DM87
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 2

Complexity hierarchies, PERT, Mathematical programming

Marco Chiarandini

Outline

Outline

Outline

Complexity Hierarchy

A problem $\mathcal{A}$ is reducible to $\mathcal{B}$ if a procedure for $\mathcal{B}$ can be used also for $\mathcal{A}$.

Ex: $1||\sum C_j \;\propto\; 1||\sum w_j C_j$

Complexity hierarchy describes relationships between different scheduling problems.

Interest in characterizing the borderline: polynomial vs NP-hard problems

Problems Involving Numbers

Partition

- **Input:** finte set A and a size $s(a) \in \mathbf{Z}^+$ for each $a \in A$
- **Question:** is there a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)?$$

3-Partition

- **Input:** set A of $3m$ elements, a bound $B \in \mathbf{Z}^+$, and a size $s(a) \in \mathbf{Z}^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$
- **Question:** can A be partitioned into m diskoint sets $A_1, \ldots, A_m$ such that for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B$ (note that each A_i must therefore contain exactly three elements from A)?

Complexity Hierarchy of Problems

TABLE D.1 POLYNOMIAL TIME SOLVABLE PROBLEMS

Single machine	Parallel machines	Shops
$1 \mid r_j, p_j = 1, prec \mid \sum C_j$	$P2 \mid p_j = 1, prec \mid L_{\max}$	$O2 \mid\mid C_{\max}$
$1 \mid r_j, prmp \mid \sum C_j$	$P2 \mid p_j = 1, prec \mid \sum C_j$	
$1 \mid tree \mid \sum w_j C_j$		$Om \mid r_j, prmp \mid L_{\max}$
	$Pm \mid p_j = 1, tree \mid C_{\max}$	
$1 \mid prec \mid L_{\max}$	$Pm \mid prmp, tree \mid C_{\max}$	$F2 \mid block \mid C_{\max}$
$1 \mid r_j, prmp, prec \mid L_{\max}$	$Pm \mid p_j = 1, outtree \mid \sum C_j$	$F2 \mid nwt \mid C_{\max}$
	$Pm \mid p_j = 1, intree \mid L_{\max}$	
$1 \mid\mid \sum U_j$	$Pm \mid prmp, intree \mid L_{\max}$	$Fm \mid p_{ij} = p_j \mid \sum C_j$
$1 \mid r_j, prmp \mid \sum U_j$		$Fm \mid p_{ij} = p_j \mid L_{\max}$
$1 \mid r_j, p_j = 1 \mid \sum w_j U_j$	$Q2 \mid prmp, prec \mid C_{\max}$	$Fm \mid p_{ij} = p_j \mid \sum U_j$
	$Q2 \mid r_j, prmp, prec \mid L_{\max}$	
$1 \mid r_j, p_j = 1 \mid \sum w_j T_j$		$J2 \mid\mid C_{\max}$
	$Qm \mid r_j, p_j = 1 \mid C_{\max}$	
	$Qm \mid r_j, p_j = 1 \mid \sum C_j$	
	$Qm \mid prmp \mid \sum C_j$	
	$Qm \mid p_j = 1 \mid \sum w_j C_j$	
	$Qm \mid p_j = 1 \mid L_{\max}$	
	$Qm \mid prmp \mid \sum U_j$	
	$Qm \mid p_j = 1 \mid \sum w_j U_j$	
	$Qm \mid p_j = 1 \mid \sum w_j T_j$	
	$Rm \mid\mid \sum C_j$	
	$Rm \mid r_j, prmp \mid L_{\max}$	

