Mathematical Optimization (linear programming)

$x\in\mathbb{R}^n$; $c\in\mathbb{R}^n$; $A\in\mathbb{R}^{n\times m}$; $b\in\mathbb{R}^m$

- **canonical form LP:** max c^Tx subj. to: $Ax\le b$ and $x\ge 0$

[general LP can be reconstructed with:

neg. x constr.: $x_i\le 0$ → substit x_i with $x_i=-x_i'$

no x constr.: $x_i\in\mathbb{R}$ → substit x_i with $x_i=x_i^+-x_i^-$

minimization: $\min c^Tx$ → $\max -c^Tx$

lower bound: $A_ix\ge b_i$ → $-A_ix\le -b_i$

equality: $A_ix=b_i$ → $A_ix\ge b_i$; $-A_ix\ge -b_i$]

- **standard form LP:** max c^Tx subj. to: $Ax=b$ and $x\ge 0$

canonical LP from standard LP: $A_ix\le b_i$ → $y_i+A_ix=b_i$; $y_i\ge 0$ y_i: slack variables appended to x

example:
$$\begin{aligned} x_1+4x_2&\le 40\\ 2x_1+x_2&\le 42\\ 1.5x_1+3x_2&\le 36\end{aligned} \rightarrow \begin{array}{ccccc|c} y_1 & y_2 & y_3 & x_1 & x_2 & \\ 1&0&0&1&4&40\\ 0&1&0&2&1&42\\ 0&0&1&\underbrace{1.5}_{A}&3&\underbrace{36}_{b}\end{array}$$

$B=(1,2,3)$

allowed operations (keep same solution)
- change row order
- multiply row with non-zero number
- add multiple of one row to an other

- **basis:** $B=(\beta(1),\dots,\beta(m))$; $\beta(i)\in\{1,\dots,n+m\}$, $i=1,\dots,m$ ($\beta(i)$: column # of A where column $=e_i$)

- **basis exchange:** modify A,b to change basis without changing solution

$$\begin{array}{ccccc|c} 1&0&0&1&4&40\\ 0&1&0&2&1&42\\ 0&0&1&1.5&3&36\end{array} \rightarrow \begin{aligned} ①&=1/4\,①\\ ②&=②-1/4\,①\\ ③&=③-3/4\,①\end{aligned} \rightarrow \begin{array}{ccccc|c} 1/4&0&0&1/4&1&10\\ -1/4&1&0&7/4&0&32\\ -3/4&0&1&3/4&0&6\end{array}$$

↳ $B=(1,2,3)$ → exchange $i=1,k=5$ → ↳ $B=(5,2,3)$

- **basic solution:** set non-basic var. to 0

$$\begin{array}{ccccc|c} 1/4&0&0&1/4&1&10\\ -1/4&1&0&7/4&0&32\\ -3/4&0&1&3/4&0&6\end{array} \rightarrow \begin{aligned} x_2&=10-1/4\,y_1-1/4\,x_1\\ y_2&=32+1/4\,y_1-7/4\,x_1\\ y_3&=6+3/4\,y_1-3/4\,x_1\end{aligned} \rightarrow \begin{gathered} y_1^*=0,\ x_1^*=0\\ x_2^*=10,\ y_2^*=32,\ y_3^*=6\end{gathered}$$

- **basic feasible solution:** basic solution that is also feasible ↔ all $x^*,y^*\ge 0$

- **short tableau:** (= compact notation)

$$\begin{array}{ccccc|c} i=\rightarrow & 2 & 3 & & 1 & \\ k=\rightarrow 1 & 2 & 3 & 4 & 5 & \\ y_1 & y_2 & y_3 & x_1 & x_2 & \\ 1/4&0&0&1/4&1&10\\ -1/4&1&0&7/4&0&32\\ -3/4&0&1&3/4&0&6\end{array} \rightarrow \begin{array}{ccccc|c} i= & k= & \longrightarrow & 1 & 4 & \\ \downarrow & \downarrow & & y_1 & x_1 & \\ 1 & 5 & x_2 & 1/4 & 1/4 & 10\\ 2 & 2 & y_2 & -1/4 & 7/4 & 32\\ 3 & 3 & y_3 & -3/4 & 3/4 & 6\end{array}$$

(basic solution x_2^*,y_2^*,y_3^* for $x_1^*=0,y_1^*=0$)

↳ basis exchange formula:

$a'_{ik}=1/a_{ik}$ (pivot)

$a'_{jk}=-a_{jk}/a_{ik}$ (pivot col.)

