Optimalisering/Besliskunde 1

College 1

6 september, 2012

Algemene informatie

- College: donderdag 9:00-10:45: Gorlaeus C1/C2, Leiden
  vrijdag: werkcollege Leiden en Delft
  vragenuur Delft

- Vier *verplichte* huiswerkopgaven

- Informatie en documenten worden op **mijn homepage** bijgehouden.
  http://ta.twi.tudelft.nl/wst/users/aardal/
  MAAR, schrijf je toch in op Blackboard.

Docenten

Karen Aardal:
(do: Leiden, vr: Delft)

Herman Blok:
(Leiden, werkcollege/ huiswerk)

Floske Spieksma:
(Leiden)

Pieter van den Berg:
(Delft, spreekuur/ huiswerk)

Keuze inhoud

	Delft		Leiden	
	Optimalisering	=	Besliskunde 1	BSc
			Besliskunde 2	
NEW!	Combinatorisce opt.	≈	Besliskunde 3	

<u>MasterMath vakken optimalisering:</u>

Discrete Optimization

Continuous Optimization

Advanced Linear Programming

Heuristic Methods in Operations Research

MSc

<u>Landelijk Netwerk Mathematische Besliskunde</u> (LNMB)

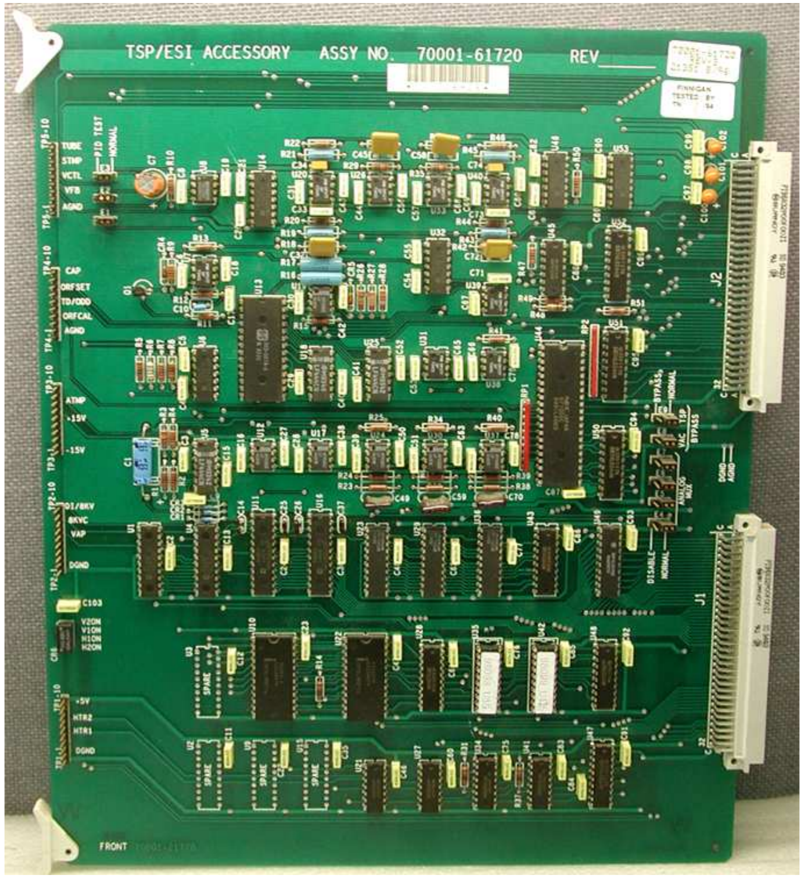
PhD

Wat is optimalisering?

Bepaal “de beste” oplossing uit een “gegeven” verzameling oplossingen.

Wat is hier moeilijk aan????

Voorbeeld:



Je moet gaten boren voor componenten en bedrading van een “printed circuit board” (pcb). De boor-machine moet *zo snel mogelijk* klaar zijn.

Hoe kunnen we dit probleem wiskundig modelleren en formuleren?

Een model gebaseerd op een graaf *G*:

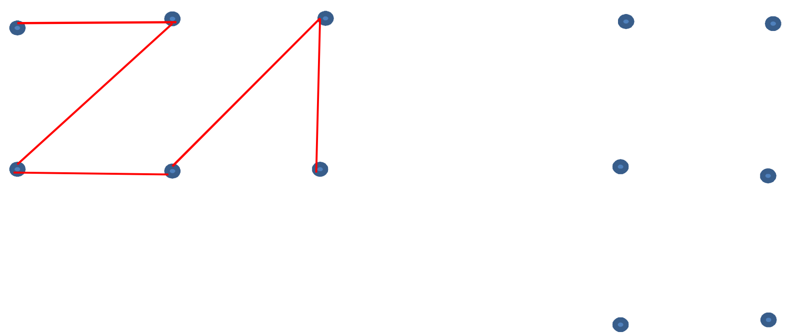
Een graaf bestaat uit “punten” en “kanten” (verbindingen)

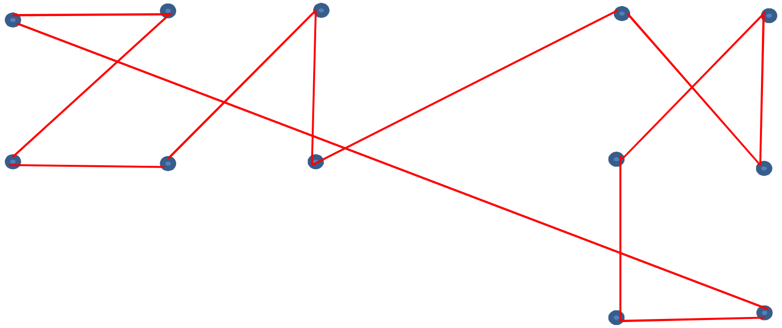
De punten representeren de gaten in het pcb.

Alle verbindingen tussen punten zijn mogelijk. Als een verbinding wordt gebruikt betekent dit dat de twee verbonden punten (=gaten) achter elkaar worden geboord.

Hoe ziet er een “toegelaten” verzameling verbindingen uit?

