DMP204
SCHEDULING,
TIMETABLING AND ROUTING

**Lecture 3**

# RCPSP and Mixed Integer Programming

Marco Chiarandini



## Outline

1. Scheduling
   - CPM/PERT
   - Resource Constrained Project Scheduling Model
2. Mathematical Programming
   - Introduction
   - Solution Algorithms





## Outline

1. Scheduling
   - CPM/PERT
   - Resource Constrained Project Scheduling Model
2. Mathematical Programming
   - Introduction
   - Solution Algorithms





## Outline

1. Scheduling
   - CPM/PERT
   - Resource Constrained Project Scheduling Model
2. Mathematical Programming
   - Introduction
   - Solution Algorithms

# Project Planning

| Milwaukee General Hospital Project | | | |
|---|---|---|---|
| Activity | Description | Immediate Predecessor | *Duration* |
| A | Build internal components | – | 2 |
| B | Modify roof and floor | – | 3 |
| C | Construct collection stack | A | 2 |
| D | Pour concrete and install frame | A,B | 4 |
| E | Build high-temperature burner | C | 4 |
| F | Install pollution control system | C | 3 |
| G | Install air pollution device | D,E | 5 |
| H | Inspect and test | F,G | 2 |



Scheduling
Math Programming
CPM/PERT
RCPSP

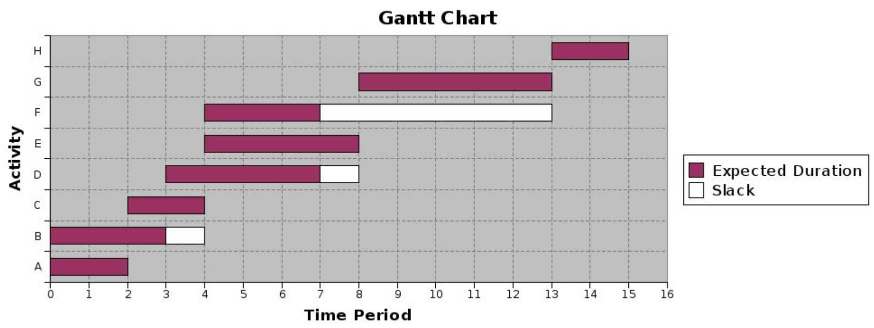
# Project Planning

| Milwaukee General Hospital Project | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Activity | Description | Immediate Predecessor | *Duration* | EST | EFT | LST | LFT | Slack |
| A | Build internal components | – | 2 | 0 | 2 | 0 | 2 | 0 |
| B | Modify roof and floor | – | 3 | 0 | 3 | 1 | 4 | 1 |
| C | Construct collection stack | A | 2 | 2 | 4 | 2 | 4 | 0 |
| D | Pour concrete and install frame | A,B | 4 | 3 | 7 | 6 | 10 | 3 |
| E | Build high-temperature burner | C | 4 | 4 | 8 | 6 | 10 | 2 |
| F | Install pollution control system | C | 3 | 4 | 7 | 10 | 13 | 6 |
| G | Install air pollution device | D,E | 5 | 8 | 13 | 8 | 13 | 0 |
| H | Inspect and test | F,G | 2 | 13 | 15 | 13 | 15 | 0 |
| | | *Expected project duration* | | | *15* | | | |