$a'_{il}=a_{il}/a_{ik}$ (pivot row)

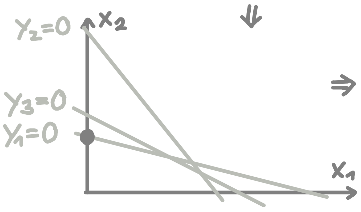
$a'_{jl}=a_{jl}-a_{jk}a_{il}/a_{ik}$

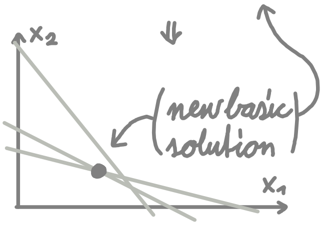
$b'_i=b_i/a_{ik}$

$b'_j=b_j-a_{jk}b_i/a_{ik}$

exchange $i=3,k=4$:

$$\begin{array}{ccc|c} & y_1 & x_1 & \\ x_2 & 1/4 & 1/4 & 10\\ y_2 & -1/4 & 7/4 & 32\\ y_3 & -3/4 & \boxed{3/4} & 6\end{array} \rightarrow \begin{array}{ccc|c} & y_1 & y_3 & \\ x_2 & 1/2 & -1/3 & 8\\ y_2 & 6/4 & -7/3 & 18\\ x_1 & -1 & \boxed{4/3} & 8\end{array}$$





cond. for feasible basic solution to stay feasible
• $a_{ik}>0$ (pivot nonnegative)
• $b_j/a_{jk}\ge b_i/a_{ik}\ \forall j$ (quotient rule)

- **short tableau with cost:** append variable z to x to track objective of basic solution ($z=c^Tx$)

$$\begin{array}{ccccccc|c} z & y_1 & \cdots & y_m & x_1 & \cdots & x_n & \\ 1 & 0 & \cdots & 0 & -c_1 & \cdots & -c_2 & 0\\ 0 & & \mathbb{I} & & & A & & b\end{array} \rightarrow \begin{array}{cccc|c} & x_1 & \cdots & x_n & \\ z & -c_1 & \cdots & -c_n & 0\\ y_1 & a_{11} & \cdots & a_{1n} & b_1\\ \vdots & \vdots & & \vdots & \vdots\\ y_m & a_{m1} & \cdots & a_{mn} & b_m\end{array}$$

cond. for increasing objective after exchange: $a_{0k}<0$

(basis exchange formula same for objective row as for other rows.)

simplex method

1) start with a feasible tableau $\Leftrightarrow b_1,..,b_m \geq 0$
2) perform exchange step with a pivot a_{ik} that satisfies:
   - $a_{ik}>0$ and $b_j/a_{jk} \geq b_i/a_{ik}\ \forall j$ → will remain feasible
   - $a_{0k}<0$ → will increase objective
3) repeat ② until:
   - $a_{0k}>0\ \forall k$ → basic solution is optimal solution!
   - $\exists k$ s.t. $a_{0k}<0$ & $a_{ik}\leq 0\ \forall i$ → solution is unbounded!

multiple pivots could satisfy these cond.
additional rules that improve efficiency:
- pick column with most negative a_{0k}
- pick viable column with smallest index k
  ↳ "Bland's rule": guaranteed to terminate

- **finding a feasible basic solution** (for when initial tableau is unfeasible)

original problem: $\max c^T x$ subj. to: $y+Ax=b$ and $x\geq 0, y\geq 0$
auxilary problem: $\max -x_0$ subj. to: $y+Ax-(1,..,1)^T\cdot x_0=b$ and $x\geq 0, y\geq 0, x_0\geq 0$

↳ problem always feasible (with large x_0)! if $x_0^*=0$, then x^*, y^* feasible solutions of original problem!

$$\begin{array}{c|ccccc|c} & x_1 & \cdots & x_n & x_0 & & \\ \hline z & 0 & \cdots & 0 & 1 & & 0 \\ y_1 & a_{11} & \cdots & a_{1n} & -1 & & b_1 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ y_m & a_{m1} & \cdots & a_{mn} & -1 & & b_m \end{array}$$

⇒ pivot on column x_0 and row with most negative b_i
↳ guaranteed feasible sol.

