

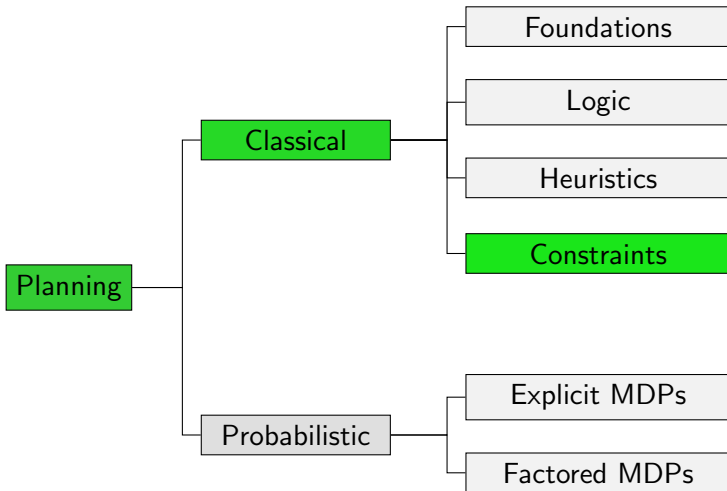
Planning and Optimization

E4. Linear & Integer Programming

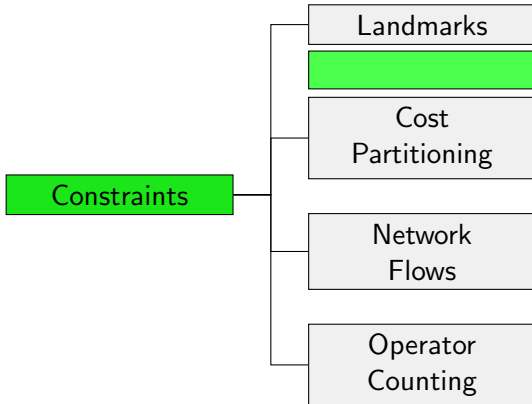
Malte Helmert and Gabriele Röger

Universität Basel

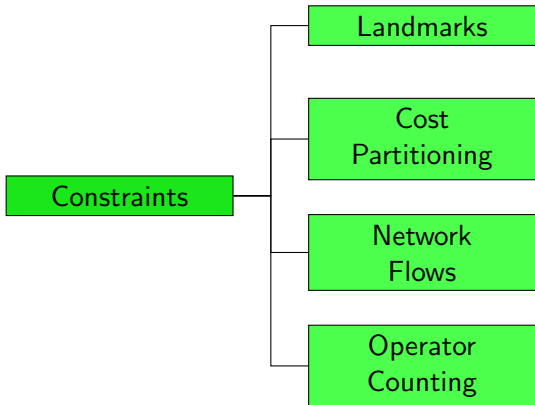
Content of this Course



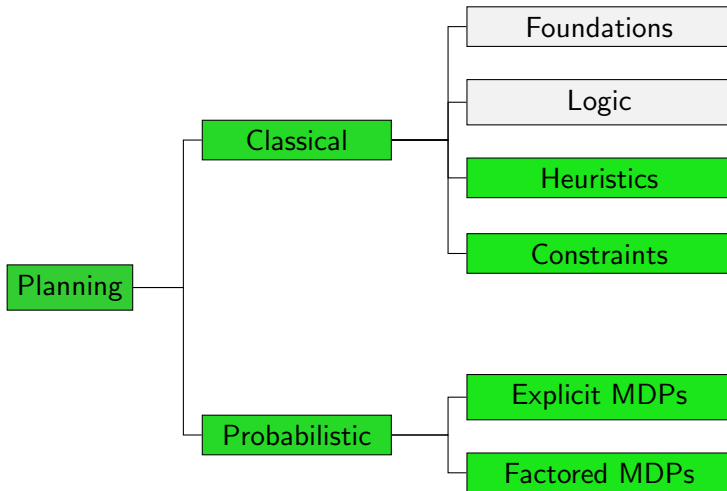
Content of this Course: Constraints (Timeline)