Scheduling
Math Programming
CPM/PERT
RCPSP

# Project Planning






Scheduling
Math Programming
CPM/PERT
RCPSP

# Project Planning

| Milwaukee General Hospital Projec | | | Expecte d | | | | | | Time Estimates | | | Activity Varianc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Activity | Description | Immediate Predecessor | *(a+4m+b)/(* | EST | EFT | LST | LFT | Slack | *a* | *m* | *b* | *((b-a)/6)^2* |
| A | Build internal components | – | 2 | 0 | 2 | 0 | 2 | 0 | 1 | 2 | 3 | 0.1111 |
| B | Modify roof and floor | – | 3 | 0 | 3 | 1 | 4 | 1 | 2 | 3 | 4 | 0.1111 |
| C | Construct collection stack | A | 2 | 2 | 4 | 2 | 4 | 0 | 1 | 2 | 3 | 0.1111 |
| D | 'our concrete and install fram | A,B | 4 | 3 | 7 | 4 | 8 | 1 | 2 | 4 | 6 | 0.4444 |
| E | 3uild high-temperature burne | C | 4 | 4 | 8 | 4 | 8 | 0 | 1 | 4 | 7 | 1.0000 |
| F | nstall pollution control systen | C | 3 | 4 | 7 | 10 | 13 | 6 | 1 | 2 | 9 | 1.7778 |
| G | Install air pollution device | D,E | 5 | 8 | 13 | 8 | 13 | 0 | 3 | 4 | 11 | 1.7778 |
| H | Inspect and test | F,G | 2 | 13 | 15 | 13 | 15 | 0 | 1 | 2 | 3 | 0.1111 |
| | | *Expected project duration* | | | *15* | | *Variance of project duration* | | | | | *3.1111* |

# Outline

1. Scheduling
   CPM/PERT
   Resource Constrained Project Scheduling Model

2. Mathematical Programming
   Introduction
   Solution Algorithms





# RCPSP

## Resource Constrained Project Scheduling Model

**Given:**

- activities (jobs) $j = 1, \ldots, n$
- renewable resources $i = 1, \ldots, m$
- amount of resources available $R_i$
- processing times $p_j$
- amount of resource used $r_{ij}$
- precedence constraints $j \rightarrow k$





# RCPSP

## Resource Constrained Project Scheduling Model

**Given:**

- activities (jobs) $j = 1, \ldots, n$
- renewable resources $i = 1, \ldots, m$
- amount of resources available $R_i$
- processing times $p_j$
- amount of resource used $r_{ij}$
- precedence constraints $j \rightarrow k$

Further generalizations

- Time dependent resource profile $R_i(t)$
  given by $(t_i^\mu, R_i^\mu)$ where $0 = t_i^1 < t_i^2 < \ldots < t_i^{m_i} = T$
  Disjunctive resource, if $R_k(t) = \{0, 1\}$; cumulative resource, otherwise





# RCPSP

## Resource Constrained Project Scheduling Model

**Given:**

- activities (jobs) $j = 1, \ldots, n$
- renewable resources $i = 1, \ldots, m$
- amount of resources available $R_i$
- processing times $p_j$
- amount of resource used $r_{ij}$
- precedence constraints $j \rightarrow k$

Further generalizations

- Time dependent resource profile $R_i(t)$
  given by $(t_i^\mu, R_i^\mu)$ where $0 = t_i^1 < t_i^2 < \ldots < t_i^{m_i} = T$
  Disjunctive resource, if $R_k(t) = \{0, 1\}$; cumulative resource, otherwise
- Multiple modes for an activity $j$
  processing time and use of resource depends on its mode $m$: $p_{jm}$, $r_{jkm}$.

# Modeling



### Assignment 1

- A contractor has to complete $n$ activities.
- The duration of activity $j$ is $p_j$
- each activity requires a crew of size $W_j$.
- The activities are not subject to precedence constraints.
- The contractor has $W$ workers at his disposal
- his objective is to complete all $n$ activities in minimum time.





### Assignment 2

- Exams in a college may have different duration.
- The exams have to be held in a gym with $W$ seats.
- The enrollment in course $j$ is $W_j$ and
- all $W_j$ students have to take the exam at the same time.
- The goal is to develop a timetable that schedules all $n$ exams in minimum time.
- Consider both the cases in which each student has to attend a single exam as well as the situation in which a student can attend more than one exam.