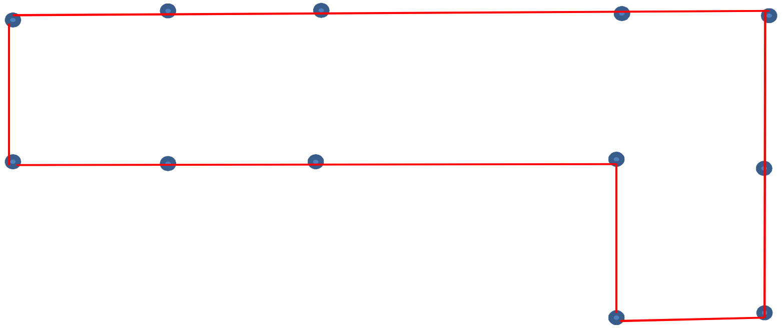
Ze vormen een route door de gaten. Elk gat wordt precies een keer geboord.





Goede route?

Niet echt…



Deze route is veel korter? Maar is het de beste?

Hoeveel routes zijn het er in een ongerichte graaf met *n* punten?

$$\frac{1}{2} \cdot (n-1) \cdot (n-2) \cdot \cdots = \tfrac{1}{2} \cdot (n-1)!$$

Voor ons voorbeeldje: 19.958.400 routes

Die zijn makkelijk te bepalen en te vergelijken. Maar dit was maar een klein voorbeeldje...

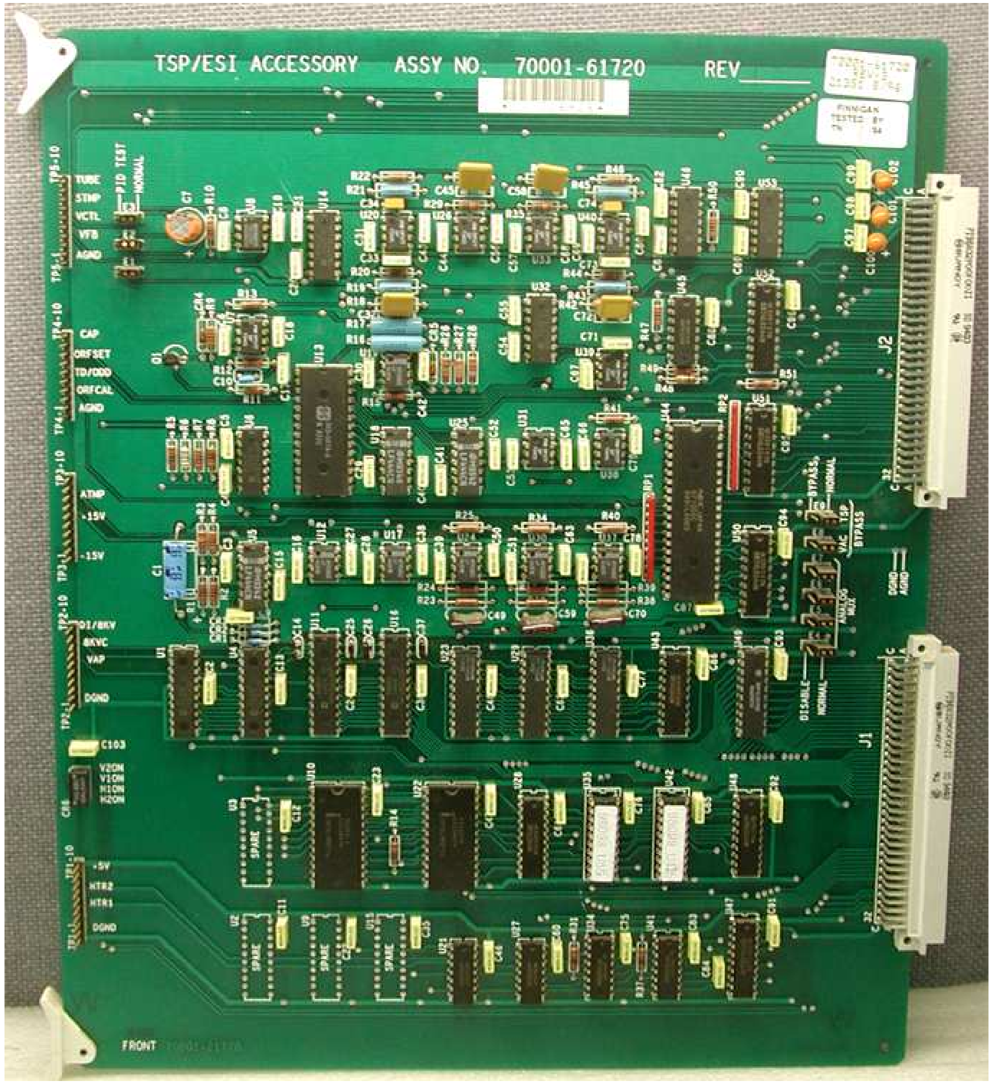
100 gaten:

39432893368239525177618160696609253114
75679888435866316473712666221797249817
01671460152142005992311952088606069459
81941512882139512131855253096331247641
49655567314286353816586186984944719612
22810725832120127016645932065613714147
42663876212120378695162016062870278978
43301130159520851620311758504293980894
61111394811851948687360000000000000000
000000000000000000000000000000000