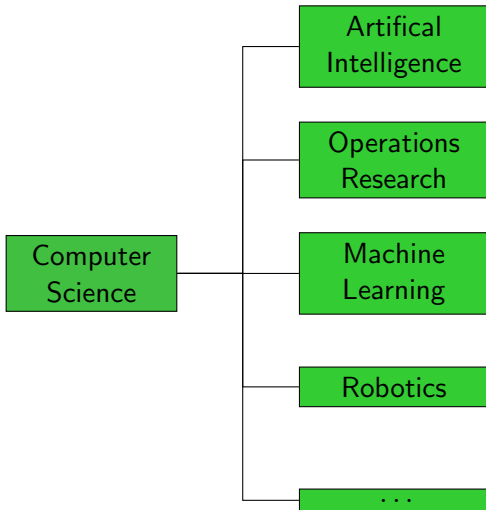
Content of this Course: Constraints (Relevance)



Content of this Course (Relevance)



Content of this Course (Relevance)



Integer Programs

Motivation

- This goes on beyond Computer Science
- Active **research** on IPs and LPs in
 - Operation Research
 - Mathematics
- Many **application** areas, for instance:
 - Manufacturing
 - Agriculture
 - Mining
 - Logistics
 - **Planning**
- As an application, we treat LPs / IPs as a **blackbox**
- We just look at **the fundamentals**

Motivation

Example (Optimization Problem)

Consider the following scenario:

- A factory produces two products A and B
- Selling a unit of B yields 5 times the profit of a unit of A.
- A client places the unusual order to “buy anything that can be produced on that day as long as the units of B do not exceed two plus twice the units of A.”
- The factory can produce at most 12 products per day.
- There is only material for 6 units of A
(there is enough material to produce any amount of B)

How many units of A and B does the client receive if the factory owner aims to maximize her profit?

Integer Program: Example

Let X_A and X_B be the (**integer**) number of produced A and B

Example (Optimization Problem as Integer Program)

$$X_A \geq 0, \quad X_B \geq 0$$

Example (Optimization Problem)

Integer Program: Example

Let X_A and X_B be the (**integer**) number of produced A and B

Example (Optimization Problem as Integer Program)

maximize $X_A + 5X_B$ subject to

$$X_A \geq 0, \quad X_B \geq 0$$

Example (Optimization Problem)

- “a unit of B yields 5 times the profit of a unit of A”
- “the factory owner aims to maximize her profit”

Integer Program: Example

Let X_A and X_B be the (**integer**) number of produced A and B

Example (Optimization Problem as Integer Program)

maximize $X_A + 5X_B$ subject to

$$2 + 2X_A \geq X_B$$

$$X_A \geq 0, \quad X_B \geq 0$$

Example (Optimization Problem)

- “the units of B may not exceed two plus twice the units of A.”

Integer Program: Example

Let X_A and X_B be the (**integer**) number of produced A and B

Example (Optimization Problem as Integer Program)

maximize $X_A + 5X_B$ subject to

$$2 + 2X_A \geq X_B$$

$$X_A + X_B \leq 12$$

$$X_A \geq 0, \quad X_B \geq 0$$

Example (Optimization Problem)

- “The factory can produce at most 12 units per day”

Integer Program: Example

Let X_A and X_B be the (**integer**) number of produced A and B

Example (Optimization Problem as Integer Program)

maximize $X_A + 5X_B$ subject to

$$2 + 2X_A \geq X_B$$

$$X_A + X_B \leq 12$$

$$X_A \leq 6$$

$$X_A \geq 0, \quad X_B \geq 0$$

Example (Optimization Problem)

- “There is only material for 6 units of A”

Integer Program: Example

Let X_A and X_B be the (**integer**) number of produced A and B

Example (Optimization Problem as Integer Program)