⇒ use above simplex method to find x_0^*, x^*, y^*
- $x_0^*>0$: original problem is infeasible
- $x_0^*=0$: use x^*, y^* as feasible starting point to solve original problem!

dual problem

primal problem: $\max c^T x$ subj. to: $Ax\leq b,\ x\geq 0$ (canonical form!) → dual problem: $\min b^T y$ subj. to $A^T y\geq c,\ y\geq 0$ $=$ ($\max -b^T y$ subj. to: $-A^T y\leq -c, y\geq 0$)

$\max\ 4x_1+x_2+5x_3+3x_4$
s.t.: $x_1-x_2-x_3+3x_4\leq 1$
$5x_1+x_2+3x_3+8x_4\leq 55$
$-x_1+2x_2+3x_3-5x_4\leq 3$
$x_1,x_2,x_3,x_4\geq 0$

→ find linear comb. of constr. s.t. every x coeff. dominates its coeff. in c^T and resulting contraint is minimal

$$\begin{array}{l} 4x_1+x_2+5x_3+3x_4=c^T x \\ \quad\wedge\quad\ \wedge\quad\ \wedge\quad\ \wedge\qquad\ \wedge \\ (\boxed{x_1}\boxed{-x_2}\boxed{-x_3}\boxed{+3x_4}\leq\boxed{1})\cdot y_1 \\ (\boxed{5x_1}\boxed{+x_2}\boxed{+3x_3}\boxed{+8x_4}\leq\boxed{55})\cdot y_2 \\ (\boxed{-x_1}\boxed{+2x_2}\boxed{+3x_3}\boxed{-5x_4}\leq\boxed{3})\cdot y_3 \end{array}$$

→ $\min\ y_1+55y_2+3y_3$
s.t.: $y_1+5y_2-y_3\geq 4$
$-y_1+y_2+2y_3\geq 1$
$-y_1+3y_2+3y_3\geq 5$
$-3y_1+8y_2-5y_3\geq 3$
$y_1,y_2,y_3\geq 0$

- **rel. to standard LP:** primal slack var. $x^s \hat{=} y$ dual optim. var.
  dual slack var. $y^s \hat{=} x$ primal optim. var.
  ($\max c^T x$ subj. to: $x^s+Ax=b$; $x^s, x\geq 0$
  $\max -b^T y$ subj. to: $y^s-A^T y=-c$; $y^s, y\geq 0$)

- **weak duality theorem:** $z=c^T x\leq b^T y=w$ feasible x objective ≤ feasible y objective
- **strong duality theorem:** $z^*=c^T x^*=b^T y^*=w^*$ optimum of primal = optimum of dual

(usefull e.g. if some non-optimal z,w are found that are close → can stop, since optimum is between)

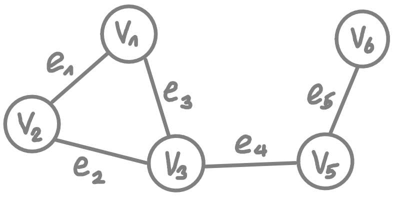
p \ d	OPT	INF	UNB
OPT	✓	✗	✗
INF	✗	✓	✓
UNB	✗	✓	✗

- **complementary slackness theorem:** necessary and sufficient cond. for x^* and y^*
  $((A)_i x^*=b_i$ OR $y_i^*=0\ \ \forall i=1,..,m)$ AND $((A^T)_j y^*=c_j$ OR $x_j^*=0\ \ \forall j=1,..,n)$
  (usefull to compute x^* if y^* is known and the opposite)

- **sensitivity theorem:** for relaxed constr. $Ax\leq b+t$ (t: small!) → $z^*_{new}=z^*+t^T\cdot y^*$
  ↳ if $y_i=0$: constr. i is inactive → relaxing it with $t_i>0$ won't raise objective
  ↳ if $y_i>0$: relaxing constr. i with $t_i>0$ will raise objective. larger y_i → larger increase in z^*

combinatorial optimization

• undirected, unweighted graph:



vertices: $V=\{v_1 \dots v_n\}$
edges: $E=\{e_1 \dots e_m\}$
graph: $G=(V,E)$

edge representation in memory:

-adjacency matrix

	V_1	V_2	V_3	V_4	V_5
V_1	•	1	1	•	•
V_2	1	•	1	•	•
V_3	1	1	•	•	1
V_4	•	•	•	•	1
V_5	•	•	1	1	•

-incidence list:

V_1: e_1 e_3
V_2: e_1 e_2
V_3: e_2 e_3 e_4
V_4: e_5
V_5: e_4 e_5

- Breadth-First search (for shortest path): find smallest # of edges connecting v_1 to $u \in V := d(v_1,u)$

