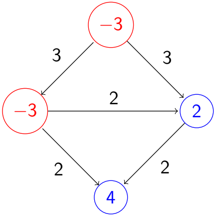
Max-flow and min-cut

Circulation with demands

- We introduce another flow problem, to deal with supply and demand inside a network.
- Instead of having a pair source/sink the new setting consider a producer/consumer scenario.
- Some nodes are able to produce a certain amount of flow.
- Some nodes are willing to consume flow.
- The question is whether it is possible to route "all" the produced flow to the consumers. When possible the flow assignment is called a circulation

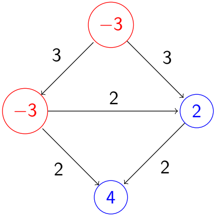
Network with demands

A network with demands $\mathcal{N}$ is a tuple (V, E, c, d) where c assigns a positive capacity to each edge, and d is a function associating a demand $d(v)$, to $v \in V$.



Network with demands

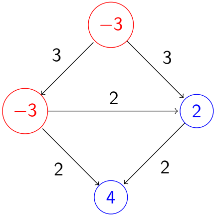
A network with demands $\mathcal{N}$ is a tuple (V, E, c, d) where c assigns a positive capacity to each edge, and d is a function associating a demand $d(v)$, to $v \in V$.



- When $d(v) > 0$, v can receive $d(v)$ units of flow more than it sends, v is a sink,.
- If $d(v) < 0$, v can send $d(v)$ units of flow more than it receives, v is a source.
- If $d(v) = 0$, v is neither a source or a sink.

Network with demands

A network with demands $\mathcal{N}$ is a tuple (V, E, c, d) where c assigns a positive capacity to each edge, and d is a function associating a demand $d(v)$, to $v \in V$.



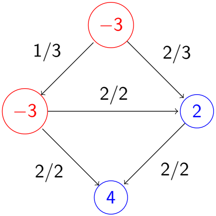
- When $d(v) > 0$, v can receive $d(v)$ units of flow more than it sends, v is a sink,.
- If $d(v) < 0$, v can send $d(v)$ units of flow more than it receives, v is a source.
- If $d(v) = 0$, v is neither a source or a sink.
- Define S to be the set of sources and T the set of sinks.

Network with demands: circulation

Given a network $\mathcal{N} = (V, E, c, d)$, a circulation is a flow assignment $f : E \to \mathbb{R}^+$ s.t.

1. capacity: For each $e \in E$, $0 \leq f(e) \leq c(e)$,
2. conservation: For each $v \in V$,

$$\sum_{(u,v)\in E} f(u,v) - \sum_{(v,z)\in E} f(v,z) = d(v).$$

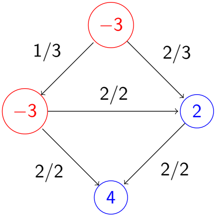


Network with demands: circulation

Given a network $\mathcal{N} = (V, E, c, d)$, a circulation is a flow assignment $f : E \to \mathbb{R}^+$ s.t.

1. capacity: For each $e \in E$, $0 \leq f(e) \leq c(e)$,
2. conservation: For each $v \in V$,

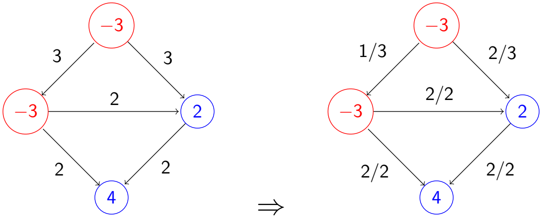
$$\sum_{(u,v)\in E} f(u,v) - \sum_{(v,z)\in E} f(v,z) = d(v).$$



Take into account that a circulation might not exist.

Network with demands: circulation problem

Circulation problem: Given $\mathcal{N} = (V, E, c, d)$ with $c > 0$, obtain a circulation provided it does exists.



