L \leq 100$

$O(x^k)$



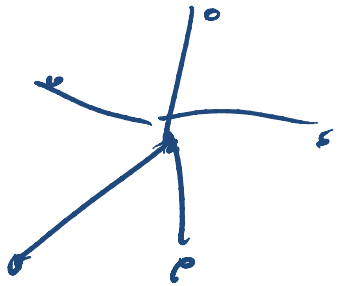
The Complexity Class “NP”

- Decision Problems! (YES vs NO)
- A problem is in NP, if, given an answer (or a proof), there exists a worst-case polynomial time algorithm that can verify that the answer is YES.

co-NP

1, 2, 3

3 | 7 | 2 | 1



NP-hard vs. NP-complete

Problem $\in NP$ ✓

Problem NP-hard.

Reduction $P_1 \leq_p P_2$

NP-compl.: $P_1 \in NP \wedge P_1$ is NP-hard

$\forall P_i, P_j \in NP\text{-}c: \quad P_i \leq_p P_j$

P vs. NP

- **Important Conjecture: P ≠NP**

- **Why do we believe this?**

- Consequence:
  - Fastest algorithms for problems that are NP-hard probably require exponential time in the worst case.

→ Inefficient algorithms (for medium-sized problems)

Mathematical Optimization

- What is “optimization”?
  - problem of making decisions to maximize or minimize an objective in the presence of complicating constraints

- Examples: scheduling aircraft, designing a jet engine, sourcing goods, predicting who will win a football game.

Example: Going Shopping

- I can buy pizza x_1, and chicken x_2
- Pizza costs $5 per unit, and gives me 3 carbohydrates and 2 proteins per unit
- Chicken costs $10 per unit, and gives me 5 proteins and 1 carbohydrate per unit
- Task:
  - Maximize the carbohydrates
  - Do not spend more than $30
  - Need at least 10 protein units
- Variables?
- Objective?
- Constraints?

$$\max_{x_1, x_2} \; 3x_1 + 1x_2$$

$$\text{s.t.} \quad 5x_1 + 10x_2 \leq 30$$

$$2x_1 + 5x_2 \geq 10$$

$$x_1, x_2 \geq 0$$

Example: Marketing campaign

- Comedy– 7 million high-income women, 2 million high-income men. Cost $50,000 x_1
- Football – 2 million high-income women and 12 million high-income men. Cost $100,000 x_2
- Goal: reach at least 28 million high-income women and 24 million high-income men at MINIMAL cost

$$\max \quad x_1 \cdot x_2$$

$$\max \quad x_1^2$$

$$\boxed{x_1, x_2 \in \mathbb{Z}_+}$$

$$\min_{x_1, x_2} \quad 50{,}000\, x_1 + 100{,}000\, x_2$$

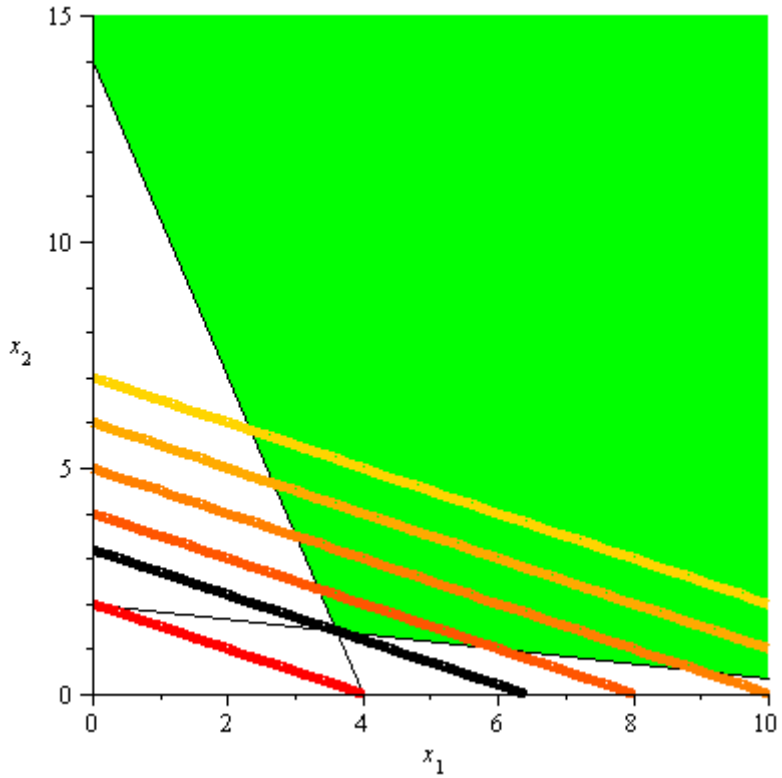
$$\text{s.t.} \quad 7x_1 + 2x_2 \geq 28$$

$$2x_1 + 12x_2 \geq 24$$

$$x_1, x_2 \geq 0$$

graphical version of problem
(solution is x_1=3.6, x_2=1.4, value 320)

$$\begin{aligned} \min \quad & z = 50x_1 + 100x_2 \\ \text{s.t.} \quad & 7x_1 + 2x_2 \geq 28 \\ & 2x_1 + 12x_2 \geq 24 \\ & x_1, x_2 \geq 0 \end{aligned}$$