### Assignment 3

- In a basic high-school timetabling problem we are given $m$ classes $c_1, \dots, c_m$,
- $h$ teachers $a_1, \dots, a_h$ and
- $T$ teaching periods $t_1, \dots, t_T$.
- Furthermore, we have lectures $i = l_1, \dots, l_n$.
- Associated with each lecture is a unique teacher and a unique class.
- A teacher $a_j$ may be available only in certain teaching periods.
- The corresponding timetabling problem is to assign the lectures to the teaching periods such that
  - each class has at most one lecture in any time period
  - each teacher has at most one lecture in any time period,
  - each teacher has only to teach in time periods where he is available.





### Assignment 4

- A set of jobs $J_1, \dots, J_g$ are to be processed by auditors $A_1, \dots, A_m$.
- Job $J_l$ consists of $n_l$ tasks ($l = 1, \dots, g$).
- There are precedence constraints $i_1 \to i_2$ between tasks $i_1, i_2$ of the same job.
- Each job $J_l$ has a release time $r_l$, a due date $d_l$ and a weight $w_l$.
- Each task must be processed by exactly one auditor. If task $i$ is processed by auditor $A_k$, then its processing time is $p_{ik}$.
- Auditor $A_k$ is available during disjoint time intervals $[s_k^\nu, l_k^\nu]$ ( $\nu = 1, \dots, m$) with $l_k^\nu < s_k^\nu$ for $\nu = 1, \dots, m_k - 1$.
- Furthermore, the total working time of $A_k$ is bounded from below by $H_k^-$ and from above by $H_k^+$ with $H_k^- \leq H_k^+$ $(k = 1, \dots, m)$.
- We have to find an assignment $\alpha(i)$ for each task $i = 1, \dots, n := \sum_{l=1}^{g} n_l$ to an auditor $A_{\alpha(i)}$ such that
  - each task is processed without preemption in a time window of the assigned auditor
  - the total workload of $A_k$ is bounded by $H_k^-$ and $H_k^k$ for $k = 1, \dots, m$.
  - the precedence constraints are satisfied,
  - all tasks of $J_l$ do not start before time $r_l$, and
  - the total weighted tardiness $\sum_{l=1}^{g} w_l T_l$ is minimized.

# Outline







# Outline







# Mathematical Programming

## Linear, Integer, Nonlinear



program = optimization problem



# Mathematical Programming

## Linear, Integer, Nonlinear



program = optimization problem

$$\begin{array}{ll} \min & f(x) \\ & g_i(x) = 0, \quad i = 1, 2, \ldots, k \\ & h_j(x) \leq 0, \quad j = 1, 2, \ldots, m \\ & x \in \mathbf{R}^n \end{array}$$

general (nonlinear) program (NLP)

# Mathematical Programming

## Linear, Integer, Nonlinear



program = optimization problem

$$\begin{array}{ll} \min & f(x) \\ & g_i(x) = 0, \quad i = 1, 2, \ldots, k \\ & h_j(x) \leq 0, \quad j = 1, 2, \ldots, m \\ & x \in \mathbf{R}^n \end{array}$$

general (nonlinear) program (NLP)

$$\begin{array}{ll} \min & c^T x \\ & Ax = a \\ & Bx \leq b \\ & x \geq 0 \\ & (x \in \mathbf{R}^n) \end{array}$$

linear program (LP)

$$\begin{array}{ll} \min & c^T x \\ & Ax = a \\ & Bx \leq b \\ & x \geq 0 \\ & (x \in \mathbf{Z}^n) \\ & (x \in \{0,1\}^n) \end{array}$$

integer (linear) program (IP, MIP)



