Graph Theory and Optimization
Integer Linear Programming

Nicolas Nisse

Université Côte d'Azur, Inria, CNRS, I3S, France

October 2018

Outline

Linear Programme (reminder)

Linear programmes can be written under the standard form:

$$\begin{array}{lrcll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\ & x_j & \geq & 0 & \text{for all } 1 \leq j \leq n. \end{array}$$

- the problem is a maximization;
- all constraints are inequalities (and not equations);
- all variables $x_1, \cdots, x_n$ are non-negative.

Linear Programme (Real variables) can be solved in polynomial-time in the number of variables and constraints (e.g., ellipsoid method)

Linear Programme (reminder)

Linear programmes can be written under the standard form:

$$
\begin{array}{lrcll}
\text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\
\text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\
 & x_j & \geq & 0 & \text{for all } 1 \leq j \leq n.
\end{array}
$$

- the problem is a maximization;
- all constraints are inequalities (and not equations);
- all variables $x_1, \cdots, x_n$ are non-negative.

Linear Programme (Real variables) can be solved in polynomial-time in the number of variables and constraints (e.g., ellipsoid method)

Integer Linear Programme

Integer Linear programmes:

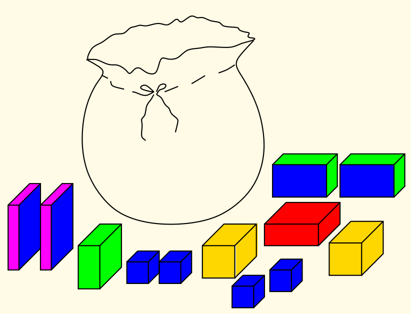
$$
\begin{array}{llcll}
\text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\
\text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\
 & x_j & \in & \mathbb{N} & \text{for all } 1 \leq j \leq n.
\end{array}
$$

- the problem is a maximization;
- all constraints are inequalities (and not equations);
- all variables $x_1, \cdots, x_n$ are Integers.

Integer Linear Programme is NP-complete in general!

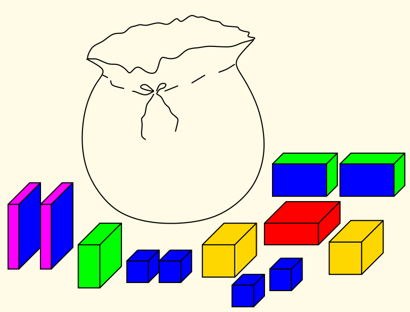
Outline

Knapsack Problem (Weakly NP-hard)



- Data:
  - a knapsack with maximum weight 15 Kg
  - 12 objects with
    - a weight w_i
    - a value v_i
- Objective: which objects should be chosen to maximize the value carried while not exceeding 15 Kg?

Knapsack Problem (Weakly NP-hard)



- Data:
  - a knapsack with maximum weight 15 Kg
  - 12 objects with
    - a weight w_i
    - a value v_i
- Objective: which objects should be chosen to maximize the value carried while not exceeding 15 Kg?

$$\begin{aligned} \textbf{max} \quad & \sum_{1 \le i \le 12} v_i x_i \\ \textbf{subject to} \quad & \sum_{1 \le i \le 12} w_i x_i \le 15 \\ & x_i \in \{0,1\} \end{aligned}$$

Minimum Vertex Cover (NP-hard)

Let $G = (V, E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Minimum Vertex Cover (NP-hard)

Let $G = (V, E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Solution: $K \subseteq V$ $\Rightarrow$ variables x_v, for each $v \in V$

$x_v = 1$ if $v \in K$, $x_v = 0$ otherwise.

Minimum Vertex Cover (NP-hard)

Let $G=(V,E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that “touch" every edge

Solution: $K \subseteq V$ $\Rightarrow$ variables x_v, for each $v \in V$

$x_v = 1$ if $v \in K$, $x_v = 0$ otherwise.