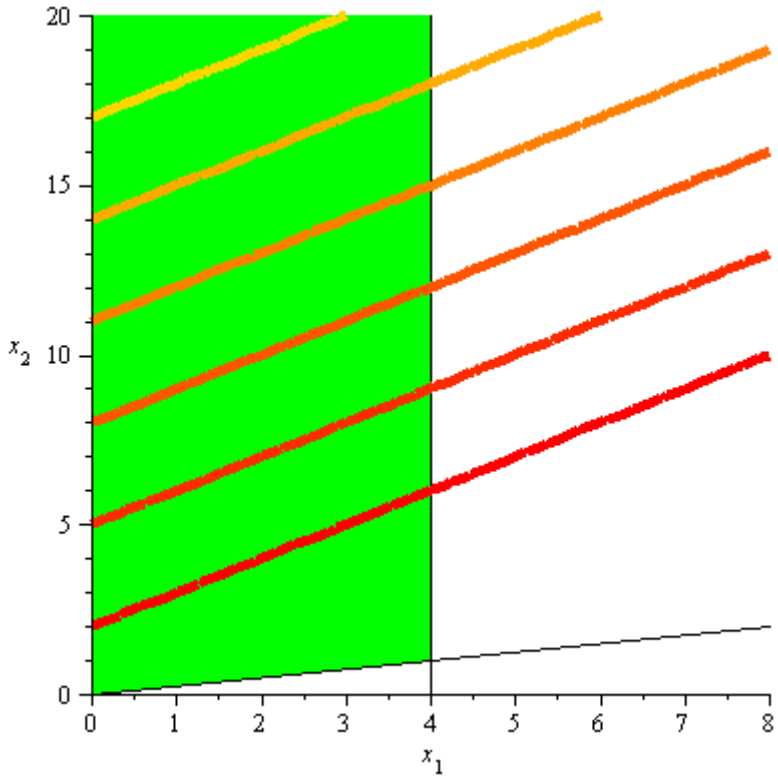
Example: Unbounded objective

$$\max \; z = -x_1 + x_2$$

$$\text{s.t.} \quad -x_1 + 4x_2 \geq 0$$

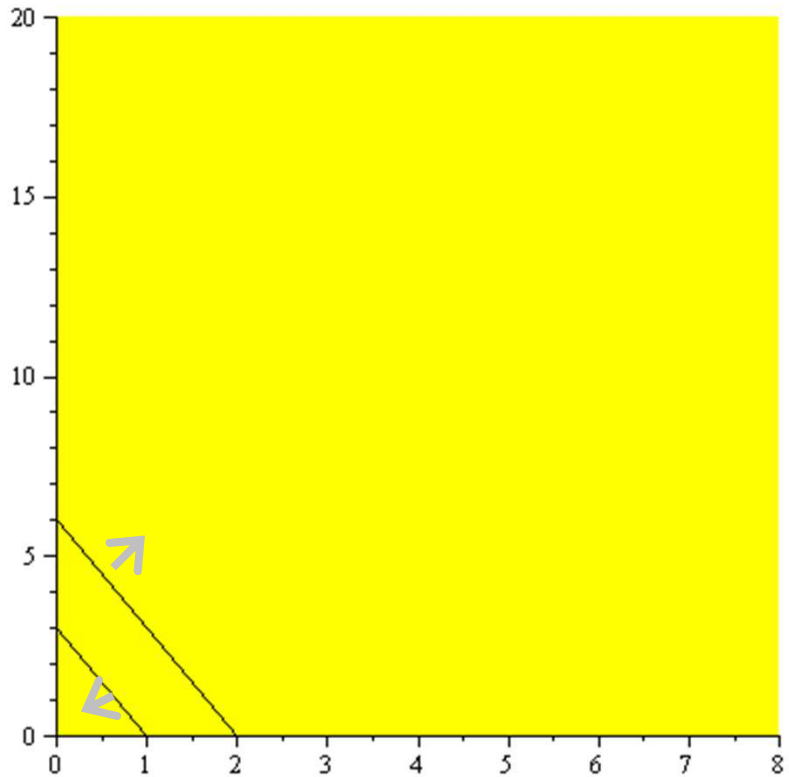
$$x_1 \leq 4$$

$$x_1, x_2 \geq 0$$



Example: Infeasible problem

$$\begin{aligned} \max \quad & z = x_1 + x_2 \\ \text{s.t.} \quad & 3x_1 + x_2 \geq 6 \\ & 3x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Post Office Problem

Union rules state that each full-time employee must work 5 consecutive days and then receive 2 days off.

Formulate an LP to minimize the number of full-time employees who must be hired.

Day	Number of full-time employees required
1=Monday	17
2=Tuesday	13
3=Wednesday	15
4=Thursday	19
5=Friday	14
6=Saturday	16
7=Sunday	11

$$\min z = \sum_i x_i$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_i x_i - x_2 - x_3 \geq 17 \\
& \sum_i x_i - x_3 - x_4 \geq 13 \\
& \sum_i x_i - x_4 - x_5 \geq 15 \\
& \sum_i x_i - x_5 - x_6 \geq 19 \\
& \sum_i x_i - x_6 - x_7 \geq 14 \\
& \sum_i x_i - x_7 - x_1 \geq 16 \\
& \sum_i x_i - x_1 - x_2 \geq 11 \\
& x_i \geq 0
\end{aligned}$$

Note: will need solution to be integral!

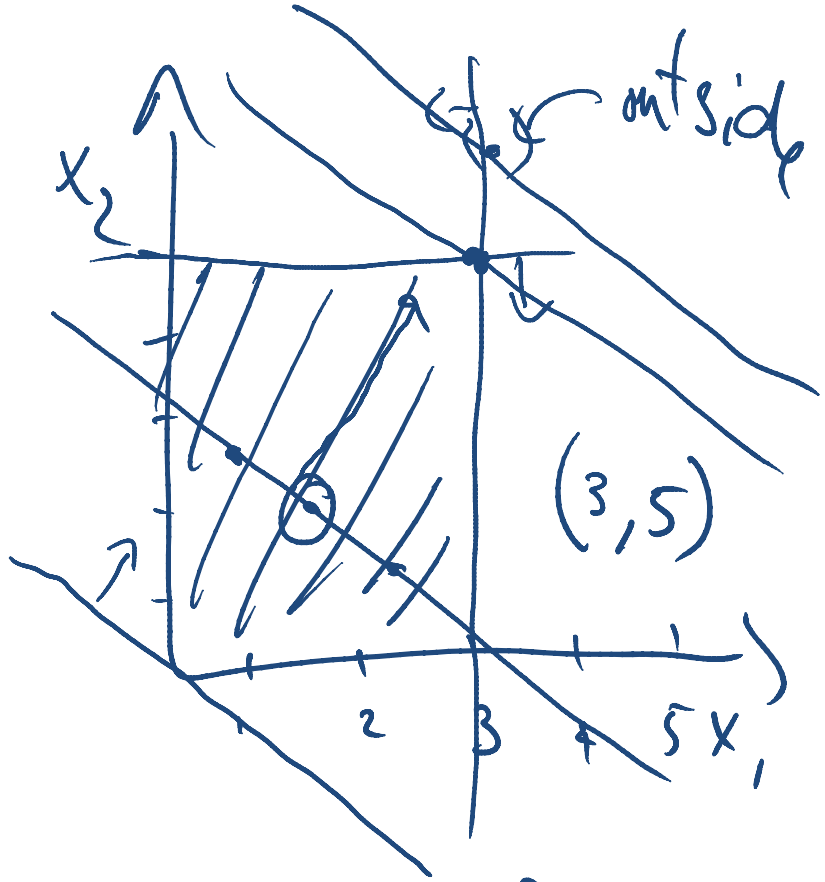
max/min ....

$x_1, \ldots$

s.t. $\leq$

$\geq$

$=$



$\max\ x_1 + x_2$

s.t. $x_1 \leq 3$ $\quad x_1, x_2 \geq 0$ → Linear Program

$x_2 \leq 5$ $\quad x_1, x_2 \in \mathbb{Z}$ → Integer Program

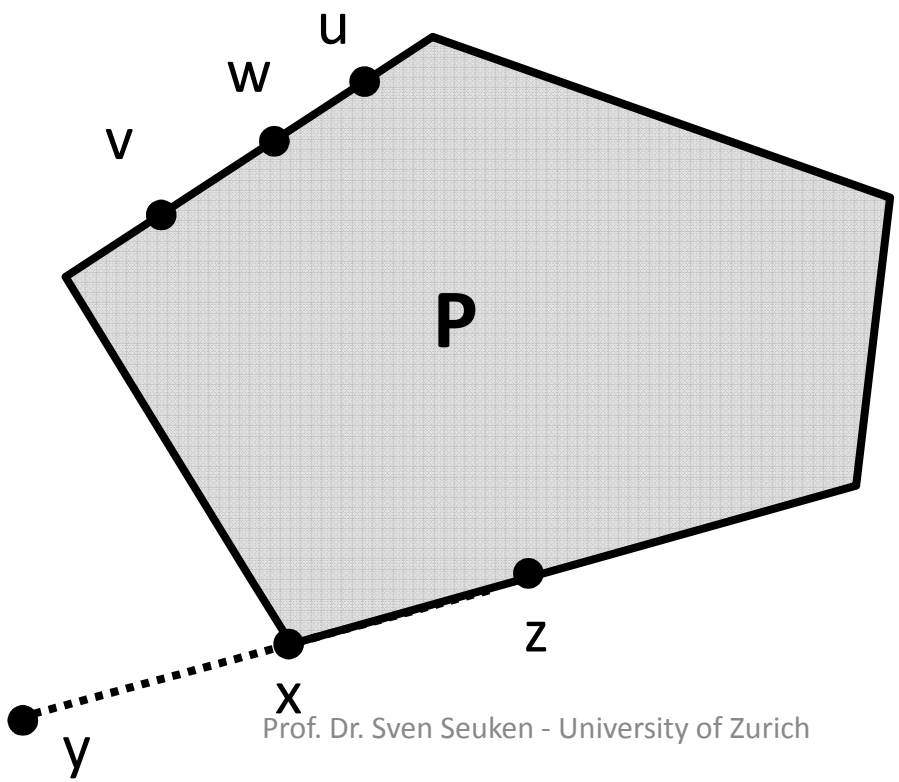
$x_1, x_2 \geq 0$

Polyhedron

- **Definition**. A **polyhedron** is a set that can be described in form **P**={x in R^n | Ax>=b}
- **Fact**: feasible set of any LP can be described in this way, and in particular {x in R^n | Ax=b, x>=0} is also a polyhedron.