A first conditions for a circulation to exists

If f is a circulation for $\mathcal{N} = (V, E, c, d)$,

$$\sum_{v \in V} d(v) = \sum_{v \in V} \left(\underbrace{\sum_{(u,v) \in E} f(u,v)}_{\text{edges to } v} - \underbrace{\sum_{(v,z) \in E} f(v,z)}_{\text{edges out of } v} \right).$$

A first conditions for a circulation to exists

If f is a circulation for $\mathcal{N} = (V, E, c, d)$,

$$\sum_{v \in V} d(v) = \sum_{v \in V} \left(\underbrace{\sum_{(u,v) \in E} f(u, v)}_{\text{edges to } v} - \underbrace{\sum_{(v,z) \in E} f(v, z)}_{\text{edges out of } v} \right).$$

For $e = (u, v) \in E$, $f(e)$ appears in the sum of edges to v and in the sum of edges out of u. Both terms cancel!

A first conditions for a circulation to exists

If f is a circulation for $\mathcal{N} = (V, E, c, d)$,

$$\sum_{v \in V} d(v) = \sum_{v \in V} \left(\underbrace{\sum_{(u,v) \in E} f(u,v)}_{\text{edges to } v} - \underbrace{\sum_{(v,z) \in E} f(v,z)}_{\text{edges out of } v} \right).$$

For $e = (u, v) \in E$, $f(e)$ appears in the sum of edges to v and in the sum of edges out of u. Both terms cancel!

Then, $\sum_{v \in V} d(v) = 0$.

A first conditions for a circulation to exists

If there is a circulation, then $\sum_{v \in V} d(v) = 0$.

Recall that
$S = \{v \in V | d(v) < 0\}$ and
$T = \{v \in V | d(v) > 0\}$.

Define $D = -\sum_{v \in S} d(v) = \sum_{v \in T} d(v)$.

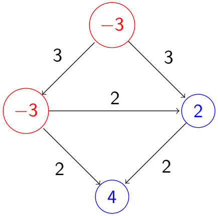
A first conditions for a circulation to exists

If there is a circulation, then $\sum_{v \in V} d(v) = 0$.

Recall that
$S = \{v \in V | d(v) < 0\}$ and
$T = \{v \in V | d(v) > 0\}$.

Define $D = -\sum_{v \in S} d(v) = \sum_{v \in T} d(v)$.

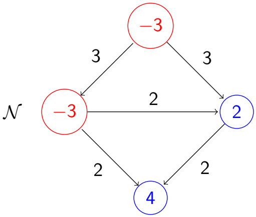
D is the total amount of extra flow that has to be transported from the sources to the sinks.



Circulation problem: reduction to Max-flow

From $\mathcal{N} = (V, E, c, d)$, define a flow network $\mathcal{N}' = (V', E', c', s, t)$:

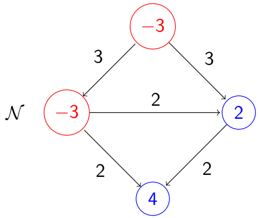
- $V' = V \cup \{s, t\}$, we add a source s and a sink t.
- For $v \in S$ $(d(v) < 0)$, add (s, v) with capacity $-d(v)$.
- For $v \in T$ $(d(v) > 0)$, add (v, t) with capacity $d(v)$.
- Keep E and, for $e \in E$, $c'(e) = c(e)$.

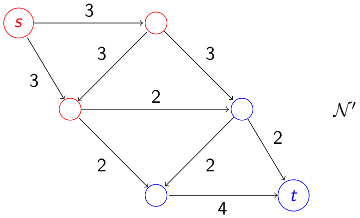


Circulation problem: reduction to Max-flow

From $\mathcal{N} = (V, E, c, d)$, define a flow network $\mathcal{N}' = (V', E', c', s, t)$:

- $V' = V \cup \{s, t\}$, we add a source s and a sink t.
- For $v \in S$ $(d(v) < 0)$, add (s, v) with capacity $-d(v)$.
- For $v \in T$ $(d(v) > 0)$, add (v, t) with capacity $d(v)$.
- Keep E and, for $e \in E$, $c'(e) = c(e)$.