routes

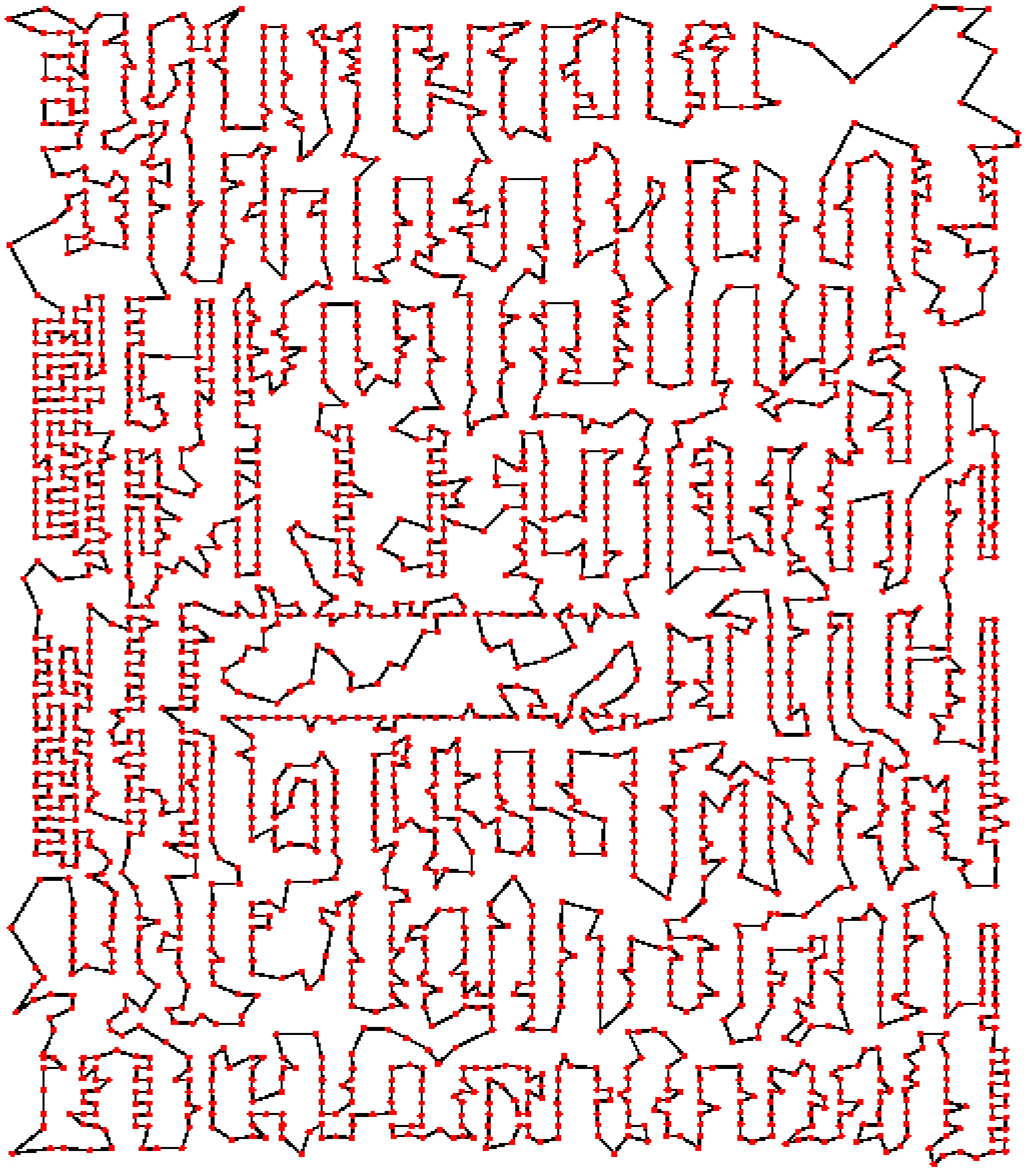
Ouch, wij moeten iets slimmers bedenken

Terug naar ons echte voorbeeld:



3038 gaten

Beste (=optimale) route:



Hoe pakken we dit aan?

Formuleer een optimaliseringsprobleem gebaseerd op het grafen-model:

Input (wat weten wij?): De graaf $G = (V, E)$ en de lengte l_e van elke verbinding (kant) $e \in E$

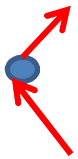
Stap 1: Definieer de beslissingsvariabelen
(wat willen wij eigenlijk weten?)

$$\begin{aligned} x_e &= 1 \text{ als kant } e \in E \text{ in de route zit} \\ x_e &= 0 \text{ anders} \end{aligned}$$

Stap 2: Formuleer de doelfunctie
(wat willen wij maximaliseren of minimaliseren?)
Hier: minimaliseer de lengte van de route

$$\min \sum_{e \in E} l_e x_e$$

Stap 3: Formuleer de voorwaarden
(Als we niks doen krijgen alle variabelen waarde 0)



(i) De boormachine gaat naar een te boren gat en gaat na het boren weer weg. Dat betekent dat elk punt precies twee verbindingen moet hebben

$$\sum_{e\in E \text{ die bij punt } v \text{ aansluiten}} x_e = 2 \text{ voor iedere } v \in V$$

(ii) Wij willen geen “sub-routes”
(iets ingewikkelder om te formuleren, maar kan wel met behulp van lineaire voorwaarden)

(iii) Restricties op de variabelwaarden

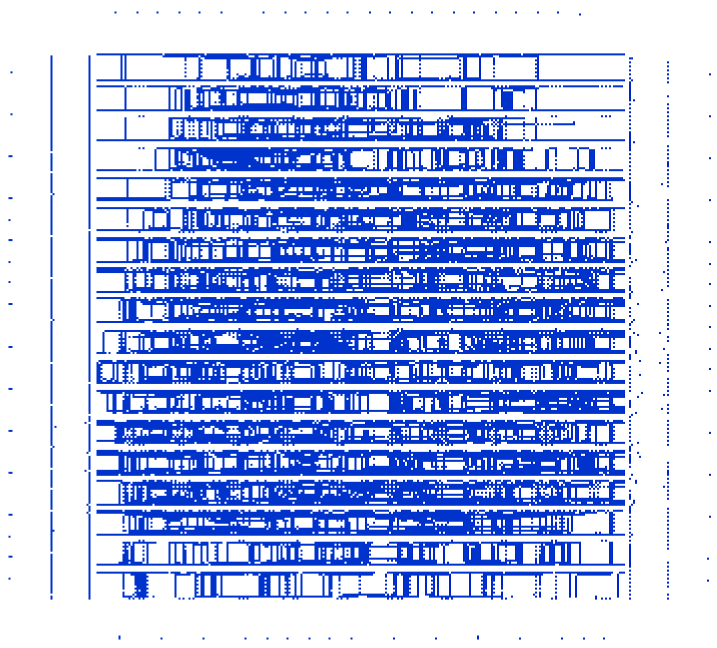
$$0 \leq x_e \leq 1, \quad \text{voor iedere } e \in E$$

$$x_e \quad \text{geheeltallig}$$

Output: De variabelwaarden 0 of 1 voor elke variabele. De variabelen met waarde 1 behoren bij verbindingen die samen een route vormen!

Om output te produceren, laten we een algoritme op het model los.

Huidig "wereldrecord": 85.900 punten in een VLSI-toepassing, opgelost in 2006



Belangrijke vragen:

Wat is een “goed” wiskundig model voor een gegeven probleem?