Objective function: minimize $|K|$ minimize $\sum_{v \in V} x_v$

Minimum Vertex Cover (NP-hard)

Let $G=(V,E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that “touch" every edge

Solution: $K \subseteq V$ $\Rightarrow$ variables x_v, for each $v \in V$

$x_v = 1$ if $v \in K$, $x_v = 0$ otherwise.

Objective function: minimize $|K|$ minimize $\sum_{v \in V} x_v$

Constraint: $\forall \{u,v\} \in E$, $u \in K$ or $v \in K$ $x_u + x_v \geq 1$

Minimum Vertex Cover (NP-hard)

Let $G = (V, E)$ be a graph

Vertex Cover: set $K \subseteq V$ such that $\forall e \in E$, $e \cap K \neq \emptyset$

set of vertices that "touch" every edge

Solution: $K \subseteq V$ $\Rightarrow$ variables x_v, for each $v \in V$

$x_v = 1$ if $v \in K$, $x_v = 0$ otherwise.

Objective function: minimize $|K|$ minimize $\sum_{v \in V} x_v$

Constraint: $\forall \{u, v\} \in E$, $u \in K$ or $v \in K$ $x_u + x_v \geq 1$

$$
\begin{array}{llcll}
\text{Minimize} & \sum_{v \in V} x_v & & & \\
\text{Subject to:} & x_v + x_u & \geq & 1 & \text{for all } \{u,v\} \in E \\
 & x_v & \in & \{0,1\} & \text{for all } v \in V
\end{array}
$$

Vertex Coloring (NP-hard)

Let $G = (V, E)$ be a graph

k-Proper coloring: $c : V \to \{1, \cdots, k\}$ s.t. $c(u) \neq c(v)$ for all $\{u, v\} \in E$.
color the vertices s ($\leq k$ colors) s.t. adjacent vertices receive $\neq$ colors

Vertex Coloring (NP-hard)

Let $G=(V,E)$ be a graph

k-Proper coloring: $c: V \rightarrow \{1, \cdots, k\}$ s.t. $c(u) \neq c(v)$ for all $\{u,v\} \in E$.
color the vertices s ($\leq k$ colors) s.t. adjacent vertices receive $\neq$ colors

Solution: $c: V \rightarrow \{1, \cdots, n\}$ $\Rightarrow$ variables y_j, is color $j \in \{1, \cdots, n\}$ used?
variable c_v^j for color j and vertex v: $c_v^j = 1$ if v colored j, $c_v^j = 0$ otherwise

Vertex Coloring (NP-hard)

Let $G = (V, E)$ be a graph

k-Proper coloring: $c : V \to \{1, \cdots, k\}$ s.t. $c(u) \neq c(v)$ for all $\{u, v\} \in E$.
color the vertices s ($\leq k$ colors) s.t. adjacent vertices receive $\neq$ colors

Solution: $c : V \to \{1, \cdots, n\}$ $\Rightarrow$ variables y_j, is color $j \in \{1, \cdots, n\}$ used?
variable c_v^j for color j and vertex v: $c_v^j = 1$ if v colored j, $c_v^j = 0$ otherwise
Objective function: minimize $\#$ of used colors minimize $\sum_{1 \leq j \leq n} y_j$

Vertex Coloring (NP-hard)

Let $G=(V,E)$ be a graph

k-Proper coloring: $c : V \rightarrow \{1,\cdots,k\}$ s.t. $c(u) \neq c(v)$ for all $\{u,v\} \in E$.
color the vertices s ($\leq k$ colors) s.t. adjacent vertices receive $\neq$ colors

Solution: $c : V \rightarrow \{1,\cdots,n\}$ $\Rightarrow$ variables y_j, is color $j \in \{1,\cdots,n\}$ used?
variable c_v^j for color j and vertex v: $c_v^j = 1$ if v colored j, $c_v^j = 0$ otherwise

Objective function: minimize $\#$ of used colors minimize $\sum_{1\leq j\leq n} y_j$