$\bar{d}(v_1,u)=\begin{cases}0: & u=v_1\\ \infty: & u\neq v_1\end{cases}$; $k=1$; $L=\{v_1\}$

while $L\neq\{\}$ do:
  $L_{new}=\{\}$
  for all $u\in N(L)=\{u\in V \mid \exists w\in L, \{u,w\}\in E\}$ do:
    if $\bar{d}(v_1,u)=\infty$ then $\bar{d}(v_1,u)=k$, $L_{new}\leftarrow u$
  $k\leftarrow k+1$, $L\leftarrow L_{new}$

proof: show $D_k=\bar{D}_k$ by induction $\left(\begin{array}{l}D_k=\{u\in V \mid d(v_1,u)=k\}\\ \bar{D}_k=\{u\in V \mid \bar{d}(v_1,u)=k\}\end{array}\right)$

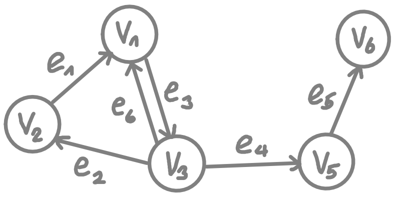
$D_0=\{v_1\}=\bar{D}_0$ ✓

assume $D_r=\bar{D}_r\ \forall r\leq k$, prove $D_{k+1}=\bar{D}_{k+1}$:

$\bar{D}_{k+1}\subseteq D_{k+1}$: let $u\in\bar{D}_{k+1}$. since $u\in N(D_k)\rightarrow d(v_1,u)\leq k+1$ and with $d(v_1,u)>k\rightarrow d(v_1,u)=k+1\rightarrow\bar{D}_{k+1}\subseteq D_{k+1}$

$D_{k+1}\subseteq\bar{D}_{k+1}$: let $u\in D_{k+1}\rightarrow\exists$ path $(v_1,\dots,w,u)\rightarrow w\in D_k$ $\rightarrow u\in N(D_k)\rightarrow u\in\bar{D}_{k+1}\rightarrow D_{k+1}\subseteq\bar{D}_{k+1}$ ■

• directed, positive weighted graph:



graph: $G=(V,E)$
vertices: $V=\{v_1 \dots v_n\}$
edges: $E=\{e_1 \dots e_m\}$
weights: $l: E\rightarrow\mathbb{Z}_{0+}$

path: $P=\{e_a,e_b,\dots\}$ path lenght: $l(P)=\sum_{e\in P}l(e)$

shortest path: $d(s,v)=\min_P l(P)$ s.t P is s-v path

comments:
- if path $l(s\rightsquigarrow w\rightsquigarrow t)=d(s,t)$ then $l(s\rightsquigarrow w)=d(s,w)$
- all edges used by shortest paths form a tree

- Dijkstras algorithm (for shortest path):

$M=\{\}$, $d_v=\infty\ \forall v\in V/\{s\}$, $d_s=0$

while $M\neq V$:
  $u=\underset{v\in V\setminus M}{\operatorname{argmin}}\ d_v$
  $M\leftarrow M\cup\{u\}$ outgoing edges
  for all $e=(u,w)\in\delta_+(u)$, $w\in V\setminus M$:
    if $d_w>d_u+l(e)$ then $d_w\leftarrow d_u+l(e)$

proof: show $d_v=d(s,v)\ \forall v\in V$ by induction

at iter 1: $d_s=d(s,s)=0$ ✓

assume at iter i: $d_x=d(s,x)\ \forall x\in M$ → show at i+1: $d_u=d(s,u)$

$d_u<d(s,u)$: can't happen, as d_u always <u>some</u> path or ∞

$d_u>d(s,u)$:

$d_y=l(P_y)\leq l(P)=d(s,u)<d_u$ ↯

■

- longest path problem: can be built by filipping all weights to negative and searching shortest path.

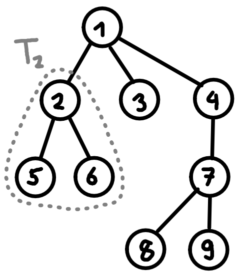
problem is that subpath of longest path is itself not longest. Dijkstras alg. breaks... very hard problem!

special case for directed acyclic graphs DAG:

vertices sorted, so all edges point right.

in this case a subpath of the longest path is also the longest subpath!