- **Theorem**. Every polyhedron is a convex set.
- Why?

Extreme points



Algorithms for LPs

- Idea: Check every extreme point
  - With n constraints: 2^n extreme points $\rightarrow O(2^n)$
  - → takes exponential time
- Simplex Algorithm:
  - Move from extreme point to extreme point
  - Stop when local optimum (= global optimum)
  - Very fast in practice (e.g., more than 10,000 constraints)
  - Worst-case time complexity: exponential

Complexity of LPs?

- Solving an LP is in P!
- Thus, there exists a worst-case polynomial time algorithm for solving LPs:
  → interior-point method
- In the worst case faster than simplex method
- In the average case, slower than simplex…
- In practice:
  – CPLEX (IBM)

Integer Programming

Some uses of IP

- Scheduling problems
  - e.g., air taxi scheduling (assign each air taxi to a particular route)
- Strategic procurement
  - e.g., hospital system determining which suppliers to use for sourcing of medical/surgical equipment
- Electricity generation planning
  - e.g., developing a schedule to determine when to start-up plants, and levels to run at
- Clearing kidney exchanges
  - e.g., finding “donor chains”
- Computational biology
  - e.g. determining evolutionary trees (“phylogenetic trees”) from population variation data

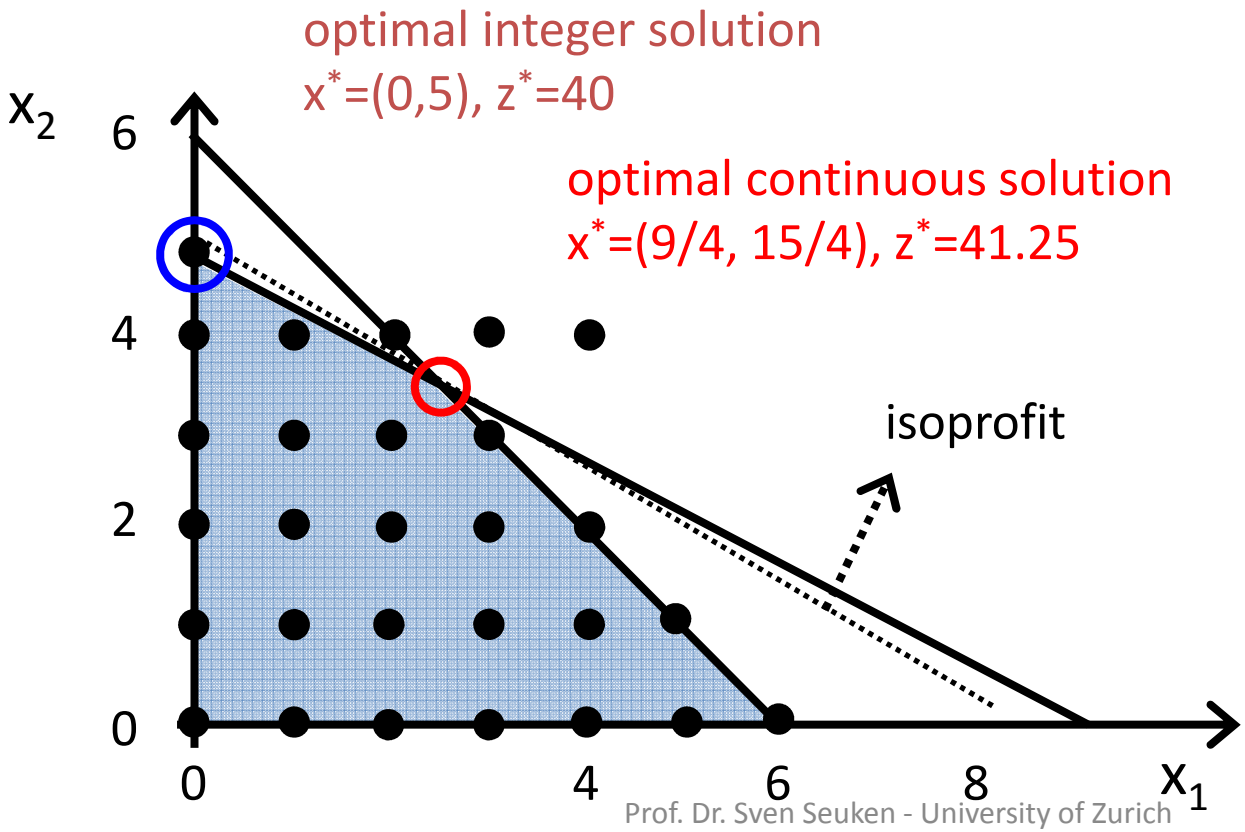
Shouldn’t IPs be easier to solve

- “There are fewer feasible solutions”

  – What is misleading about this argument?

$$\max\ 5x_1 + 8x_2$$
$$\text{s.t. } x_1 + x_2 <= 6$$
$$5x_1 + 9x_2 <= 45$$
$$x_1,\ x_2 >= 0, \text{ integer}$$

	cont	round off	nearest feas	integer
x_1	2.25	2	2	0
x_2	3.75	4	3	5
z	41.25	infeas	34	40



Example: Weighted Knapsack

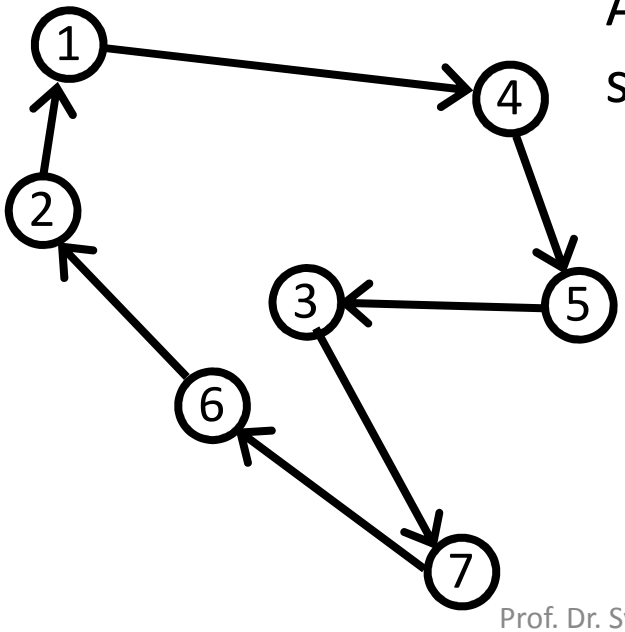
- E.g., Budget b available for investment in projects. Project j in {1,…,n} has cost ("size") a_j and value ("weight") c_j.
- Let $x_j = 1$ if project j selected, $x_j = 0$ otherwise
- $\max \sum_j c_j x_j$ (maximize value)

  s.t. $\sum_j a_j x_j <= b$ (budget constraint)

  x_j in {0,1} (no fractional projects)

Example: Traveling Salesman Problem (TSP)

- Salesman must visit each of N={1,…,n} cities once and then return to starting point. Cost $c_{ij} \geq 0$ to travel from i to j. Goal: find tour that minimizes total cost.