TABLE D.2 NP-HARD PROBLEMS IN THE ORDINARY SENSE

Single machine	Parallel machines	Shops
$1 \mid\mid \sum w_j U_j$ (*)	$P2 \mid\mid C_{\max}$ (*)	$O2 \mid prmp \mid \sum C_j$
$1 \mid r_j, prmp \mid \sum w_j U_j$ (*)	$P2 \mid r_j, prmp \mid \sum C_j$	
	$P2 \mid\mid \sum w_j C_j$ (*)	$O3 \mid\mid C_{\max}$
$1 \mid\mid \sum T_j$ (*)	$P2 \mid r_j, prmp \mid \sum U_j$	$O3 \mid prmp \mid \sum w_j U_j$
	$Pm \mid prmp \mid \sum w_j C_j$	
	$Qm \mid\mid \sum w_j C_j$ (*)	
	$Rm \mid r_j \mid C_{\max}$ (*)	
	$Rm \mid\mid \sum w_j U_j$ (*)	
	$Rm \mid prmp \mid \sum w_j U_j$	

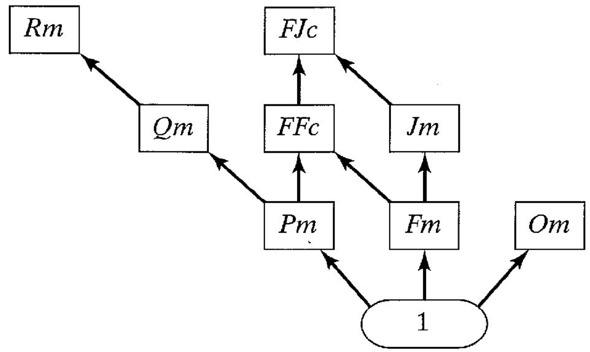
TABLE D.3 STRONGLY NP-HARD PROBLEMS

Single machine	Parallel machines	Shops
$1 \mid s_{jk} \mid C_{\max}$	$P2 \mid chains \mid C_{\max}$	$F2 \mid r_j \mid C_{\max}$
	$P2 \mid chains \mid \sum C_j$	$F2 \mid r_j, prmp \mid C_{\max}$
$1 \mid r_j \mid \sum C_j$	$P2 \mid prmp, chains \mid \sum C_j$	$F2 \mid\mid \sum C_j$
$1 \mid prec \mid \sum C_j$	$P2 \mid p_j = 1, tree \mid \sum w_j C_j$	$F2 \mid prmp \mid \sum C_j$
$1 \mid r_j, prmp, tree \mid \sum C_j$		$F2 \mid\mid L_{\max}$
$1 \mid r_j, prmp \mid \sum w_j C_j$	$R2 \mid prmp, chains \mid C_{\max}$	$F2 \mid prmp \mid L_{\max}$
$1 \mid r_j, p_j = 1, tree \mid \sum w_j C_j$		
$1 \mid p_j = 1, prec \mid \sum w_j C_j$		$F3 \mid\mid C_{\max}$
		$F3 \mid prmp \mid C_{\max}$
$1 \mid r_j \mid L_{\max}$		$F3 \mid nwt \mid C_{\max}$
$1 \mid r_j \mid \sum U_j$		$O2 \mid r_j \mid C_{\max}$
$1 \mid p_j = 1, chains \mid \sum U_j$		$O2 \mid\mid \sum C_j$
		$O2 \mid prmp \mid \sum w_j C_j$
$1 \mid r_j \mid \sum T_j$		$O2 \mid\mid L_{\max}$
$1 \mid p_j = 1, chains \mid \sum T_j$		
$^*1 \mid\mid \sum w_j T_j$		$O3 \mid prmp \mid \sum C_j$
		$J2 \mid recrc \mid C_{\max}$
		$J3 \mid p_{ij} = 1, recrc \mid C_{\max}$