Constraints: each vertex v has 1 color $\sum_{1\leq j\leq n} c_v^j = 1$

ends of each edge $\{u,v\} \in E$ have $\neq$ colors: $c_v^j + c_u^j \leq 1$ for all $j \in \{1,\cdots,n\}$

color j used if ≥ 1 vertex colored with j $c_v^j \leq y_j$ for all $v \in V$

Vertex Coloring (NP-hard)

Let $G=(V,E)$ be a graph

k-Proper coloring: $c : V \rightarrow \{1,\cdots,k\}$ s.t. $c(u) \neq c(v)$ for all $\{u,v\} \in E$.
color the vertices s ($\leq$ k colors) s.t. adjacent vertices receive $\neq$ colors

$$
\begin{array}{llcll}
\text{Minimize} & \sum_{1\leq j\leq n} y_j & & & \\
\text{Subject to:} & \sum_{1\leq j\leq n} c_v^j & = & 1 & \text{for all } v \in V \\
 & c_v^j + c_u^j & \leq & 1 & \text{for all } j \in \{1,\cdots,n\}, \{u,v\} \in E \\
 & c_v^j & \leq & y_j & \text{for all } j \in \{1,\cdots,n\}, v \in V \\
 & y_j & \in & \{0,1\} & \text{for all } j \in \{1,\cdots,n\} \\
 & c_v^j & \in & \{0,1\} & \text{for all } j \in \{1,\cdots,n\}, v \in V
\end{array}
$$

Outline

Integer Programme vs. Linear Programme

Integer Linear programme (*ILP*):

$$\begin{array}{rlcll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \in & \mathbb{N} & \forall 1 \leq j \leq n. \end{array}$$

NP-hard in general

Integer Programme vs. Linear Programme

Integer Linear programme (*ILP*):

$$\begin{array}{llll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j \leq & b_i & \forall 1 \leq i \leq m \\ & x_j \in & \mathbb{N} & \forall 1 \leq j \leq n. \end{array}$$

NP-hard in general

Fractional Relaxation: Linear Programme

$$\begin{array}{llll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j \leq & b_i & \forall 1 \leq i \leq m \\ & x_j \geq & 0 & \forall 1 \leq j \leq n. \end{array}$$

Polynomial-time solvable

Integer Programme vs. Linear Programme

Integer Linear programme (*ILP*):

$$\begin{array}{llccl} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \in & \mathbb{N} & \forall 1 \leq j \leq n. \end{array}$$

NP-hard in general

Fractional Relaxation: Linear Programme

$$\begin{array}{llccl} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \geq & 0 & \forall 1 \leq j \leq n. \end{array}$$

Polynomial-time solvable

What is the difference between Optimal solutions of *LP* and of *ILP*?

Integer Programme vs. Linear Programme

Integer Linear programme (*ILP*):

$$\begin{array}{llccl} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \in & \mathbb{N} & \forall 1 \leq j \leq n. \end{array}$$

NP-hard in general

Fractional Relaxation: Linear Programme

$$\begin{array}{llccl} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \geq & 0 & \forall 1 \leq j \leq n. \end{array}$$

Polynomial-time solvable

What is the difference between Optimal solutions of *LP* and of *ILP*?

$$OPT(LP) \geq OPT(ILP) \qquad (\textit{for a maximization problem})$$

$$OPT(LP) \leq OPT(ILP) \qquad (\textit{for a minimization problem})$$

If $OPT(LP)$ is "closed" to $OPT(ILP)$, then solving the Fractional Relaxation (in polynomial-time) gives a good bound for the ILP

Fractional Relaxation of Vertex Coloring

Integer Linear programme (*ILP*):

$$
\begin{array}{llcl}
\text{Minimize} & \sum_{1\leq j\leq n} y_j & & \\
\text{Subject to:} & \sum_{1\leq j\leq n} c_v^j & = & 1 \\
 & c_v^j + c_u^j & \leq & 1 \\
 & c_v^j & \leq & y_j \\
 & y_j & \in & \{0,1\} \\
 & c_v^j & \in & \{0,1\}
\end{array}
$$