A feasible tour in a seven-city TSP

Examples: FedEx pick-up, machine placing modules on a circuit board, visiting colleges,…

- How many solutions?
- Starting at city 1, there are n-1 choices
- For the next city, n-2 choices,…
- (n-1)! feasible tours

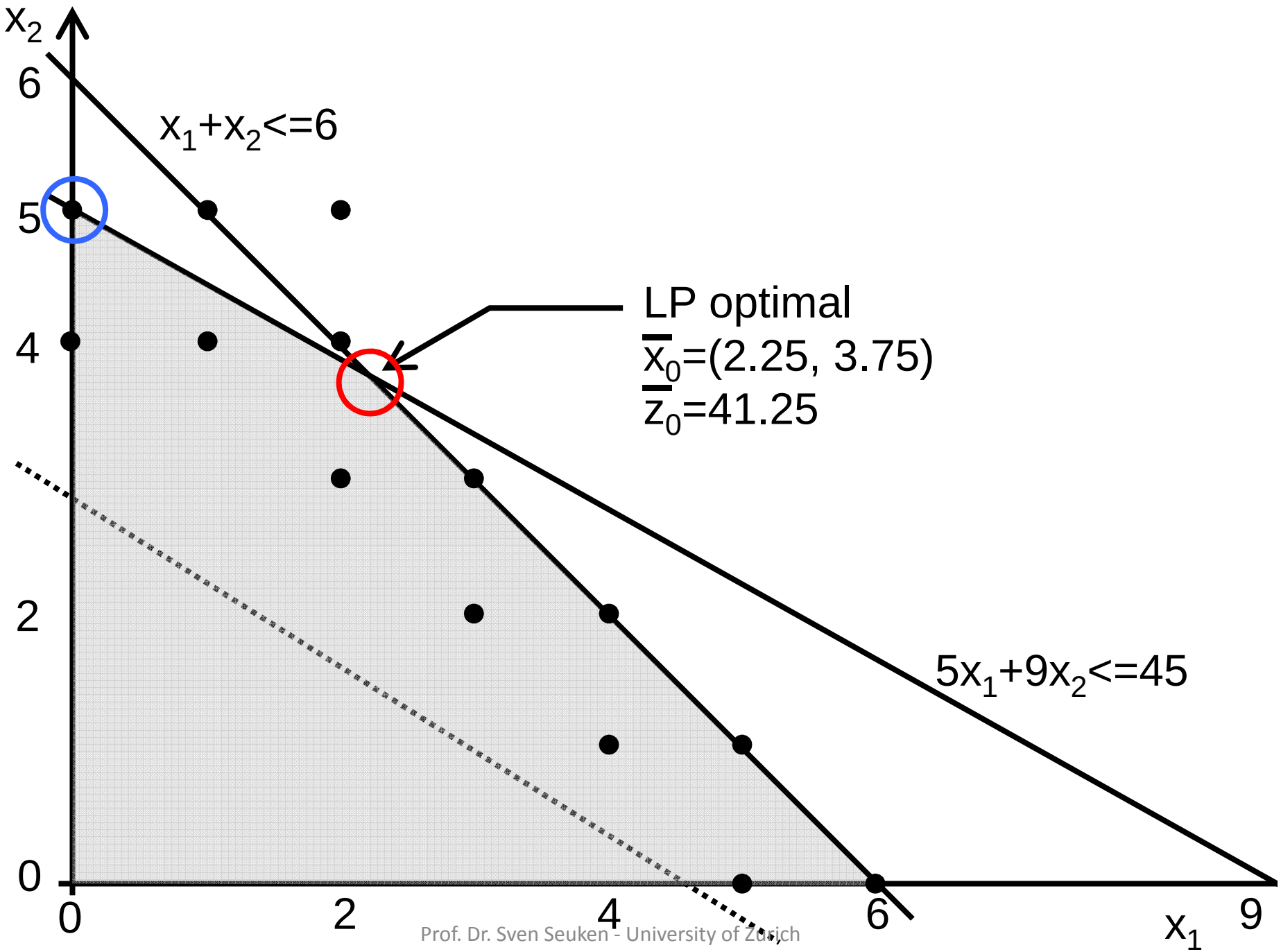
n	**log n**	**$n^{0.5}$**	**n^2**	**2^n**	**n!**
10	3.32	3.16	10^2	1.02×10^3	3.6×10^6
100	6.64	10.00	10^4	1.27×10^{30}	9.33×10^{157}
1000	9.97	31.62	10^6	1.07×10^{301}	4.02×10^{2567}

Branch and Bound: Solving IPs

- Ignore integrality constraints, solve the LP relaxation and hope the solution is integer!
- Example:

$$\begin{aligned} \max\ & 5x_1 + 8x_2 \\ \text{s.t. } & x_1 + x_2 <= 6 \\ & 5x_1 + 9x_2 <= 45 \\ & x_1,\ x_2 >= 0, \text{ integer} \end{aligned}$$

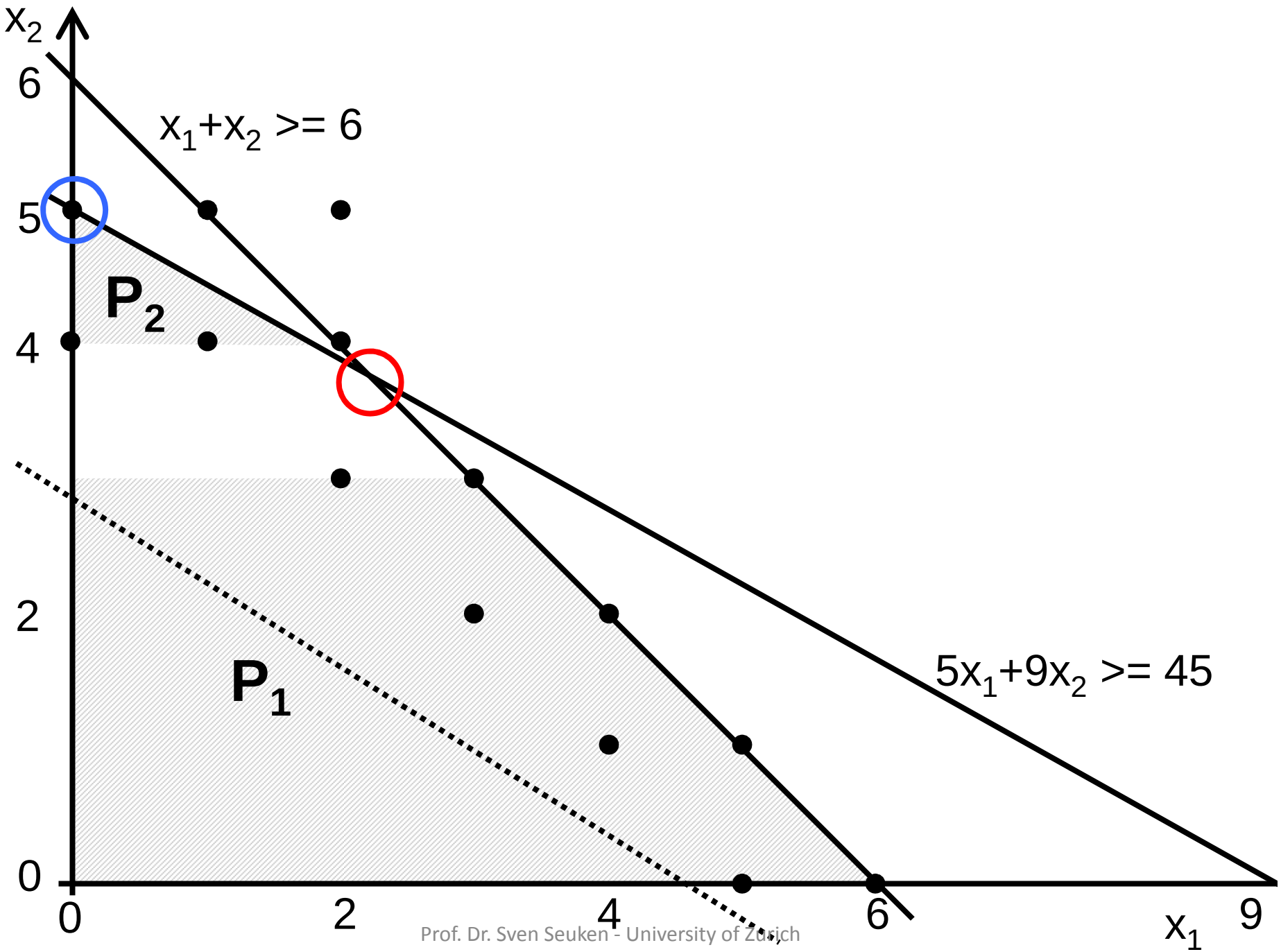
- Let S_0 denote feasible solution space of IP
- Obtain the initial LP relaxation by dropping the integrality constraints


x_2
6
5
4
2
0
$x_1+x_2<=6$
LP optimal
$\bar{x}_0=(2.25, 3.75)$
$\bar{z}_0=41.25$
$5x_1+9x_2<=45$
0
2
4
6
9
x_1