maximize $X_A + 5X_B$ subject to

$$2 + 2X_A \geq X_B$$

$$X_A + X_B \leq 12$$

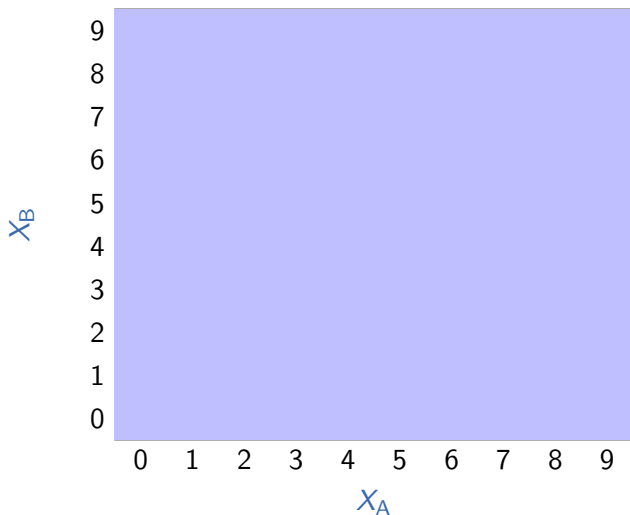
$$X_A \leq 6$$

$$X_A \geq 0, \quad X_B \geq 0$$

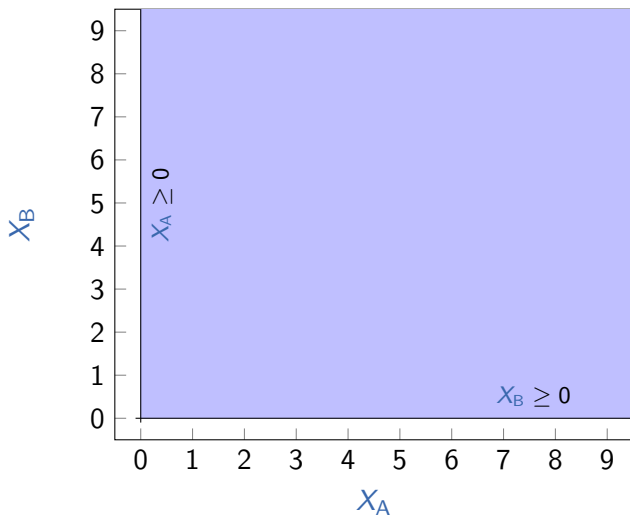
↪ unique optimal solution:

produce 4 A ($X_A = 4$) and 8 B ($X_B = 8$) for a profit of 44

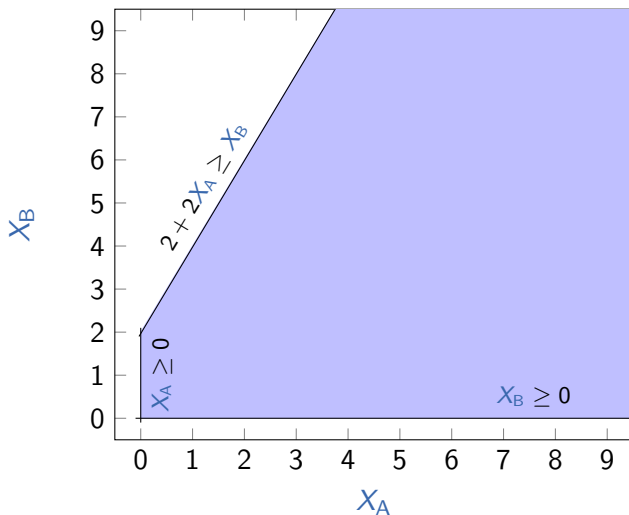
Integer Program Example: Visualization



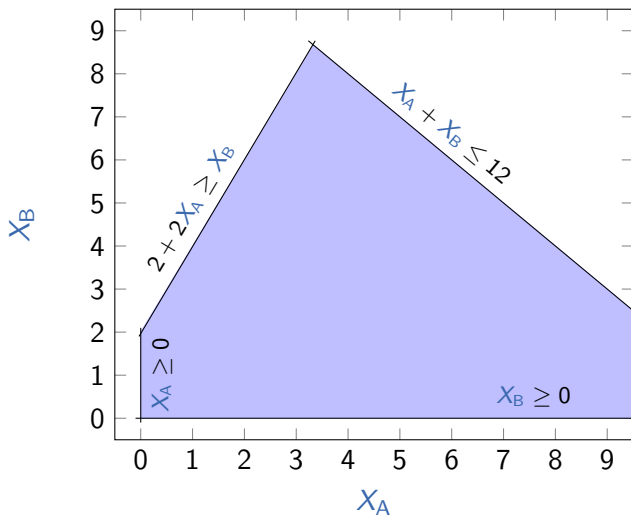
Integer Program Example: Visualization



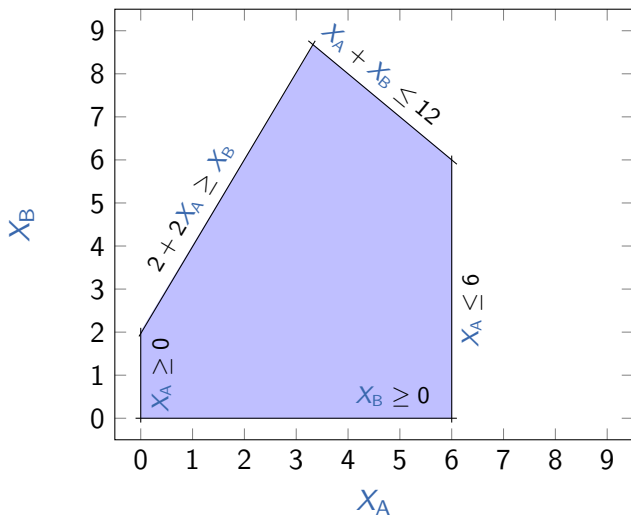
Integer Program Example: Visualization



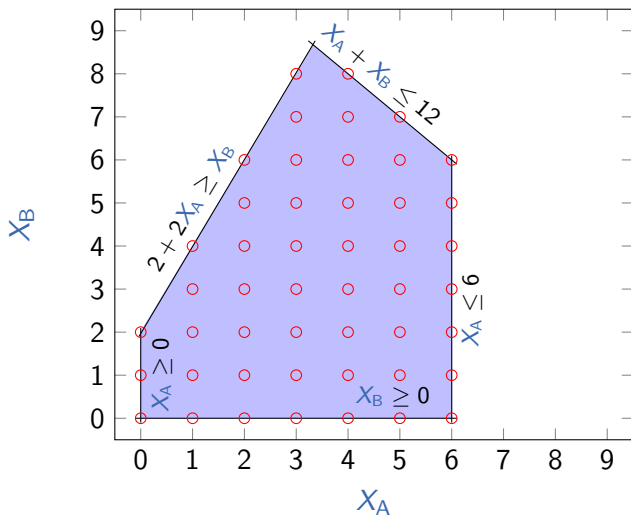
Integer Program Example: Visualization



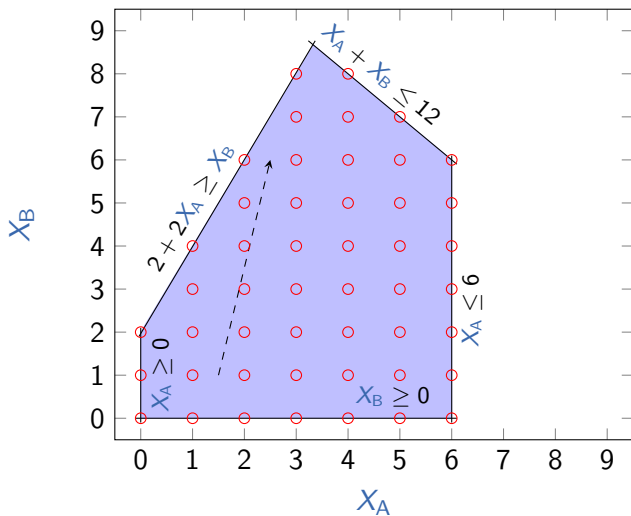
Integer Program Example: Visualization



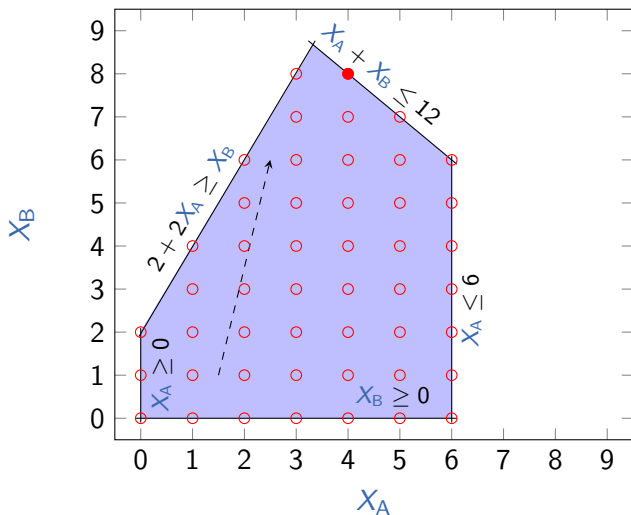
Integer Program Example: Visualization



Integer Program Example: Visualization



Integer Program Example: Visualization



Integer Programs

Integer Program

An **integer program (IP)** consists of:

- a finite set of **integer-valued variables** V
- a finite set of **linear inequalities** (constraints) over V
- an **objective function**, which is a linear combination of V
- which should be **minimized** or **maximized**.