Wat is een “goed” algoritme om het probleem, gegeven het model, op te lossen?

Hoe ziet er een “optimaliteitsbewijs” eruit?

Het net beschreven probleem is een toepassing van het Handelsreizigersprobleem (Traveling Salesman Problem, TSP)
Zie http://www.tsp.gatech.edu/ voor een boel info!

Wat is optimalisering?

Bepaal “de beste” oplossing uit een “gegeven” verzameling oplossingen.

Wat is hier moeilijk aan????

De oplossingen zijn *impliciet gegeven* als een verzameling vectoren die voldoen aan gegeven voorwaarden.
Hoe kunnen we oplossingen karakteriseren?
Oneindig veel oplossingen?

Ook in gevallen waar wij de oplossingen expliciet zouden kunnen opschrijven en vergelijken, zou dit in de meeste gevallen *veel te lang duren*.

WIJ HEBBEN WISKUNDE NODIG (algebra, discrete wiskunde, analyse, algoritmiek…)!!!

OPTIMALISERING

Algemene formulering:

$$\begin{array}{rrcll} & \min f(x) & & & \\ \text{o.d.v.} & g_i(x) & \geq & 0, & i = 1, \ldots, n \\ & h_j(x) & = & 0, & j = 1, \ldots, p \end{array}$$

$f, \; g_i$ en h_j zijn functies van $x \in \mathbb{R}^n$.

o.d.v. = “onder de voorwaarden”
In het Engels: s.t. = “subject to”

Speciale gevallen:

- f is convex, g_i concaaf, h_j lineair: *convexe optimalisering*
- f, g_i, h_j lineair: *lineaire optimalisering*

  Als $x \in \mathbb{Z}^n$: *lineaire geheeltallige optimalisering*

Focus van dit vak

LINEAIRE PROGRAMMERINGS- problemen: LP-problemen

LINEAIRE GEHEELTALLIGE PROGRAMMERINGS-problemen:
(Eng: INTEGER LINEAR PROGRAMMING): ILP- of IP-problemen

Het gebied ontstond in de 1940-er jaren uit de behoefte om grote logistieke operaties te plannen ("programming") ivm WW II. Toen kwam ook de eerste computers en kon er gerekend worden!

Belangrijke artikelen:

George B. Dantzig (1951): *Maximization of a linear function of variables subject to linear inequalities.*

Ralph E. Gomory (1958): *Outline of an algorithm for integer solutions to linear programs*.

Een aantal definities

Def. Een *instantie* (geval) van een optimaliseringsprobleem is een paar F, c waarbij F is de verzameling toegelaten oplossingen en c is de kostenfunctie (doelfunctie):

$$c : F \to \mathbb{R}$$

Voorbeeld van zo net:

$$F = \{x \in \mathbb{R}^{|E|} \mid x \text{ voldoet aan alle voorwaarden}\}$$

$$c : x \to \sum_{e \in E} l_e x_e$$

Optimaliseringsprobleem: Bepaal $f \in F$ zodanig dat

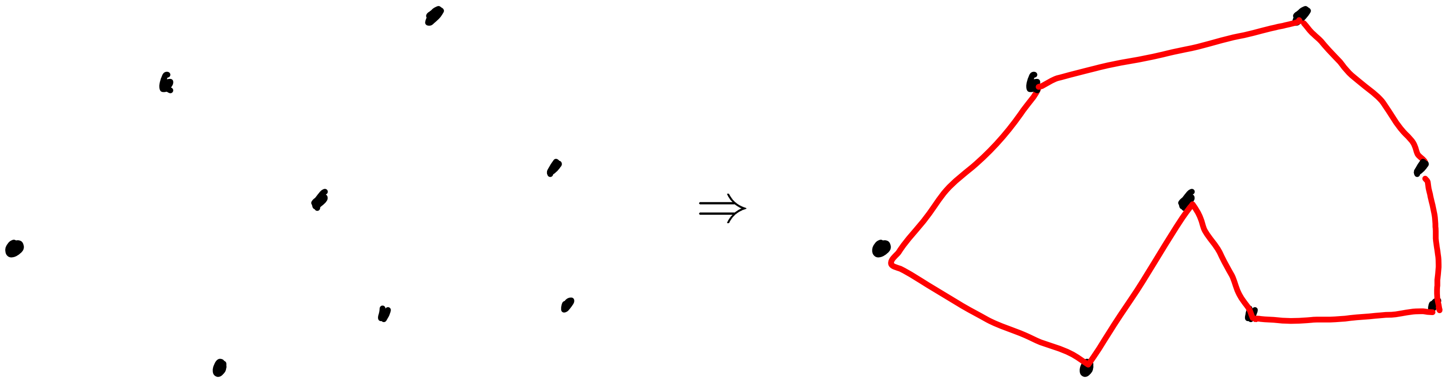
$c(f) \leq c(y)$ voor iedere $y \in F$

Def. Een *optimaliserings-probleem(type)* is de verzameling van al zijn instanties.

Voorbeelden:

1. Het Handelsreizigersprobleem (Traveling Salesman Problem (TSP))

Gegeven zijn *n* steden en een *n x n* afstandenmatrix $[d_{ij}]$.
Een *tour (of route)* is een gesloten pad dat elke "stad" precies een keer bezoekt. Bepaal de kortste tour.



2. Lineaire optimalisering, LP

Instantie van LP:

$F = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$ (of $F = \{x \in \mathbb{R}^n \mid Ax \geq b,\ x \geq 0\}$)

(toegelaten "gebied")

$c : x \to c^T x = \sum_j c_j x_j$ (doelfunctie)

Zo gaan we het LP-probleem vaak opschrijven:

$$\boxed{\begin{aligned} &\min c^T x \\ &\text{odv } Ax \;=\; b, \\ &\quad\ (Ax \;\geq\; b,) \\ &\qquad\ \ x \;\geq\; 0 \end{aligned}}$$

NEIGHBORHOODS

Gegeven een toegelaten oplossing $f \in F$ van een bepaald probleem, *de buurruimte van* f is de verzameling oplossingen die in een bepaalde opzicht "dichtbij" f zijn.