Caution: the use of subscripts (e.g. S_k, P_k) in BnB can be to denote the index of a subproblem, not the index of a variable

- Let P_0=LP(S_0)={x in R^2: x_1+x_2 <= 6, $5x_1+9x_2$ <= 45,x>=0}. z_0=41.25, x_0=(2.25,3.75)
- Consider variable x_2. Will be integer in solution to IP. Add constraints x_2 <= 3, x_2 >= 4.
- Let S_1 denote S_0 Å {x_2<= 3}; P_1=LP(S_1)
- Let S_2 denote S_0 Å {x_2 >= 4}; P_2=LP(S_2)

- Can plot the feasible region for P_1 and P_2


x_2
6
5
4
2
0
$x_1+x_2 >= 6$
P_2
P_1
$5x_1+9x_2 >= 45$
0
2
4
6
9
x_1

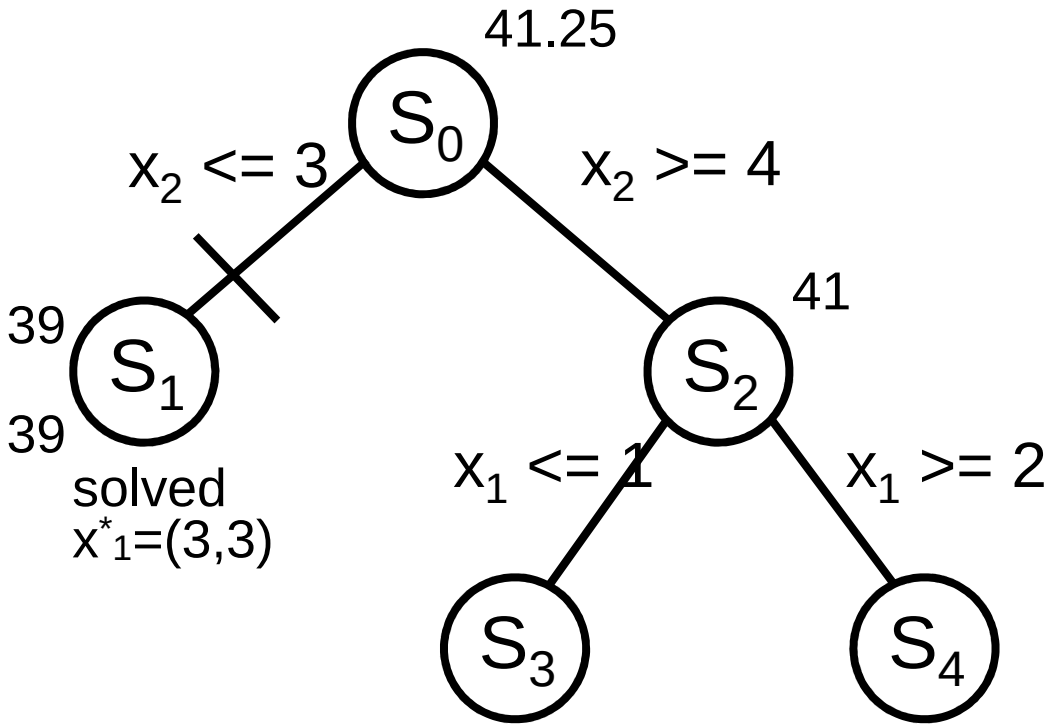
- Study two new IPs, with feasible solution spaces S_1 and S_2.
- Call S_1 and S_2 **subproblems** and this process **branching**.

- Consider S_1 first. Optimal solution to LP(S_1) is x_1=(3,3), z_1=39. Especially helpful! Our first feasible solution; write x^*_1=(3,3), z^*_1=39.

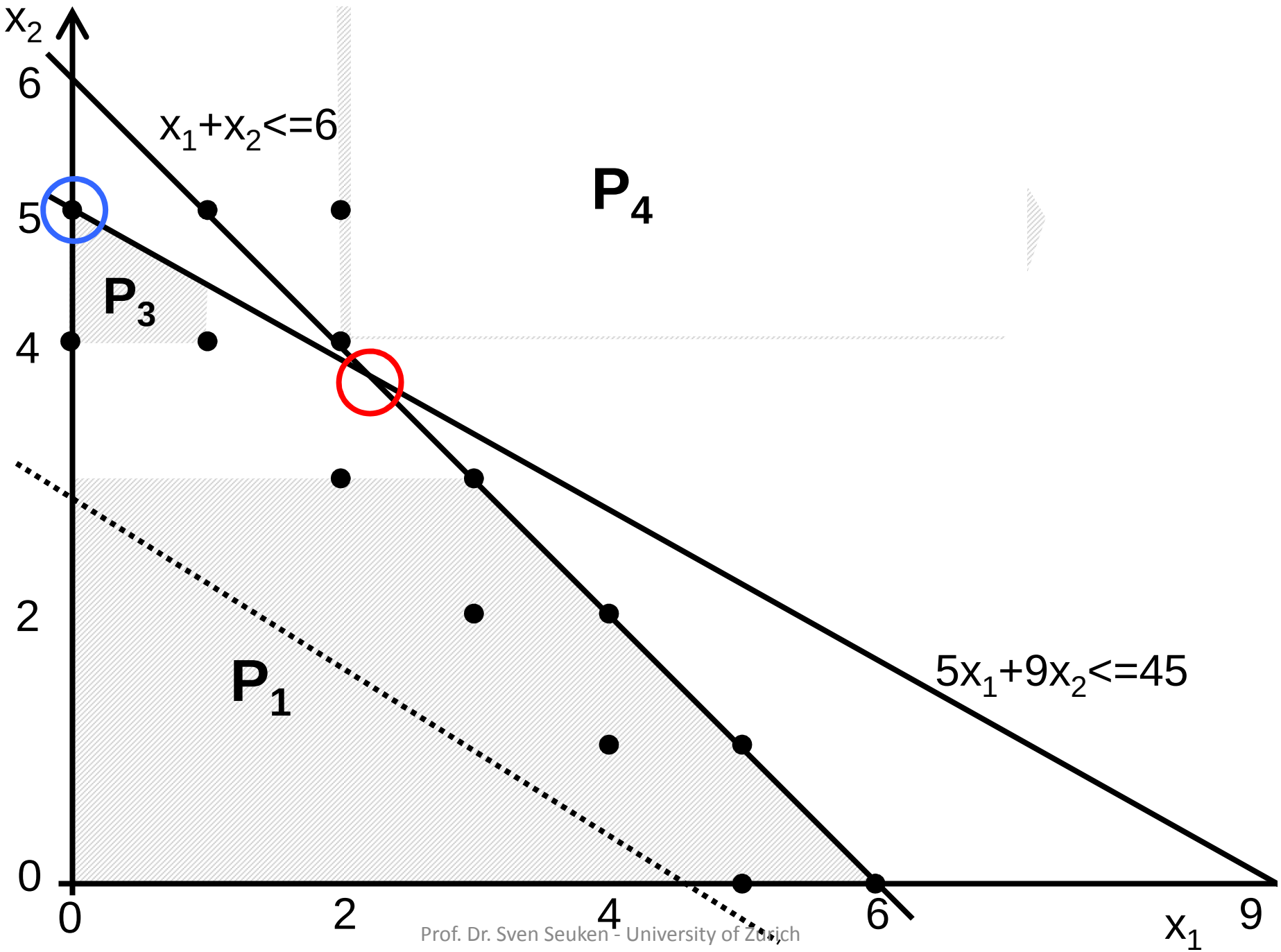
- Update the **incumbent**, z:=max(z, z^*_1) = max(-1,39) = 39, and x:=(3,3).