Fractional Relaxation (*LP*):

$$
\begin{array}{llcl}
\text{Minimize} & \sum_{1\leq j\leq n} y_j & & \\
\text{Subject to:} & \sum_{1\leq j\leq n} c_v^j & = & 1 \\
 & c_v^j + c_u^j & \leq & 1 \\
 & c_v^j & \leq & y_j \\
 & y_j & \geq & 0 \\
 & c_v^j & \geq & 0
\end{array}
$$

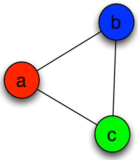
$y_{red} = y_{blue} = y_{green} = 1$
$c_a^{red} = c_b^{blue} = c_c^{green} = 1$
$OPT(ILP) = \sum_c y_c = 3$

$y_{red} = y_{blue} = 1/2$, $y_{green} = 0$
$c_a^{red} = c_b^{red} = c_c^{red} = 1/2$
$c_a^{blue} = c_b^{blue} = c_c^{blue} = 1/2$
$OPT(LP) = \sum_c y_c = 1$

Fractional Relaxation of Vertex Coloring

Integer Linear programme (*ILP*):

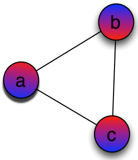
$$
\begin{array}{lrcl}
\text{Minimize} & \sum_{1\leq j\leq n} y_j & & \\
\text{Subject to:} & \sum_{1\leq j\leq n} c_v^j & = & 1 \\
& c_v^j + c_u^j & \leq & 1 \\
& c_v^j & \leq & y_j \\
& y_j & \in & \{0,1\} \\
& c_v^j & \in & \{0,1\}
\end{array}
$$



$y_{red} = y_{blue} = y_{green} = 1$
$c_a^{red} = c_b^{blue} = c_c^{green} = 1$
$OPT(ILP) = \sum_c y_c = 3$

Fractional Relaxation (*LP*):

$$
\begin{array}{lrcl}
\text{Minimize} & \sum_{1\leq j\leq n} y_j & & \\
\text{Subject to:} & \sum_{1\leq j\leq n} c_v^j & = & 1 \\
& c_v^j + c_u^j & \leq & 1 \\
& c_v^j & \leq & y_j \\
& y_j & \geq & 0 \\
& c_v^j & \geq & 0
\end{array}
$$



$y_{red} = y_{blue} = 1/2,\ y_{green} = 0$
$c_a^{red} = c_b^{red} = c_c^{red} = 1/2$
$c_a^{blue} = c_b^{blue} = c_c^{blue} = 1/2$
$OPT(LP) = \sum_c y_c = 1$

Fractional Relaxation of Knapsac

Integer Linear programme (ILP):

$$\begin{array}{rl} \textbf{max} & \sum_{1\leq i\leq n} v_i x_i \\ \textbf{subject to} & \\ & \sum_{1\leq i\leq n} w_i x_i \leq W \\ & x_i \in \{0,1\} \end{array}$$

Fractional Relaxation (LP):

$$\begin{array}{rl} \textbf{max} & \sum_{1\leq i\leq n} v_i x_i \\ \textbf{subject to} & \\ & \sum_{1\leq i\leq n} w_i x_i \leq W \\ & x_i \geq 0 \end{array}$$

Example:

- Sac: $W = n$
- Objects:
  - one object (O_1) of weight $n+0,1$ and value n
  - $n-1$ objects ($O_2, \cdots, O_n$) of weight 1 and value $1/n$

$x_1 = 0, x_2 = \cdots = x_n = 1$

$OPT(ILP) = \sum_c v_i x_i = (n-1)/n$

$x_1 = \frac{n}{n+0,1}, x_2 = \cdots = x_n = 0$

$OPT(LP) = \sum_c v_i x_i = \frac{n^2}{n+0,1}$

$\Rightarrow$ the ratio between the LP optimal solution and the Integral opt. solution may be arbitrary large