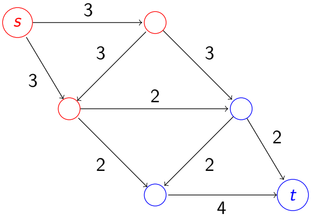
Circulation problem: reduction to Max-flow

1.- Every flow f' in $\mathcal{N}'$ verifies $|f'| \leq D$

Circulation problem: reduction to Max-flow

1.- Every flow f' in $\mathcal{N}'$ verifies $|f'| \leq D$

The capacity $c'(\{s\}, V) = D$, by the capacity restriction on flows, $|f'| \leq D$.



$\mathcal{N}'$

Circulation problem: reduction to Max-flow

2.- If there is a circulation f in $\mathcal{N}$, we have a max-flow f' in $\mathcal{N}'$ with $|f'| = D$.

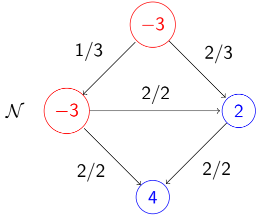
Extend f to a flow f', assigning $f'(s, v) = -d(v)$, for $v \in S$, and $f'(u, t) = d(u)$, for $u \in T$.

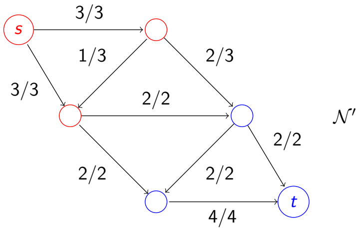
Circulation problem: reduction to Max-flow

2.- If there is a circulation f in $\mathcal{N}$, we have a max-flow f' in $\mathcal{N}'$ with $|f'| = D$.

Extend f to a flow f', assigning $f'(s, v) = -d(v)$, for $v \in S$, and $f'(u, t) = d(u)$, for $u \in T$.

By the circulation condition, f' is a flow in $\mathcal{N}'$. Furthermore, $|f'| = D$.



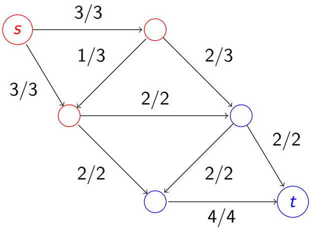


Analysis

3.- If there is a flow f' in $\mathcal{N}'$ with $|f'| = D$, $\mathcal{N}$ has a circulation

For $e \in E$, define $f(e) = f'(e)$.

- As $|f'| = D$, all edges $(s, v) \in E'$ and $(u, t) \in E'$ are saturated by f'.
- By flow conservation, f satisfies $d(v) = \underbrace{\sum_{(u,v)\in E} f(u, v)}_{\text{edges to } v} - \underbrace{\sum_{(v,z)\in E} f(v, z)}_{\text{edge.s out of } v}$.
- So, f is a circulation for $\mathcal{N}$.



Circulation: main results

From the previous discussion, we can conclude:

Theorem (Necessary and sufficient condition)

There is a circulation for $\mathcal{N} = (V, E, c, d)$ iff the maxflow in $\mathcal{N}'$ has value D.

Circulation: main results

From the previous discussion, we can conclude:

Theorem (Necessary and sufficient condition)

There is a circulation for $\mathcal{N} = (V, E, c, d)$ iff the maxflow in $\mathcal{N}'$ has value D.

Theorem (Circulation integrality theorem)

If all capacities and demands are integers, and there exists a circulation, then there exists an integer valued circulation.

Sketch Proof Max-flow formulation + integrality theorem for max-flow □

Circulation: main results

Theorem

There is a polynomial time algorithm to solve the circulation problem.

The cost of the algorithm is the same as the cost of the algorithm used for the MaxFlow computation.

Circulation: main results

Theorem

There is a polynomial time algorithm to solve the circulation problem.

The cost of the algorithm is the same as the cost of the algorithm used for the MaxFlow computation.