# Linear Programming



Linear Program in standard form

$$\begin{array}{ll} \min & c_1 x_1 + c_2 x_2 + \ldots c_n x_n \\ s.t. & a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n = b_1 \\ & a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n = b_2 \\ & \vdots \\ & a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n = b_n \\ & x_1, x_2, \ldots, x_n \geq 0 \end{array}$$

$$\begin{array}{ll} \min & c^T x \\ & Ax = b \\ & x \geq 0 \end{array}$$



# Historic Roots



- 1939 L. V. Kantorovitch: Foundations of linear programming (Nobel Prize 1975)
- George J. Stigler's 1945 (Nobel Prize 1982) "Diet Problem": "the first linear program"
  find the cheapest combination of foods that will
  satisfy the daily requirements of a person
  Army's problem had 77 unknowns and 9 constraints.
  `http://www.mcs.anl.gov/home/otc/Guide/CaseStudies/diet/index.html`
- 1947 G.B. Dantzig: Invention of the simplex algorithm
- Founding fathers:
  - 1950s Dantzig: Linear Programming 1954, the Beginning of IP
    G. Dantzig, D.R. Fulkerson, S. Johnson TSP with 49 cities
  - 1960s Gomory: Integer Programming



# LP Theory



- Max-Flow Min-Cut Theorem
  The maximal (s,t)-flow in a capaciatetd network is equal to the minimal capacity of an (s,t)-cut
- The Duality Theorem of Linear Programming

$$\begin{array}{ll} \max & c^T x \\ & Ax \leq b \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \min & y^T b \\ & y^T A \geq c^T \\ & y \geq 0 \end{array}$$

If feasible solutions to both the primal and the dual problem in a pair of dual LP problems exist, then there is an optimum solution to both systems and the optimal values are equal.

# LP Theory



- Max-Flow Min-Cut Theorem
  does not hold if several source-sink relations are given (multicommodity flow)

- The Duality Theorem of Integer Programming

$$\begin{array}{ll} \max & c^T x \\ & Ax \leq b \\ & x \geq 0 \\ & x \in \mathbf{Z}^n \end{array} \quad \leq \quad \begin{array}{ll} \min & y^T b \\ & y^T A \geq c^T \\ & y \geq 0 \\ & y \in \mathbf{Z}^n \end{array}$$



# LP Solvability



- Linear programs can be solved in polynomial time with
  the Ellipsoid Method (Khachiyan, 1979)
  Interior Point Methods (Karmarkar, 1984, and others)
- Open: is there a strongly polynomial time algorithm for the solution of LPs?
- Certain variants of the Simplex Algorithm run - under certain conditions - in expected polynomial time (Borgwardt, 1977...)
- Open: Is there a polynomial time variant of the Simplex Algorithm?



# IP Solvability



- Theorem
  Integer, 0/1, and mixed integer programming are NP-hard.
- Consequence
  - special cases
  - special purposes
  - heuristics



# Outline

# Solution Algorithms



- Algorithms for the solution of nonlinear programs
- Algorithms for the solution of linear programs
  - 1) Fourier-Motzkin Elimination (hopeless)
  - 2) The Simplex Method (good, above all with duality)
  - 3) The Ellipsoid Method (total failure)
  - 4) Interior-Point/Barrier Methods (good)
- Algorithms for the solution of integer programs
  - 1) Branch & Bound
  - 2) Cutting Planes



# Nonlinear programming



- Iterative methods that solve the equation and inequality systems representing the necessary local optimality conditions.
- Steepest descent (Kuhn-Tucker sufficient conditions)
- Newton method
- Subgradient method