Fractional Relaxation of Knapsac

Integer Linear programme (*ILP*):

$$\begin{aligned} \textbf{max} \quad & \sum_{1\leq i\leq n} v_i x_i \\ \textbf{subject to} \quad & \\ & \sum_{1\leq i\leq n} w_i x_i \leq W \\ & x_i \in \{0,1\} \end{aligned}$$

Fractional Relaxation (*LP*):

$$\begin{aligned} \textbf{max} \quad & \sum_{1\leq i\leq n} v_i x_i \\ \textbf{subject to} \quad & \\ & \sum_{1\leq i\leq n} w_i x_i \leq W \\ & x_i \geq 0 \end{aligned}$$

Example:

- Sac: $W = n$
- Objects:
  - one object (O_1) of weight $n+0,1$ and value n
  - $n-1$ objects ($O_2, \cdots, O_n$) of weight 1 and value $1/n$

$x_1 = 0, x_2 = \cdots = x_n = 1$

$OPT(ILP) = \sum_c v_i x_i = (n-1)/n$

$x_1 = \frac{n}{n+0,1}, x_2 = \cdots = x_n = 0$

$OPT(LP) = \sum_c v_i x_i = \frac{n^2}{n+0,1}$

$\Rightarrow$ the ratio between the LP optimal solution and the Integral opt. solution may be arbitrary large

Fractional Relaxation of Knapsac

Integer Linear programme (ILP):

$$\begin{aligned} \textbf{max} \quad & \sum_{1 \le i \le n} v_i x_i \\ \textbf{subject to} \quad & \\ & \sum_{1 \le i \le n} w_i x_i \le W \\ & x_i \in \{0,1\} \end{aligned}$$

Fractional Relaxation (LP):

$$\begin{aligned} \textbf{max} \quad & \sum_{1 \le i \le n} v_i x_i \\ \textbf{subject to} \quad & \\ & \sum_{1 \le i \le n} w_i x_i \le W \\ & x_i \ge 0 \end{aligned}$$

Example:

- Sac: $W = n$
- Objects:
  - one object (O_1) of weight $n + 0,1$ and value n
  - $n-1$ objects ($O_2, \cdots, O_n$) of weight 1 and value $1/n$

$x_1 = 0, x_2 = \cdots = x_n = 1$
$OPT(ILP) = \sum_c v_i x_i = (n-1)/n$

$x_1 = \frac{n}{n+0,1}, x_2 = \cdots = x_n = 0$
$OPT(LP) = \sum_c v_i x_i = \frac{n^2}{n+0,1}$

$\Rightarrow$ the ratio between the LP optimal solution and the Integral opt. solution may be arbitrary large

Outline

No integrality gap

Integer Linear programme (*ILP*):

$$\begin{array}{llcll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \in & \mathbb{N} & \forall 1 \leq j \leq n. \end{array}$$

NP-hard in general

Fractional Relaxation: Linear Programme

$$\begin{array}{llcll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \geq & 0 & \forall 1 \leq j \leq n. \end{array}$$

Polynomial-time solvable

In some cases: $OPT(ILP) = OPT(LP)$.

$\Leftrightarrow$ there exists an integral solution with value $OPT(LP)$.
In such case, Polynomial-time solvable: solve the Fractional Relaxation

Integer Programme Example: Shortest path

$D = (V, A)$ be a digraph with length $\ell : A \to \mathbb{R}^+$, and $s, t \in V$.
Problem: Compute a shortest directed path from s to t.

Integer Programme Example: Shortest path

$D = (V, A)$ be a digraph with length $\ell : A \to \mathbb{R}^+$, and $s, t \in V$.
Problem: Compute a shortest directed path from s to t.