Terminology

- An integer assignment to all variables in V is **feasible** if it satisfies the constraints.
- An integer program is **feasible** if there is such a feasible assignment. Otherwise it is **infeasible**.
- A feasible maximum (resp. minimum) problem is **unbounded** if the objective function can assume arbitrarily large positive (resp. negative) values at feasible assignments. Otherwise it is **bounded**.
- The **objective value** of a bounded feasible maximum (resp. minimum) problem is the maximum (resp. minimum) value of the objective function with a feasible assignment.

Three Classes of IPs

IPs fall into three classes:

- **bounded feasible:** IP is solvable and optimal solutions exist
- **unbounded feasible:** IP is solvable and arbitrarily good solutions exist
- **infeasible:** IP is unsolvable

Another Example

Example

minimize $3X_{o_1} + 4X_{o_2} + 5X_{o_3}$ subject to

$$X_{o_4} \geq 1$$

$$X_{o_1} + X_{o_2} \geq 1$$

$$X_{o_1} + X_{o_3} \geq 1$$

$$X_{o_2} + X_{o_3} \geq 1$$

$$X_{o_1} \geq 0, \quad X_{o_2} \geq 0, \quad X_{o_3} \geq 0, \quad X_{o_4} \geq 0$$

What example from a previous chapter does this IP encode?

Another Example

Example

minimize $3X_{o_1} + 4X_{o_2} + 5X_{o_3}$ subject to

$$X_{o_4} \geq 1$$

$$X_{o_1} + X_{o_2} \geq 1$$

$$X_{o_1} + X_{o_3} \geq 1$$

$$X_{o_2} + X_{o_3} \geq 1$$

$$X_{o_1} \geq 0, \quad X_{o_2} \geq 0, \quad X_{o_3} \geq 0, \quad X_{o_4} \geq 0$$

What example from a previous chapter does this IP encode?

↪ the minimum hitting set from Chapter E2

Complexity of Solving Integer Programs

- As an IP can compute an MHS, solving an IP must be **at least as complex** as computing an MHS
 - Reminder: MHS is a “classical” NP-complete problem
 - Good news: Solving an IP is **not harder**
- ↪ Finding solutions for IPs is **NP-complete**.

Complexity of Solving Integer Programs

- As an IP can compute an MHS, solving an IP must be **at least as complex** as computing an MHS
- Reminder: MHS is a “classical” NP-complete problem
- Good news: Solving an IP is **not harder**

↪ Finding solutions for IPs is **NP-complete**.

Removing the requirement that solutions must be **integer-valued** leads to a simpler problem

Linear Programs

Linear Programs

Linear Program

A **linear program** (LP) consists of:

- a finite set of **real-valued variables** V
- a finite set of **linear inequalities** (constraints) over V
- an **objective function**, which is a linear combination of V
- which should be **minimized** or **maximized**.

We use the introduced IP terminology also for LPs.

Mixed IPs (MIPs) generalize IPs and LPs:

some variables are integer-values, some are real-valued.

Linear Program: Example

Let X_A and X_B be the (**real-valued**) number of produced A and B

Example (Optimization Problem as **Linear Program**)

maximize $X_A + 5X_B$ subject to

$$2 + 2X_A \geq X_B$$

$$X_A + X_B \leq 12$$

$$X_A \leq 6$$

$$X_A \geq 0, \quad X_B \geq 0$$

Linear Program: Example

Let X_A and X_B be the (**real-valued**) number of produced A and B

Example (Optimization Problem as **Linear Program**)