- Now consider S_2. Optimal solution to LP(S_2) is fractional.
- $\bar{x}_2$=(1.8,4), $\bar{z}_2$=41
- If $\bar{z}_2 \cdot z \leq 39$ would be done… any integer solution in S_2 would be smaller yet.
- But, not the case. Divide S_2 into two subproblems
- $S_3 = S_0$ Å $\{x_2 >= 4, x_1 >= 1\}$
- $S_4 = S_0$ Å $\{x_2 >= 4, x_1 >= 2\}$
- $S=S_1$ [S_3 [S_4, S_1Å S_3Å S_4 = ;

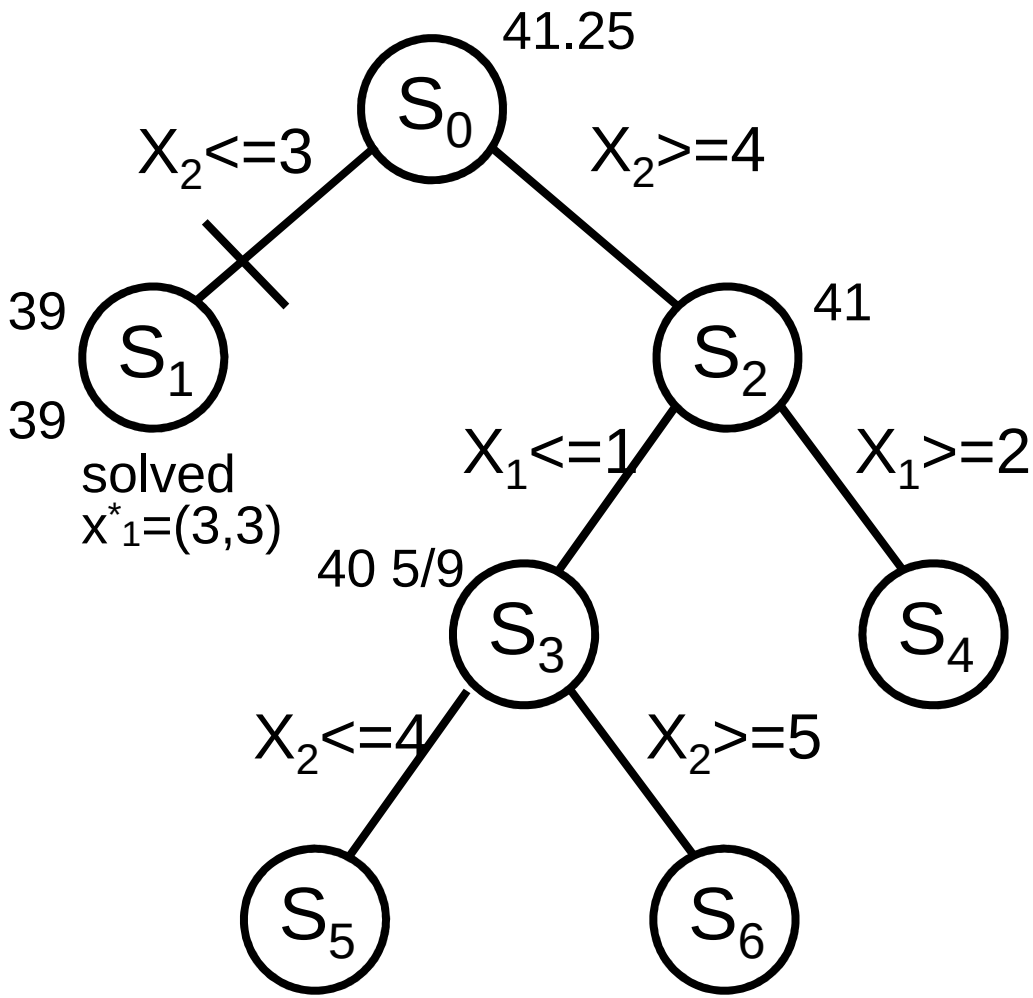
- Develop a **search tree** of IP subproblems



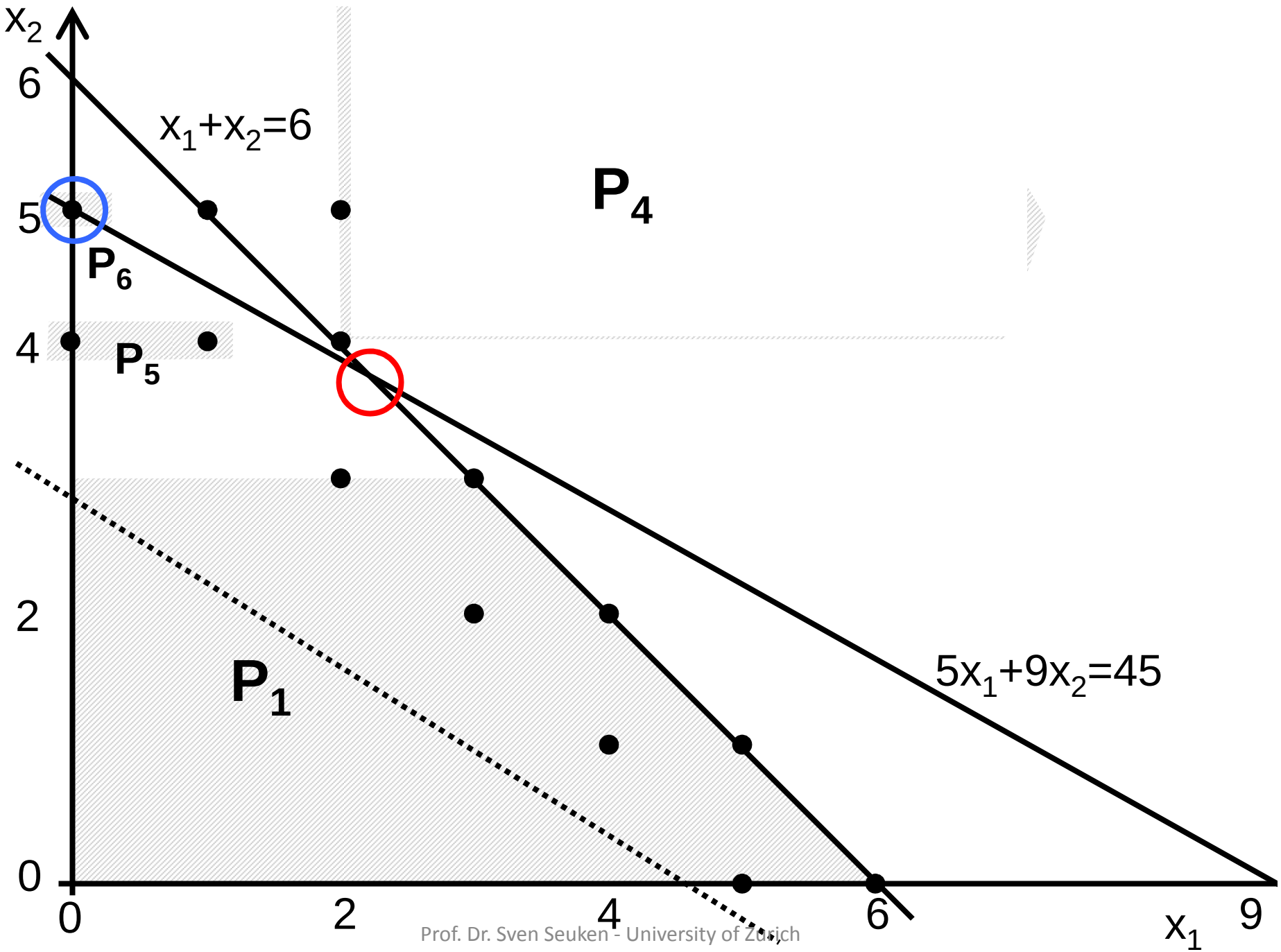
- Consider left branch. Solve LP(S_3)