Solution: A path P from s to t $\Rightarrow$ variables x_a for each $a \in A$

$x_a = 1$ if $a \in A(P)$, $x_a = 0$ otherwise.

Integer Programme Example: Shortest path

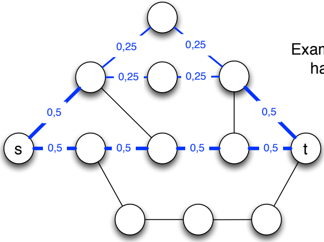
$D=(V,A)$ be a digraph with length $\ell : A \to \mathbb{R}^+$, and $s,t \in V$.
Problem: Compute a shortest directed path from s to t.

Solution: A path P from s to t $\Rightarrow$ variables x_a for each $a \in A$
$x_a = 1$ if $a \in A(P)$, $x_a = 0$ otherwise.

$$\begin{array}{llcll}
\text{Minimize} & \sum_{a\in A} \ell(a)x_a & & & \\
\text{Subject to:} & \sum_{u\in N^+(s)} x(su) & = & 1 & \\
 & \sum_{u\in N^-(t)} x(tu) & = & 1 & \\
 & \sum_{u\in N^+(v)} x(uv) & = & \sum_{u\in N^-(v)} x(vu) & \text{for all } v \in V \setminus \{s,t\} \\
 & x(a) & \in & \{0,1\} & \text{for all } a \in A
\end{array}$$

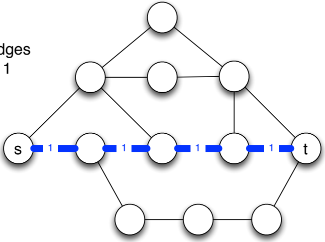
Integer Programme Example: Shortest path

$$
\begin{array}{llcll}
\text{Minimize} & \sum_{a\in A} \ell(a)x_a & & & \\
\text{Subject to:} & \sum_{u\in N^+(s)} x(su) & = & 1 & \\
 & \sum_{u\in N^-(t)} x(tu) & = & 1 & \\
 & \sum_{u\in N^+(v)} x(uv) & = & \sum_{u\in N^-(v)} x(vu) & \text{for all } v\in V\setminus\{s,t\} \\
 & x(a) & \color{red}{\geq} & \color{red}{0} & \text{for all } a\in A
\end{array}
$$



a fractional solution
OPT(LP)=4

Example: all edges have length 1



an integral solution
OPT(ILP)=4

Exercise: Prove that this LP always admits an integral optimal solution

Integer Programme Example: Maximum Matching

$G = (V, E)$ be a graph
Problem: Compute a maximum matching

Solution: a set $M \subseteq E$ of pairwise disjoint edges

$\Rightarrow$ variables x_e for each $e \in E$

$x_e = 1$ if $e \in M$, $x_e = 0$ otherwise.

$$
\begin{array}{llcll}
\text{Maximize} & \sum_{e \in E} x_e & & & \\
\text{Subject to:} & \sum_{e \in E, v \in e} x_e & \leq & 1 & \text{for all } v \in V \\
 & x_e & \in & \{0,1\} & \text{for all } e \in E
\end{array}
$$

Exercise: Prove that the fractional relaxation of this ILP always admits an integral optimal solution

Totally unimodular matrices

unimodular matrix: square matrix with determinant $+1$ or -1

totally unimodular matrix: every square non-singular submatrix is unimodular

Integer Linear programme (*ILP*):

$$\begin{array}{llcll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \in & \mathbb{N} & \forall 1 \leq j \leq n. \end{array}$$

NP-hard in general

Fractional Relaxation: Linear Programme

$$\begin{array}{llcll} \text{Max.} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{s.t.:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \forall 1 \leq i \leq m \\ & x_j & \geq & 0 & \forall 1 \leq j \leq n. \end{array}$$

Polynomial-time solvable

Theorem [Hoffman,Kruskal, 1956]

If the matrix $A = [a_{ij}]$ is totally unimodular then every *basic* feasible solution (the “corner" of the polytope) is integral