maximize $X_A + 5X_B$ subject to

$$2 + 2X_A \geq X_B$$

$$X_A + X_B \leq 12$$

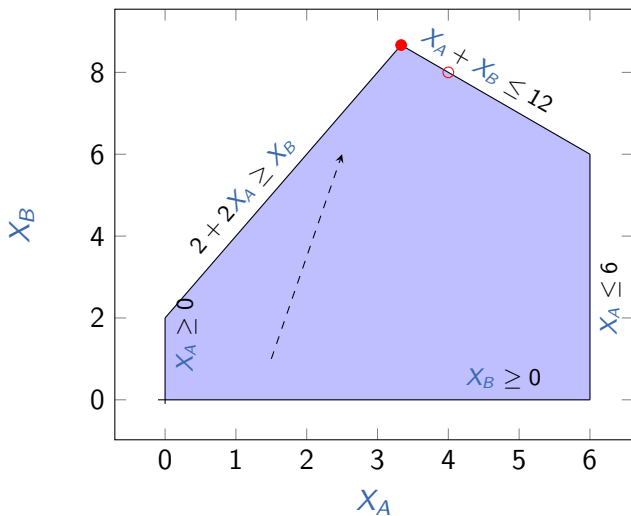
$$X_A \leq 6$$

$$X_A \geq 0, \quad X_B \geq 0$$

↪ unique optimal solution:

$$X_A = 3\frac{1}{3} \text{ and } X_B = 8\frac{2}{3} \text{ with objective value } 46\frac{2}{3}$$

Linear Program Example: Visualization



Solving Linear Programs

- Observation:

For an maximization problem, the objective value of the LP is **not lower** than the one of the IP.

Solving Linear Programs

- **Observation:**
For an maximization problem, the objective value of the LP is **not lower** than the one of the IP.
- **Complexity:**
LP solving is a **polynomial-time** problem.

Solving Linear Programs

- **Observation:**
For an maximization problem, the objective value of the LP is **not lower** than the one of the IP.
- **Complexity:**
LP solving is a **polynomial-time** problem.
- **Common idea:**
Approximate IP objective value with corresponding LP (**LP relaxation**).

LP Relaxation

Theorem (LP Relaxation)

The *LP relaxation* of an integer program is the problem that arises by removing the requirement that variables are integer-valued.

For a *maximization* (resp. minimization) problem, the objective value of the LP relaxation is an *upper* (resp. lower) *bound* on the value of the IP.

Proof idea.

Every feasible assignment for the IP is also feasible for the LP.

LP Relaxation of MHS heuristic

Example (Minimum Hitting Set)

minimize $3X_{o_1} + 4X_{o_2} + 5X_{o_3}$ subject to

$$X_{o_4} \geq 1$$

$$X_{o_1} + X_{o_2} \geq 1$$

$$X_{o_1} + X_{o_3} \geq 1$$

$$X_{o_2} + X_{o_3} \geq 1$$

$$X_{o_1} \geq 0, \quad X_{o_2} \geq 0, \quad X_{o_3} \geq 0, \quad X_{o_4} \geq 0$$

↪ optimal solution of LP relaxation:

$X_{o_4} = 1$ and $X_{o_1} = X_{o_2} = X_{o_3} = 0.5$ with objective value 6

↪ LP relaxation of MHS heuristic is **admissible**
and can be computed **in polynomial time**

Some LP Theory: Duality

Every LP has an alternative view (its **dual** LP).

- roughly: variables and constraints swap roles
- roughly: objective coefficients and bounds swap roles
- dual of maximization LP is minimization LP and vice versa
- dual of dual: original LP

Duality Theorem

Theorem (Duality Theorem)

*If a linear program is **bounded feasible**, then so is its dual, and their **objective values are equal**.*

(Proof omitted.)

The dual provides a different perspective on a problem.

Summary

Summary

- **Linear (and integer) programs** consist of an **objective function** that should be **maximized or minimized** subject to a set of given **linear constraints**.
- Finding solutions for **integer** programs is **NP-complete**.
- **LP solving** is a **polynomial time** problem.
- The dual of a maximization LP is a minimization LP and vice versa.
- The **dual** of a bounded feasible LP has the **same objective value**.

Further Reading

The slides in this chapter are based on the following excellent tutorial on LP solving:



Thomas S. Ferguson.

Linear Programming – A Concise Introduction.

UCLA, unpublished document available online.