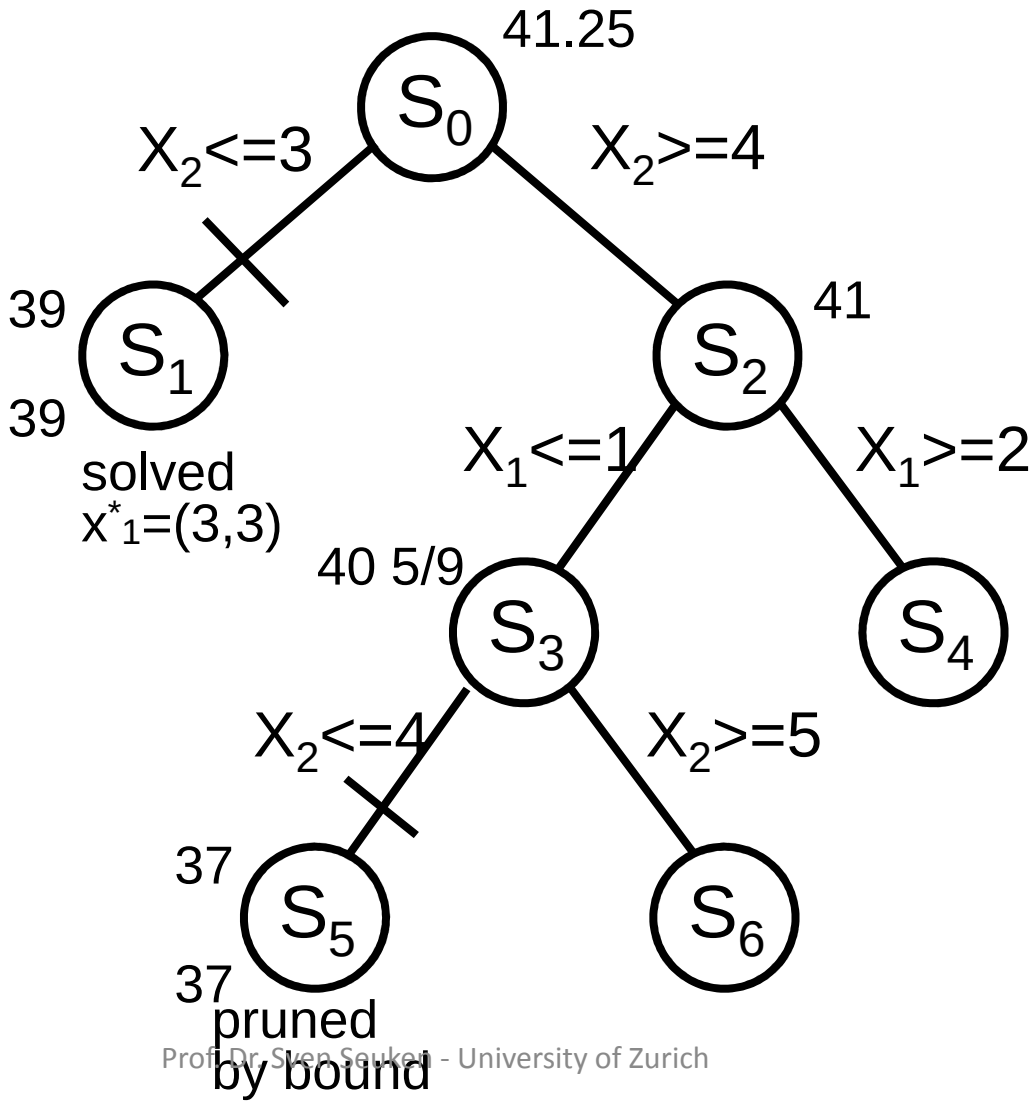

x_2
6
5
4
2
0
$x_1+x_2<=6$
P_4
P_3
P_1
$5x_1+9x_2<=45$
0
2
4
6
9
x_1

- Consider left branch. Solve LP(S_3)
- $\bar{x}_3$=(1, 4 4/9), $\bar{z}_3$ = 40 5/9
- Fractional, and $\bar{z}_3 > \underline{z} = 39$… again subdivide and continue
- $S_5 = S_0$ Å $\{x_2, 4, x_1 \cdot 1, x_2 \cdot 4\} = S_0$ Å $\{x_1 \cdot 1, x_2 = 4\}$
- $S_6 = S_0$ Å $\{x_2, 4, x_1 \cdot 1, x_2, 5\} = S_0$ Å $\{x_1 \cdot 1, x_2, 5\}$