Theorem

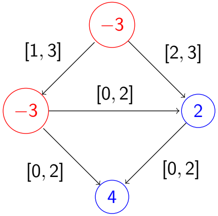
If all capacities and demands are integers, and there exists a circulation, then we can obtain an integer valued circulation in time $O(Dm)$.

Networks with demands and lower bounds

Generalization of the previous problem: besides satisfy demands at nodes, we want to force the flow to use certain edges.

Introduce a new constrain $\ell(e)$ on each $e \in E$, indicating the min-value the flow must be on e.

A network $\mathcal{N}$ with demands and lower bounds is a tuple (V, E, c, ℓ, d) with $c(e) \geq \ell(e) \geq 0$, for each $e \in E$,

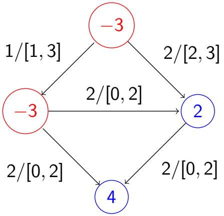


Networks with demands and lower bounds: circulation

Given a network $\mathcal{N} = (V, E, c, \ell, d)$ a circulation as a flow assignment $f : E \to \mathbb{R}^+$ s.t.

1. capacity: For each $e \in E$, $\ell(e) \leq f(e) \leq c(e)$,
2. conservation: For each $v \in V$,

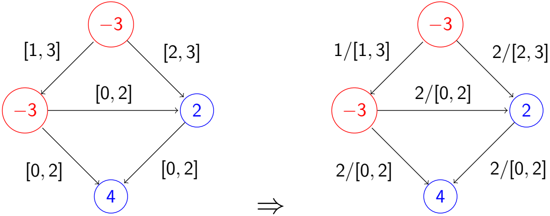
$$\sum_{(u,v)\in E} f(u,v) - \sum_{(v,z)\in E} f(v,z) = d(v).$$



A circulation might not exist.

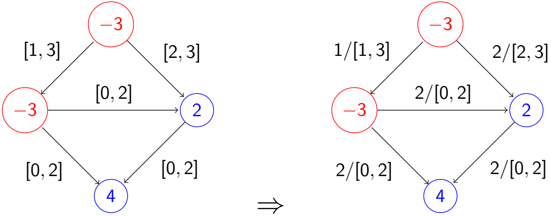
Circulations with demands and lower bounds problem

Circulation with demands and lower bounds problem: Given $\mathcal{N} = (V, E, c, \ell, d)$, obtain a circulation for $\mathcal{N}$, provided it does exists



Circulations with demands and lower bounds problem

Circulation with demands and lower bounds problem: Given $\mathcal{N} = (V, E, c, \ell, d)$, obtain a circulation for $\mathcal{N}$, provided it does exists



We devise an algorithm to the problem by a reduction to a circulation with demands problem.

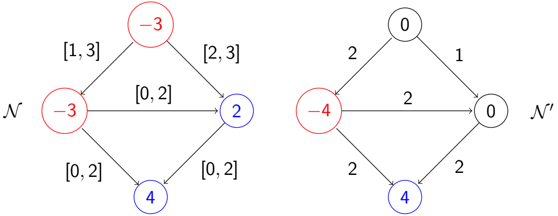
Circulations with demands and lower bounds: the reduction

Let $\mathcal{N} = (V, E, c, \ell, d)$, construct a network $\mathcal{N}' = (V, E, c', d')$ with only demands as follows:

Initially set $c' = c$ and $d' = d$.

For each $e = (u, v) \in E$, with $\ell(e) > 0$:

- $c'(e) = c(e) - \ell(e)$.
- Update the demands on both ends of e: $d'(u) = d(u) + \ell(e)$ and $d'(v) = d(v) - \ell(e)$



Circulations with demands and lower bounds: the reduction

1.- If f is a circulation in $\mathcal{N}$, $f'(e) = f(e) - \ell(e)$, for $e \in E$, is a circulation in $\mathcal{N}'$.