Def. Gegeven een optimaliseringsprobleem met instanties (F, c), een *buurruimte N* (neighborhood) is een afbeelding

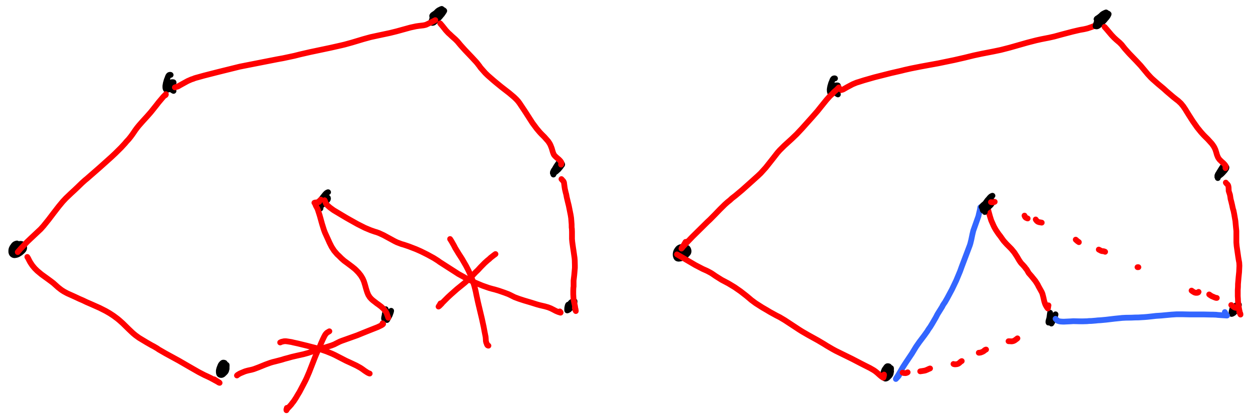
$$N : F \to 2^F$$

die gedefinieerd is voor elke instantie.

Voorbeeld: TSP, 2-exchange neighborhood

$N_2(f) =$ verzameling oplossingen die kunnen worden verkregen door het verwijderen van 2 kanten van de tour en daarna weer toevoegen van 2 nieuwe kanten zodat er een nieuwe tour wordt gecreëerd.

Een “2-exchange”:



Def. Een oplossing $f \in F$ is een *locaal optimum* mtb N als

$$c(f) \leq c(g) \quad \text{voor iedere} \quad g \in N(f)$$

Een oplossing $f \in F$ is een *globaal optimum* als

$$c(f) \leq c(g) \quad \text{voor iedere} \quad g \in F$$

Def. Gegeven een optimaliseringsprobleem met verzameling toegelaten oplossingen F, als wanneer $f \in F$ locaal optimaal is mbt N, f ook globaal optimaal is, dan noemen we de buurruimte N *exact*.

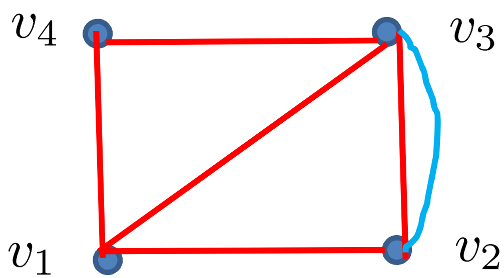
Voorbeeld: Voor TSP: $N_2(f)$ is niet exact, maar $N_n(f)$ wel.

Grafen en netwerken

Een *graaf G* is een paar *G=(V,E)*
V = verzameling *knooppunten/knopen* (nodes, vertices)
E = verzameling *kanten* (edges); ongeordend paar punten

Voorbeeld:

$$G = (\{v_1, v_2, v_3, v_4\},\ \{[v_1, v_2],\ [v_2, v_3],\ [v_3, v_4],\ [v_4, v_1],\ [v_1, v_3]\})$$

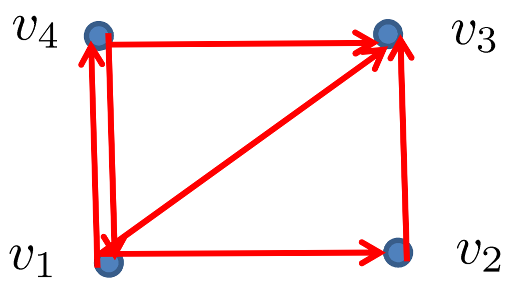


Multigraaf: *paralelle* kanten toegestaan

Een *gerichte graaf G* is een paar *G=(V,A)*
V = verzameling *knooppunten/knopen* (nodes, vertices)
A = verzameling *pijlen* (arcs); *geordende* paren knopen

Voorbeeld:

$$G = (\{v_1, v_2, v_3, v_4\},\ \{(v_1, v_2),\ (v_2, v_3),\ (v_4, v_3),\ (v_4, v_1),\ (v_1, v_4),\ (v_1, v_3)\})$$



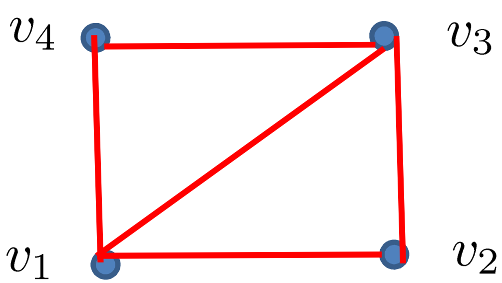
Begrippen

Ongerichte grafen $G = (V, E),\ e = [v_1, v_2] \in E$:

v_1 *grenst aan* v_2 (en vv) (v_1 *adjacent to* v_2) als er een kant $e = [v_1, v_2]$ bestaat