41.25
S_0
X_2<=3
X_2>=4
39
S_1
39
solved
x^*_1=(3,3)
41
S_2
X_1<=1
X_1>=2
40 5/9
S_3
S_4
X_2<=4
X_2>=5
S_5
S_6

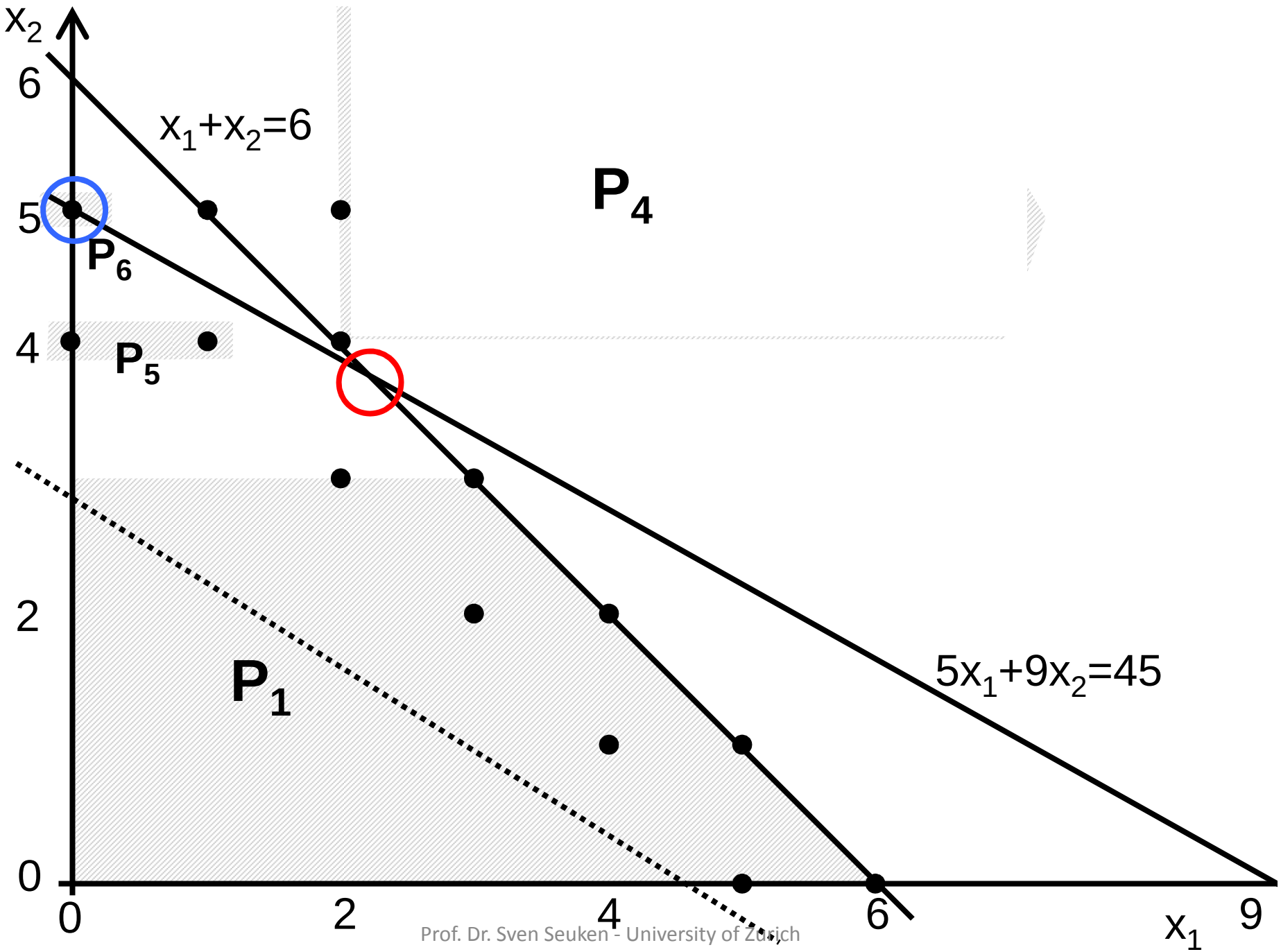
- At this point have 3 possible directions.
- Could consider subproblems S_4, S_5 or S_6
- To be specific in the example, consider **depth first search** and consider a child of the most recently considered subproblem; and take the **left branch**.
- Solve LP(S_5).
- $\bar{x}_5 = (1,4)$, $\bar{z}_5 = 37$
- $\bar{z}_5 \cdot z \underline{=} 39$. Ignore branch, any feasible solution is bounded above by 37.
- Say that this node is "pruned by bound."
- (Hence "branch and bound").


x_2
6
5
4
2
0
$x_1+x_2=6$
P_4
P_6
P_5
P_1
$5x_1+9x_2=45$
0
2
4
6
9
x_1

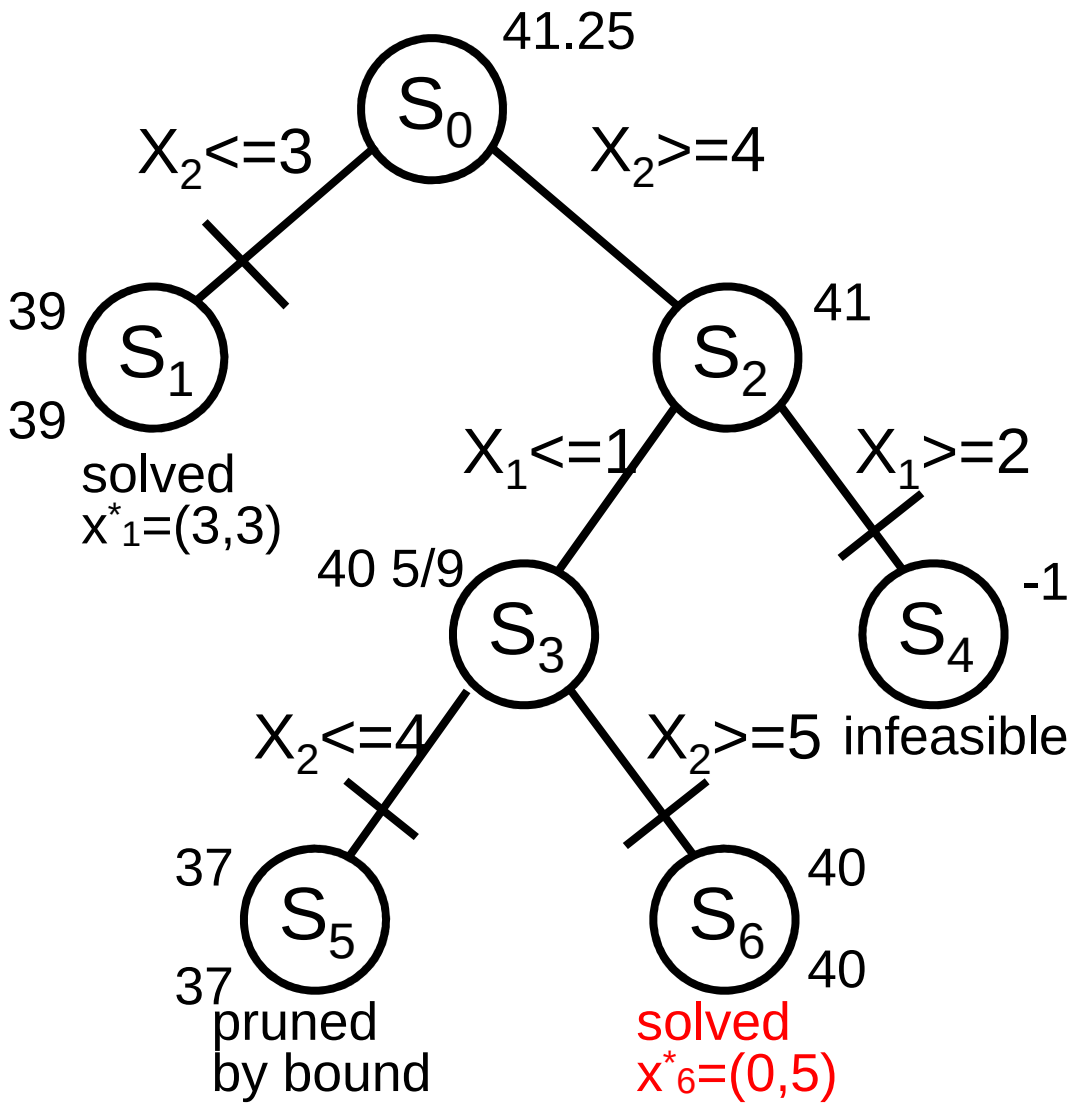

41.25
S0
X2<=3
X2>=4
39
S1
39
solved
x*1=(3,3)
S2
41
X1<=1
X1>=2
40 5/9
S3
S4
X2<=4
X2>=5
37
S5
37
pruned
by bound
S6

- Now at a leaf. Back up, find the first node with an unsolved problem.
- Right branch under S_3 (i.e., S_6)
- $\underline{z}_6$=40, $\underline{x}_6$=(0,5). Another feasible solution!
- Write z^*_6=40, x^*_6=(0,5)
- Since z^*_6>z, $\underline{z}$:=max(z,z^*_6) =max(39,40)=40, $\underline{x}$:=(0,5). A new incumbent.

- Finally, solve the problem under S_2 (i.e. S_4), and including any subproblems thereof. Infeasible.


x_2
6
5
4
2
0
$x_1+x_2=6$
P_4
P_6
P_5
P_1
$5x_1+9x_2=45$
0
2
4
6
9
x_1

Final search tree. Completely fathomed!



- Now completely **fathomed**. No branches left to explore.
- Can conclude that $x^*=x=(0,5)$, $z^*=z=40$ is the optimal solution.

Complexity of IPs

- Integer Programming Problems are **NP-hard**
- Thus, it is likely that even the best possible algorithms for solving IPs will need **exponential time** in the worst case!
- Depending on the amount of structure in the problem, powerful algorithms (CPLEX) can solve problems with hundreds of variables and constraints