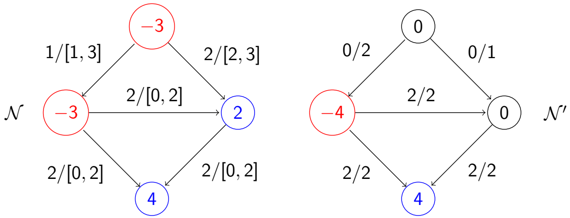
Circulations with demands and lower bounds: the reduction

1.- If f is a circulation in $\mathcal{N}$, $f'(e) = f(e) - \ell(e)$, for $e \in E$, is a circulation in $\mathcal{N}'$.

By construction of $\mathcal{N}'$, f' verifies the capacity constraint.

Besides, for (u, v) with $\ell(u, v) > 0$, the flow out of u and the flow in v is decreased by $\ell(u, v)$.

f is a circulation in $\mathcal{N}$ so, the flow imbalance of f' matches the demand d' at each node.



Circulations with demands and lower bounds: the reduction

2.- If f' is a circulation in $\mathcal{N}'$, $f(e) = f'(e) + \ell(e)$, for $e \in E$, is a circulation in $\mathcal{N}$.

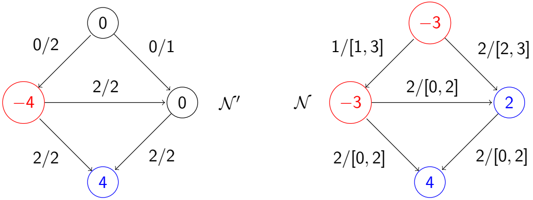
Circulations with demands and lower bounds: the reduction

2.- If f' is a circulation in $\mathcal{N}'$, $f(e) = f'(e) + \ell(e)$, for $e \in E$, is a circulation in $\mathcal{N}$.

f' verifies the capacity constraint $f'(e) \geq 0$, so $f'(e) \geq \ell(e)$.

f' is a circulation, the f' imbalance at u is $d'(u)$.

Therefore, for (u, v) with $\ell(u, v) > 0$, the increase of flow in (u, v) balances $\ell(u, v)$ units of flow out of u with $\ell(u, v)$ units of flow entering v. Thus the f imbalance at u is $d(u)$.



Main result

Theorem

There exists a circulation in $\mathcal{N}$ iff there exists a circulation in $\mathcal{N}'$. Moreover, if all demands, capacities and lower bounds in $\mathcal{N}$ are integers, and $\mathcal{N}$ admits a circulation, there is a circulation in $\mathcal{N}$ that is integer-valued.

The integer-valued circulation part is a consequence of the integer-value circulation Theorem for f' in G'.

Circulation with demands and lower bounds: main results

Theorem

There is a polynomial time algorithm to solve the circulation with demands and lower bounds problem.

The cost of the algorithm is the same as the cost of the algorithm used for the circulation with demands computation.

Circulation with demands and lower bounds: main results

Theorem

There is a polynomial time algorithm to solve the circulation with demands and lower bounds problem.

The cost of the algorithm is the same as the cost of the algorithm used for the circulation with demands computation.

Theorem

If all capacities, lower bounds, and demands are integers, and there exists a circulation, then we can obtain an integer valued circulation in time $O((D + L)m)$ where L is the sum of al lower bounds.

Survey Design problem

Problem: Design a survey among customers of products (KT-7.8)

- Customer i can only be asked about a bought product and must receive a questionnaire for at least c_i such products, those values are determined as function of the purchased products.
- For each product j, we want to collect data from a minimum of p_j customers.
- The c and p values cannot exceed the number of

Survey Design problem

The input to the problem is:
A set C of n customers and a set P of m products.

- For each customer $i \in C$, a list of purchased products and the two values $c_i \leq c'_i$.
- For each product $j \in P$, two values p_j and p'_j.

Alternatively,

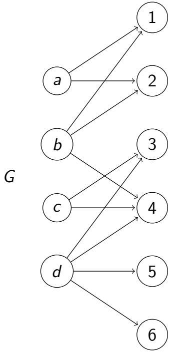
- The information about purchases can be represented as a bipartite graph $G = (C \cup P, E)$, where C is the set of customers and P is the set of products.
- $(i, j) \in E$ means $i \in C$ has purchased product $j \in P$.