http://www.mathematik.uni-osnabrueck.de/research/OR/class/

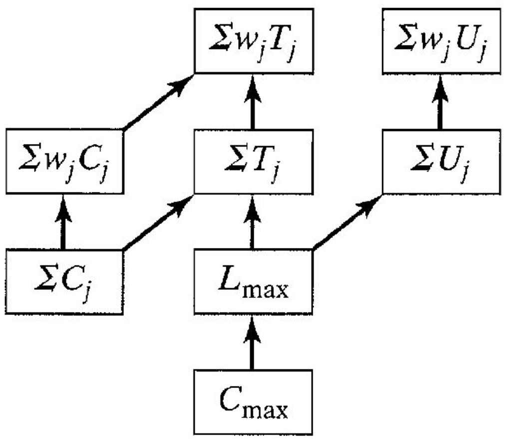
Complexity Hierarchy

Elementary reductions for machine environment

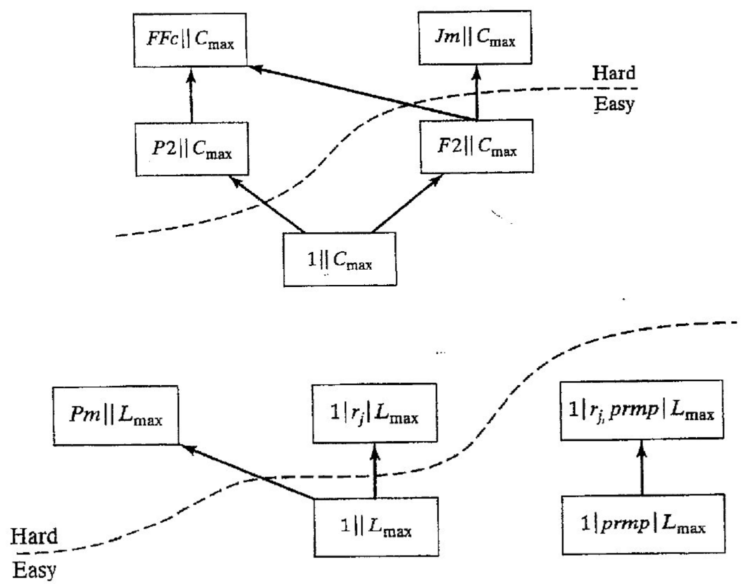


Complexity Hierarchy

Elementary reductions for regular objective functions



Complexity Hierarchy of Problems



Outline

1. Resume
2. Complexity Hierarchy
3. CPM/PERT
4. Mathematical Programming

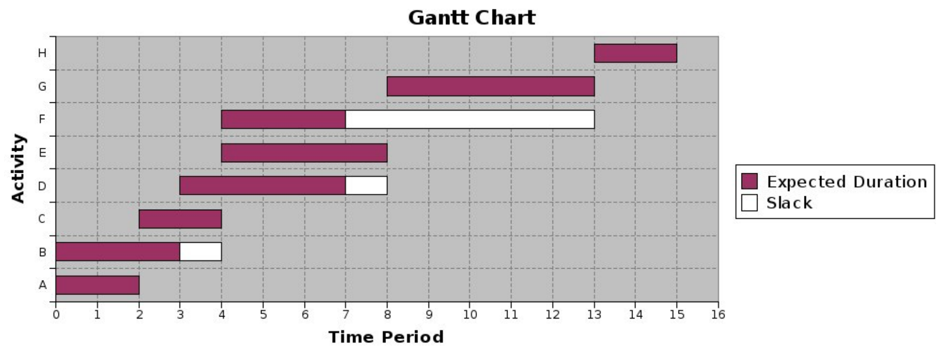
Project Planning

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

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***			

Project Planning



Project Planning

Milwaukee General Hospital Projec

Activity	Description	Immediate Predecessor	Expected (a+4m+b)/6	EST	EFT	LST	LFT	Slack	Time Estimates a	m	b	Activity Varianc ((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	Pour concrete and install frame	A,B	4	3	7	4	8	1	2	4	6	0.4444
E	Build high-temperature burne	C	4	4	8	4	8	0	1	4	7	1.0000
F	Install pollution control system	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

Linear, Integer, Nonlinear Programming

program = optimization problem

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

general (nonlinear) program (NLP)

$$\begin{aligned} \min \quad & c^T x \\ & Ax = a \\ & Bx \le b \\ & x \ge 0 \\ & (x \in \mathbf{R}^n) \end{aligned}$$

linear program (LP)

$$\begin{aligned} \min \quad & c^T x \\ & Ax = a \\ & Bx \le b \\ & x \ge 0 \\ & (x \in \mathbf{Z}^n) \\ & (x \in \{0,1\}^n) \end{aligned}$$

integer (linear) program (IP, MIP)