$e = [v_1, v_2]$ *raakt (is incident to)* v_1 en v_2

De *graad* (degree) van een knoop v is het aantal kanten die v raakt

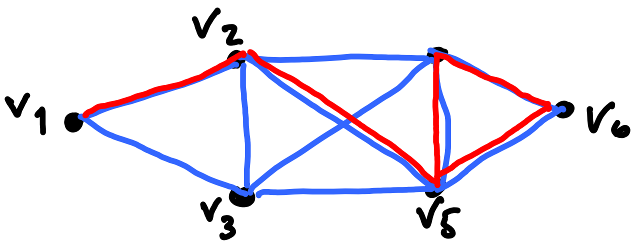


De graad van v_1 is gelijk aan 3

Een *wandeling* w in G is een aaneenschakeling van knooppunten

$$w = [v_1, v_2, v_3, \ldots, v_k], \;\; k \geq 1$$

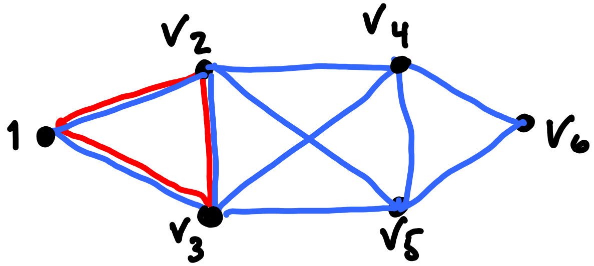
zodanig dat $[v_j, v_{j+1}] \in E, j = 1, \ldots k-1.$



Wandeling:

$w = [v_1, v_2, v_5, v_6, v_4, v_5]$

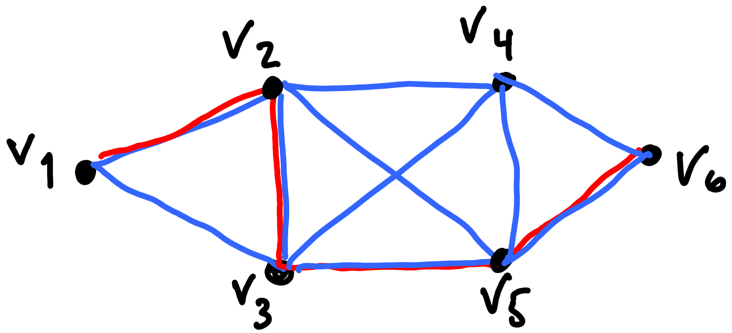
De wandeling is *gesloten* als $k > 1$ en $v_k = v_1$.



Gesloten wandeling

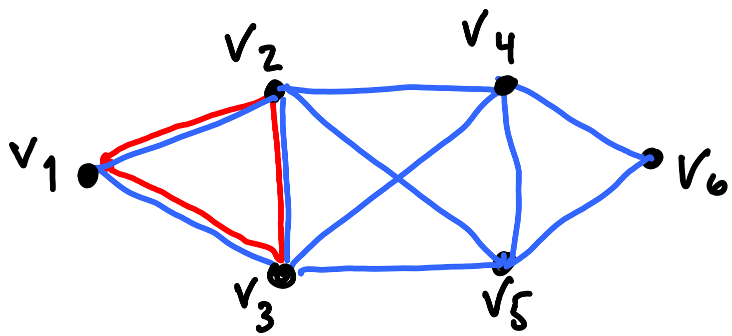
$[v_1, v_2, v_3, v_1]$

Een wandeling zonder repetities van knopen is een *pad (path).*



Pad
[v1, v3, v5, v6]

Een gesloten wandeling zonder repetities van knopen is een *kring (circuit/cycle).*

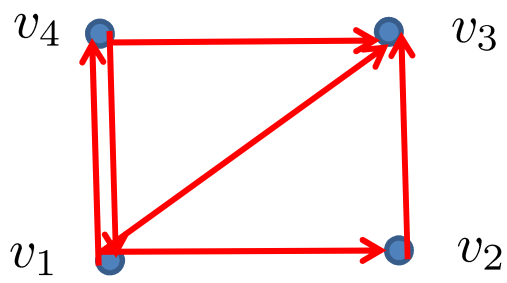


Gesloten wandeling
[v1, v2, v3, v1]
kring

Gerichte grafen $G = (V, A)$:

De *in-graad* van knoop v, $\delta^-(v)$, is het aantal pijlen dat v als eindpunt heeft *(indegree)*

De *uit-graad* van knoop v, $\delta^+(v)$, is het aantal pijlen dat v als beginpunt heeft *(outdegree)*



$$\delta^-(v_1) = 1$$

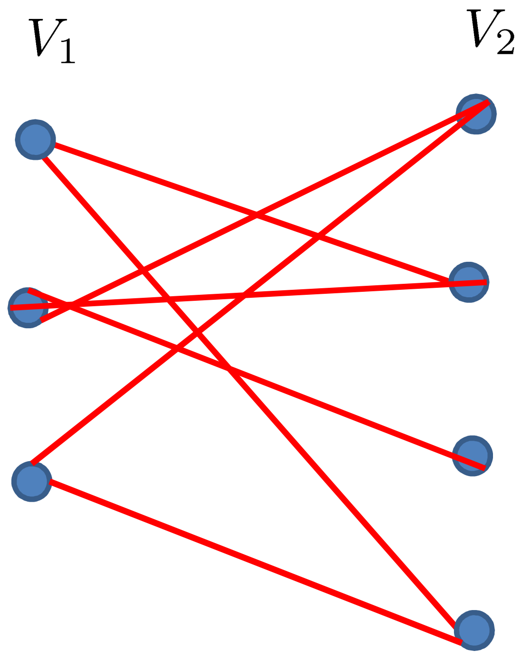
$$\delta^+(v_1) = 3$$

$$\delta(v) = \delta^-(v) + \delta^+(v)$$

Analoog met het ongerichte geval hebben wij
gerichte wandeling, gericht pad, gerichte kring

Bipartiete grafen

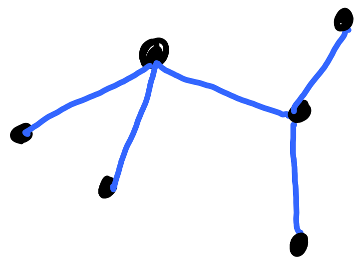
Een graaf $G = (V, E)$ heet *bipartiet* als er een partitie van de knopen V in V_1 en V_2 bestaat, zodanig dat elke kant $e \in E$ één eindpunt in V_1 en één eindpunt in V_2 heeft.



Boom en bos

$G = (V, E)$ is *samenhangend* als er een pad bestaat tussen elk tweetal knopen.