- **trees**: any 2 vertices are connected by exactly one path.
  - **independent set problem**: largest set of vertices that don't share edges $S \subseteq V$ s.t. $\forall u,v \in S: \{u,v\} \notin E$



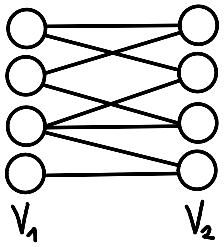
T: entire tree with root at v_1
T_i: subtree with root at v_i
X_i: size of max. indep. set in T_i
Y_i: " , without using v_i
Z_i: " , with using v_i

comments:
- if $v_i \notin S$ then $|S \cap V[T_j]| = X_j \quad \forall v_j$ children of v_i
- if $v_i \in S$ then $|S \cap V[T_j]| = Y_j \quad \forall v_j$ children of v_i

algorithm: traverse tree bottom up, updating Y_i, Z_i for each node:
$$Y_i = \sum_j \max(Y_j, Z_j) \; ; \; Z_i = 1 + \sum_j Y_j \quad \text{with } j \text{ children of } i \quad \rightarrow \quad |S| = X_1 = \max(Y_1, Z_1)$$

combinatorial optimization via LP

- **perfect bipartite matching problem**: find largest set $M \subseteq E$ s.t. $\forall e_1 \neq e_2,\ e_1, e_2 \in M: \ e_1 \cap e_2 = \emptyset$



$G = (V_1 \cup V_2, E)$
$|V_1| = |V_2| = n$

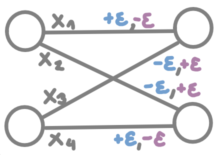
variable: $\forall e \in E \quad x_e \in \{0,1\}$ (1 for pick e for matching M)
constraint: $\forall v \in V \quad \sum_{e \in \delta(v)} x_e = 1$ (perfect matching: one edge out of every vertex)
objective: $\max \sum_{e \in E} x_e \cdot c_e$ (c_e: cost of edge)

$\hookrightarrow Z^*_{IP} = \max \sum_{e \in E} x_e \cdot c_e$ subj. to: $\sum_{e \in \delta(v)} x_e = 1 \ \forall v \in V \ ; \ x_e \geq 0 \ \forall e \in E \ ; \ x_e \in \mathbb{Z}$

relaxed: $Z^*_{LP} = \max \sum_{e \in E} x_e \cdot c_e$ subj. to: $\sum_{e \in \delta(v)} x_e = 1 \ \forall v \in V \ ; \ x_e \geq 0 \ \forall e \in E \ ; \ x_e \in \mathbb{R}$

in this problem if x^* is any extreme point (basic feasible solution) of the relaxed problem, then $x^* \in \mathbb{Z}$.
since LP solutions are always extreme points it follows that $Z^*_{IP} = Z^*_{LP}$!

proof by contradiction: assume x extreme point and fractional



for x to be fractional and feasible it has to contain loops of even lenght. two new feasible solutions y, z can be built by adding/subtracting a sufficiently small ε to edges of x as shown.
(i.e. $0 < x_e < 1 \ \forall e$ for fractional x) $\rightarrow x = \frac{1}{2}y + \frac{1}{2}z \rightarrow x$ is not an extreme point!

- **maximum matching / minimum vertex cover**:

max. matching: $z^*_{IP} = \max \sum_{e \in E} x_e$ subj. to: $\sum_{e \in \delta(v)} x_e \leq 1 \ \forall v \in V, \ x_e \geq 0 \ \forall e \in E, \ x_e \in \mathbb{Z}$
relaxed max. matching: $z^*_{LP} = \max \sum_{e \in E} x_e$ subj. to: $\sum_{e \in \delta(v)} x_e \leq 1 \ \forall v \in V, \ x_e \geq 0 \ \forall e \in E, \ x_e \in \mathbb{R}$
relaxed min vertex cover: $w^*_{LP} = \min \sum_{v \in V} y_v$ subj. to: $y_v + y_u \geq 1 \ \forall \{u,v\} \in E, \ y_v \geq 0 \ \forall v \in V, \ y_v \in \mathbb{R}$
min vertex cover: $w^*_{IP} = \min \sum_{v \in V} y_v$ subj. to: $y_v + y_u \geq 1 \ \forall \{u,v\} \in E, \ y_v \geq 0 \ \forall v \in V, \ y_v \in \mathbb{Z}$

dual!

$\hookrightarrow z^*_{IP} \leq z^*_{LP} = w^*_{LP} \leq w^*_{IP} \rightarrow$ weak duality holds for max. matching $\Leftrightarrow$ min. vertex cover!

for bipartite graps these can be shown to be equalities (similar to above proof)