Linear Programming

Linear Program in standard form

$$\begin{aligned} \min \quad & c_1x_1 + c_2x_2 + \dots c_nx_n \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ & \vdots \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_n \\ & x_1, x_2, \dots, x_n \ge 0 \end{aligned}$$

$$\begin{aligned} \min \quad & c^T x \\ & Ax = b \\ & x \ge 0 \end{aligned}$$

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.h
- 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

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

  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

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

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

- 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

Algorithms for 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

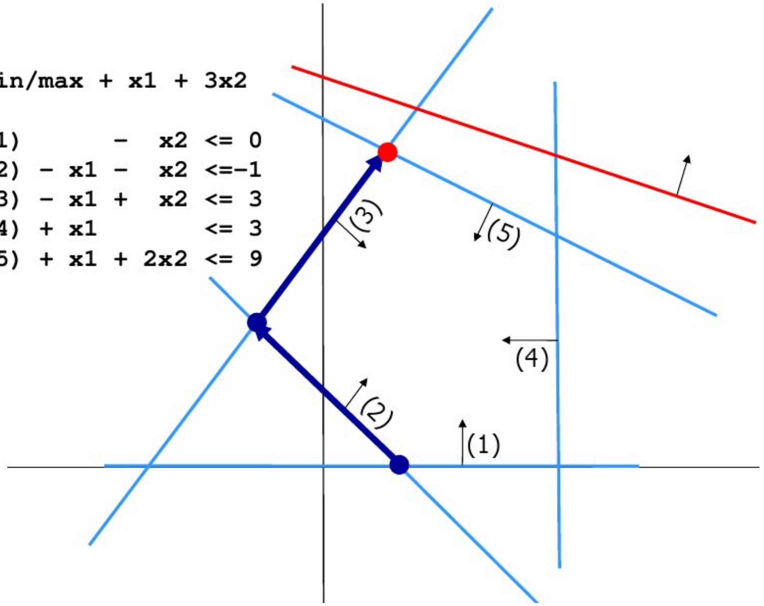
Algorithms for linear programming

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.

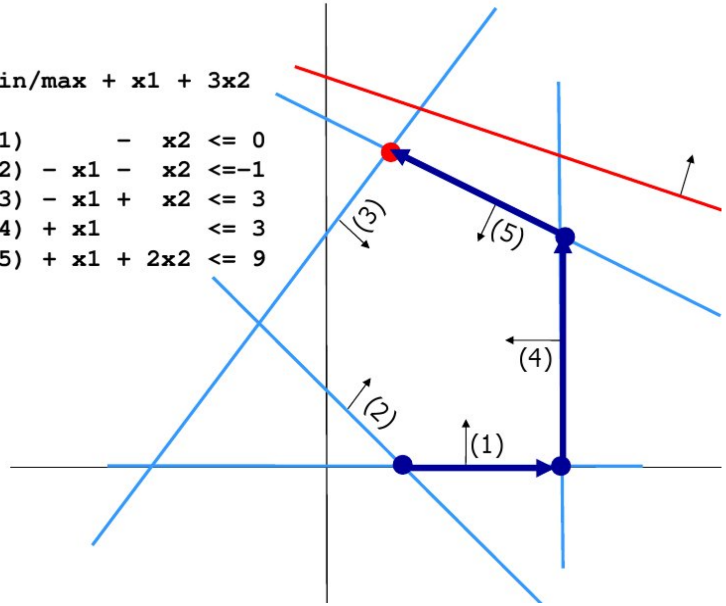
The simplex method

```
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

```
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).

Algorithms for Integer Programming

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

Algorithms for Integer Programs

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.

Algorithms for Integer Programs

- 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/]