Een *boom (tree)* $G = (V, T)$ is een samenhangende graaf zonder kringen



$|V| = 6$
$|T| = 5$

Observatie: Zij $G = (V, E)$ een ongerichte graaf. De volgende beweringen zijn equivalent:

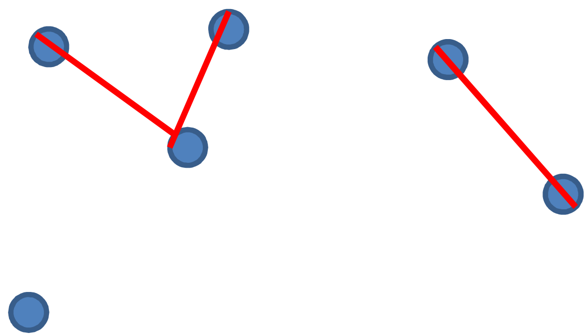
- G is een boom
- G is samenhangend en heeft $|V| - 1$ kanten
- G heeft geen kringen, maar als een kant aan G wordt toegevoegd dan ontstaat er een unieke kring

Een deelgraaf $G = (V, T)$ van een samenhangende graaf $G' = (V', E)$ is een *opspannende boom* in G' als G een boom is en als $V = V'$.

Een *bos* is een verzameling knoop-disjuncte bomen

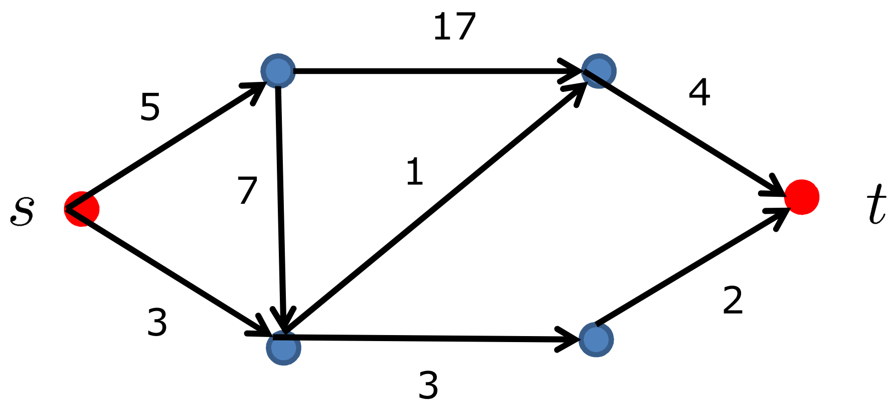
$$F = \{(V_1, T_1), \ldots, (V_k, T_k)\}$$

Voorbeeld:



Netwerk

Een *netwerk* $N = (s, t, V, A, b)$ is een gerichte graaf (V, A) samen met een *startpunt (source)* s met in-graad 0, een *eindpunt (terminal)* t met uit-graad 0, *bovengrens* (capaciteit) $b(u, v) \in \mathbb{Z}_+$ op de stroom op elke pijl $(u, v) \in A$.

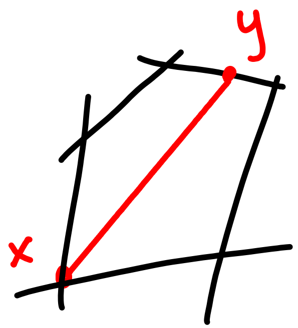


CONVEXE FUNCTIES EN VERZAMELINGEN

Def. Een *convexe combinatie* van twee gegeven punten $x, y \in \mathbb{R}^n$ is elk punt z dat geschreven kan worden als

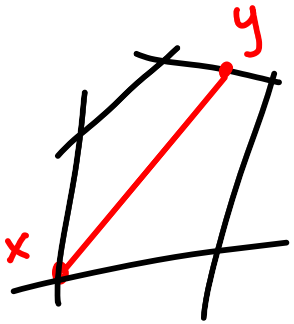
$$z = \lambda x + (1-\lambda)y, \quad 0 \leq \lambda \leq 1$$

dat wil zeggen, elk punt dat op het lijnsegment ligt tussen (en inclusief) de punten x en y.

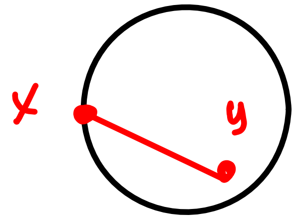


Def. Een *verzameling* $S \subseteq \mathbb{R}^n$ is *convex* als die alle convexe combinaties bevat van paren punten $x, y \in S$.

Convex:

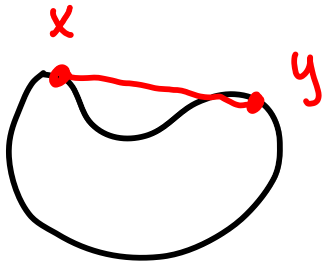


Polytoop



Bal

Niet-convex:



Lemma. De doorsnede van convexe verzamelingen S_i is een convexe verzameling.

Bewijs. Als x en y tot $\cap S_i$ behoren, dan behoren ze tot elke deelverzameling S_i. Iedere convexe combinatie van x en y behoren dan ook tot elke S_i, en daardoor ook tot $\cap S_i$.