# Linear programming



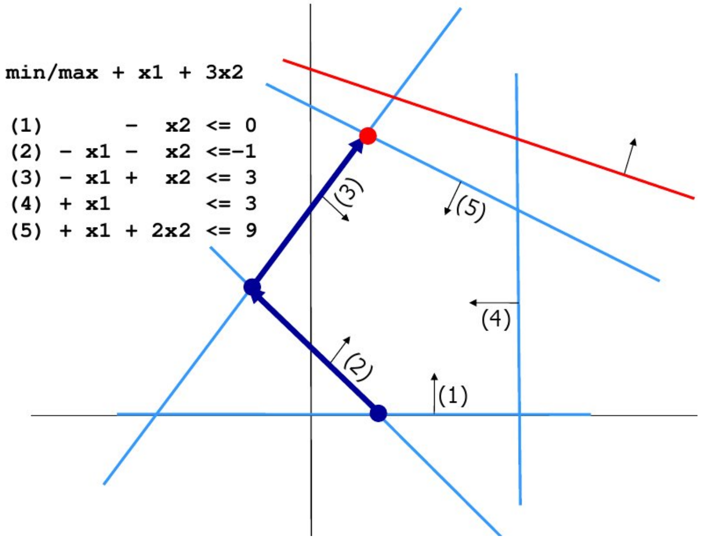
The Simplex Method

- Dantzig, 1947: primal Simplex Method
- Lemke, 1954; Beale, 1954: dual Simplex Method
- Dantzig, 1953: revised Simplex Method
- ....
- Underlying Idea: Find a vertex of the set of feasible LP solutions (polyhedron) and move to a better neighbouring vertex, if possible.



# The simplex method



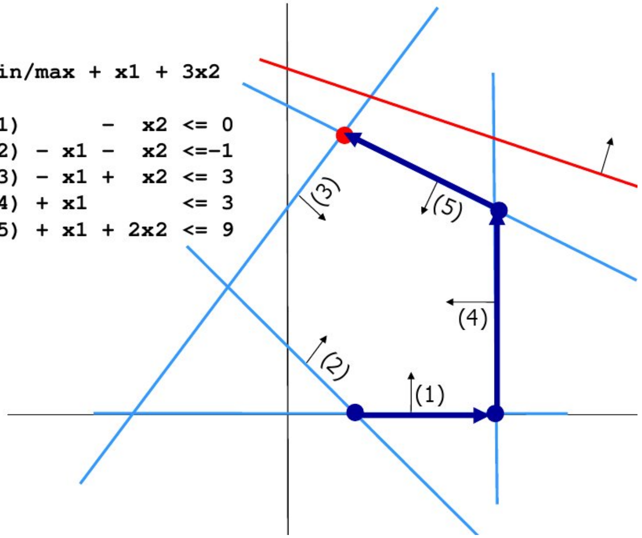


# The simplex method



```
min/max + x1 + 3x2

(1)      -  x2 <= 0
(2) - x1 -  x2 <=-1
(3) - x1 +  x2 <= 3
(4) + x1       <= 3
(5) + x1 + 2x2 <= 9
```






# The simplex method



Hirsch Conjecture
If P is a polytope of dimension n with m facets then every vertex of P can be reached from any other vertex of P on a path of length at most m-n.

In the example before: m=5, n=2 and m-n=3, conjecture is true.

At present, not even a polynomial bound on the path length is known.
Best upper bound: Kalai, Kleitman (1992): The diameter of the graph of an n-dimensional polyhedron with m facets is at most m(log n+1).
Lower bound: Holt, Klee (1997): at least m-n (m, n large enough).



# Integer Programming (easy)



special „simple" combinatorial optimization problems Finding a:

- minimum spanning tree
- shortest path
- maximum matching
- maximal flow through a network
- cost-minimal flow
- ...

solvable in polynomial time by special purpose algorithms



# Integer Programming (hard)



special „hard" combinatorial optimization problems

- traveling salesman problem
- location and routing
- set-packing, partitioning, -covering
- max-cut
- linear ordering
- scheduling (with a few exceptions)
- node and edge colouring
- ...

NP-hard (in the sense of complexity theory)
The most successful solution techniques employ linear programming.

# Integer Programming (hard)



- 1) Branch & Bound
- 2) Cutting Planes

Branch & cut, Branch & Price (column generation), Branch & Cut & Price



# Summary



- We can solve today explicit LPs with
  - up to 500,000 of variables and
  - up to 5,000,000 of constraints routinely

  in relatively short running times.
- We can solve today structured implicit LPs (employing column generation and cutting plane techniques) in special cases with hundreds of million (and more) variables and almost infinitely many constraints in acceptable running times. (Examples: TSP, bus circulation in Berlin)

[Martin Grötschel, Block Course at TU Berlin, "Combinatorial Optimization at Work", 2005 http://co-at-work.zib.de/berlin/]