$\Rightarrow$ exist integral optimal solution of the LP

$\Rightarrow$ $OPT(ILP)$ can be computed by solving the Fractional relaxation

Outline

Integer Programme Example: Minimum Spanning Tree

$G = (V, E)$ be a graph with weight $w : E \to \mathbb{R}^+$, and $s, t \in V$.
Problem: Compute a minimum spanning tree

Solution: A spanning tree T $\Rightarrow$ variables x_e for each $e \in E$
$x_E = 1$ if $e \in E(T)$, $x_e = 0$ otherwise.

$$
\begin{array}{llcll}
\text{Minimize} & \displaystyle\sum_{e \in E} w(e) x_e & & & \\
\text{Subject to:} & \displaystyle\sum_{e=\{u,v\} \in E, u \in S, v \notin S} x_e & \geq & 1 & \text{for all } S \subseteq V \\
 & x_e & \in & \{0,1\} & \text{for all } e \in E
\end{array}
$$

Remark: The number of constraints is exponential

Optical Networks (WDM)

Optical network: optical fiber connecting e.g. routers

Wavelength-division Multiplexing (WDM): technology which multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths (i.e., colors) of laser light [Wikipedia]

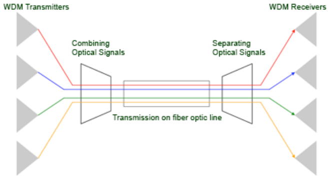
$\Rightarrow$: different signals on the same link must have different wavelengths (colors)

Optical Networks (WDM)

Optical network: optical fiber connecting e.g. routers

Wavelength-division Multiplexing (WDM): technology which multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths (i.e., colors) of laser light [Wikipedia]

⇒: different signals on the same link must have different wavelengths (colors)



Optical Networks (WDM) RWA problem

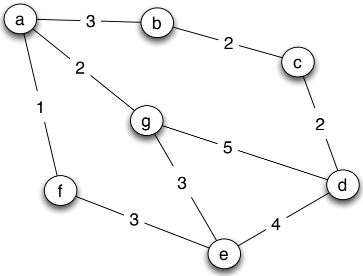
RWA: Routing and Wavelentgh Assignment Problem

Given a graph $G = (V, E)$ with capacity on links, and a traffic-demand matrix T, where $T[u, v]$ is the amount of traffic that must transit from u to v, for any $u, v \in V$.
Find a set of paths and one wavelength assignment for each path such that:

- all demands are routed
- capacity of each link cannot be exceeded
- total number of wavelength is as small as possible

demand matrix:

	a	b	c	d	e	f	g
a	0	0	**1**	**1**	0	**1**	0
b	0	0	0	0	0	**2**	0
c	0	0	0	0	0	0	0
d	0	0	0	0	0	0	**5**
e	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0
g	0	**1**	0	0	0	0	0



Optical Networks (WDM) RWA problem

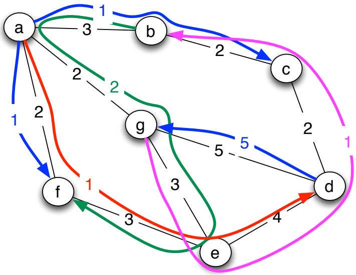
RWA: Routing and Wavelentgh Assignment Problem

Given a graph $G = (V, E)$ with capacity on links, and a traffic-demand matrix T, where $T[u, v]$ is the amount of traffic that must transit from u to v, for any $u, v \in V$.
Find a set of paths and one wavelength assignment for each path such that:

- all demands are routed
- capacity of each link cannot be exceeded
- total number of wavelength is as small as possible

demand matrix:

	a	b	c	d	e	f	g
a	0	0	**1**	**1**	0	**1**	0
b	0	0	0	0	0	**2**	0
c	0	0	0	0	0	0	0
d	0	0	0	0	0	0	**5**
e	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0
g	0	**1**	0	0	0	0	0