Def. Zij $S \subseteq \mathbb{R}^n$ een convexe verzameling. De functie

$$c : S \mapsto \mathbb{R}$$

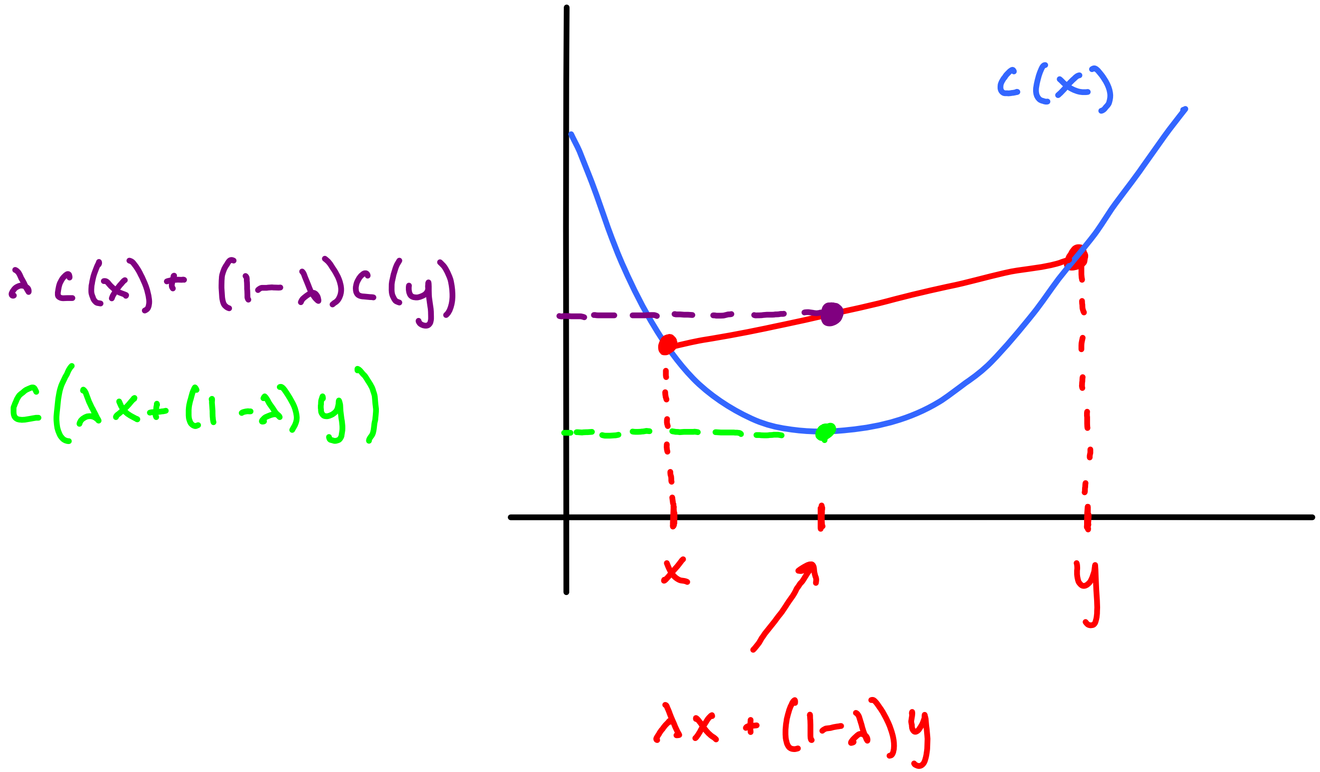
is *convex in* S als voor ieder paar punten $x, y \in S$ geldt dat

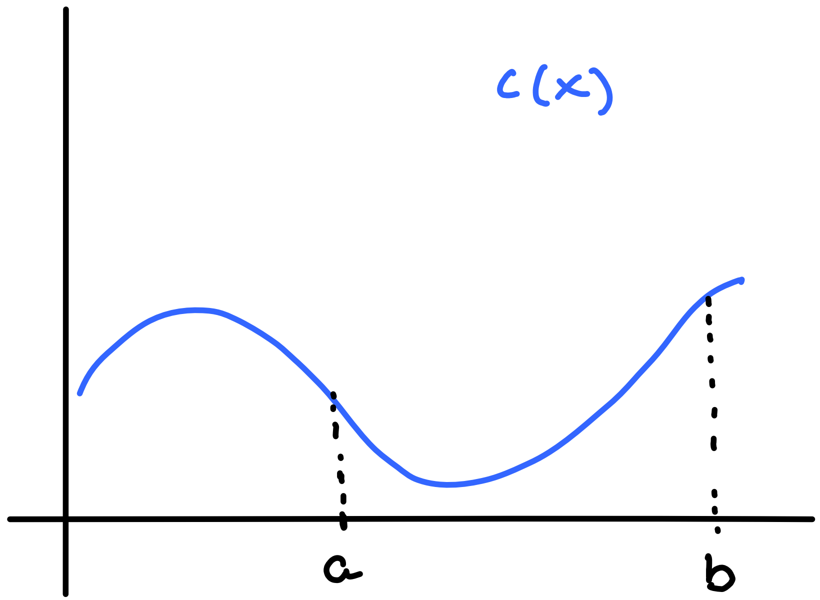
$$c(\lambda x + (1-\lambda)y) \leq \lambda c(x) + (1-\lambda)c(y), \quad 0 \leq \lambda \leq 1$$

Als $S = \mathbb{R}^n$ zeggen we dat c *convex* is

Voorbeelden:

$$c(\lambda x + (1-\lambda)y) \leq \lambda c(x) + (1-\lambda)c(y), \quad 0 \leq \lambda \leq 1$$





c is convex in [a,b]

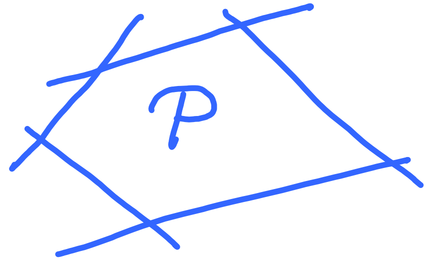
Def. Een functie c gedefinieerd in een convexe verzameling $S \subseteq \mathbb{R}^n$ noemen we *concaaf* als $-c$ convex is in S.

Als we een lineaire functie minimaliseren (of maximaliseren) over een convexe verzameling S, is een minimum (maximum) te vinden in een *extreempunt* van S.

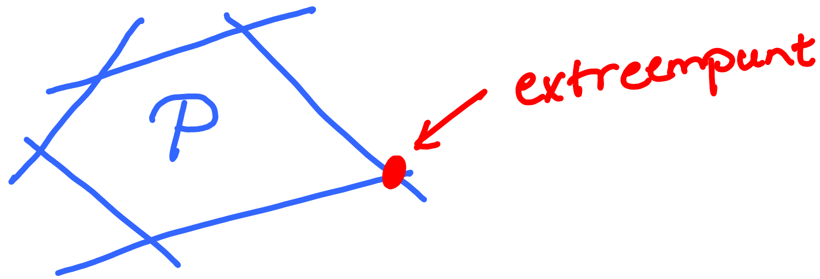
Eerste helft van het college: Lineaire optimaliseringsproblemen (LP).

De verzameling toegelaten oplossingen van een LP is een polyeder.
Een polyeder *P* is een verzameling punten die voldoet aan een eindig aantal lineaire ongelijkheden.

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$



Een extreempunt van een polyeder is een "hoekpunt".



Intuitie van het Simplexalgoritme voor LP:

Ga van een hoekpunt van de polyeder het toegelaten gebied definieert naar een volgende hoekpunt zodanig dat de doelfunctiewaarde niet slechter wordt!