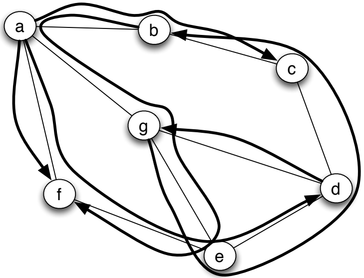


Optical Networks (WDM) RWA problem

Let us simplify the problem $\Rightarrow$ consider only Wavelength assignment

WA: Wavelentgh Assignment Problem

Given a graph $G = (V, E)$ with capacity on links, and a set of paths
Give One color to each path s.t. no two paths with the same color cross a same link
Minimize the number of colors

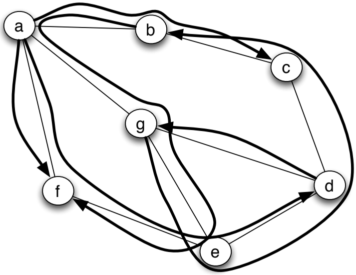


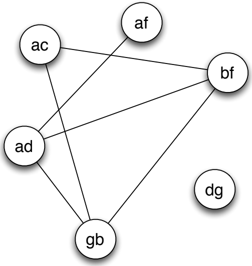
Optical Networks (WDM) RWA problem

Let us simplify the problem ⇒ consider only Wavelength assignment

WA: Wavelentgh Assignment Problem

Given a graph $G = (V, E)$ with capacity on links, and a set of paths
Give One color to each path s.t. no two paths with the same color cross a same link
Minimize the number of colors





It is the problem of PROPER COLORING in graphs !!

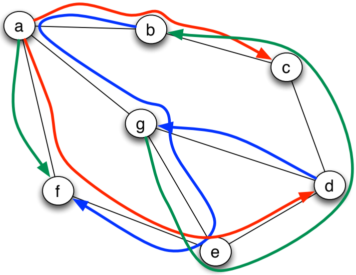
the “simplified" problem is already NP-complete :(

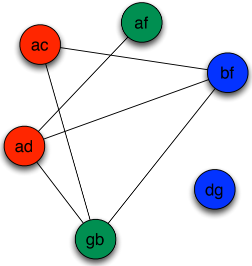
Optical Networks (WDM) RWA problem

Let us simplify the problem ⇒ consider only Wavelength assignment

WA: Wavelentgh Assignment Problem

Given a graph $G = (V, E)$ with capacity on links, and a set of paths
Give One color to each path s.t. no two paths with the same color cross a same link
Minimize the number of colors





It is the problem of PROPER COLORING in graphs !!

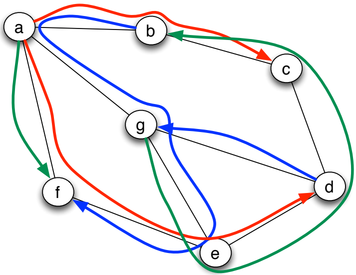
the “simplified" problem is already NP-complete :(

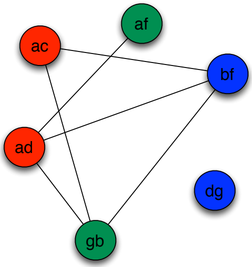
Optical Networks (WDM) RWA problem

Let us simplify the problem ⇒ consider only Wavelength assignment

WA: Wavelentgh Assignment Problem

Given a graph $G = (V, E)$ with capacity on links, and a set of paths
Give One color to each path s.t. no two paths with the same color cross a same link
Minimize the number of colors





Exercise: Write a (Integer) Linear Programme that solves the RWA problem

Summary: To be remembered

- ILP allow to model many problems
- there may be a huge integrality gap
  (between $OPT(LP)$ and $OPT(ILP)$).
- if no integrality gap (e.g., totally unimodular matrices)
  $\Rightarrow$ Fractional Relaxation gives Optimal Integral Solution