Survey Design: Input

Customers $C = \{a, b, c, d\}$
Products $P = \{1, 2, 3, 4, 5, 6\}$

Customer	Bought	c
a	1,2	1
b	1,2,4	1
c	3,6	1
d	3,4,5,6	2

Prod.	1	2	3	4	5	6
d	1	1	1	1	0	1



Survey Design: Circulation with lower bounds formulation

We construct a network $\mathcal{N} = (V', E', c, \ell)$ from G as follows:

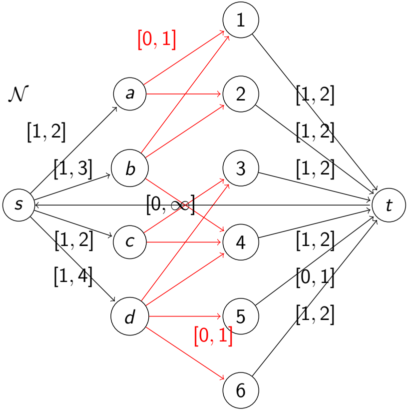
- Nodes: $V' = V \cup \{s, t\}$
- Edges: E' contains E and edges $s \to \{C\}$, $\{P\} \to t$, and (t, s).
- Capacities and lower bounds:
  - $c(t, s) = \infty$ and $\ell(t, s) = 0$
  - For $i \in C$, $\ell(s, i) = c_i$ and $c(s, i) =$ the number of purchased products.
  - For $j \in P$, $\ell(j, t) = p_j$ and $c(j, t) =$ number of customers that purchased j.
  - For $(i, j) \in E$, $c(i, j) = 1$, and $\ell(i, j) = 0$.

Survey Design: Circulation with lower bounds formulation

Customers $C = \{a, b, c, c\}$
Products $P = \{1, 2, 3, 4, 5, 6\}$

Customer	Bought	c
a	1,2	1
b	1,2,4	1
c	3,6	1
d	3,4,5,6	2

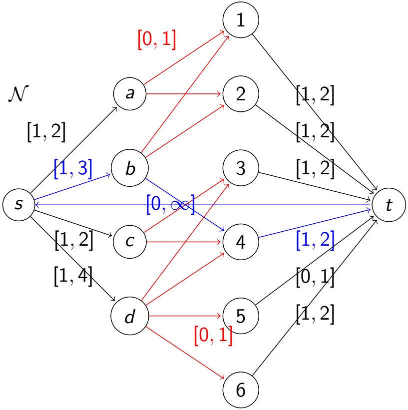
Prod.	1	2	3	4	5	6
d	1	1	1	1	0	1



Survey Design: Circulation interpretation

If f is a circulation in $\mathcal{N}$:

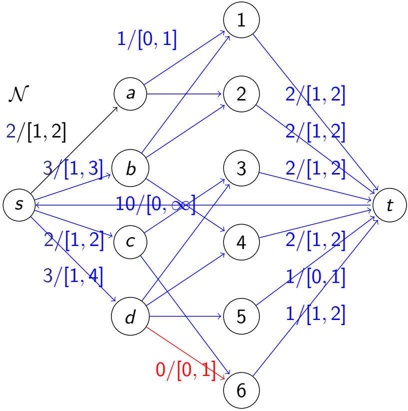
- one unit of flow circulates $s \to i \to j \to t \to s$.
- $f(i, j) = 1$ means ask i about j,
- $f(s, i)$ # products to ask i for opinion,
- $f(j, t) = \#$ customers to be asked to review j,
- $f(t, s)$ is the total number of questionnaires.



Survey Design: Circulation vs solutions

A solution

- Ask a about $1, 2$.
- Ask b about $1, 2, 4$.
- Ask c about $3, 6$.
- Ask d about $3, 4, 5$.



Main result

Theorem $\mathcal{N}$ has a circulation iff there is a feasible way to design the survey.

Proof if there is a feasible way to design the survey:

- if i is asked about j then $f(i,j) = 1$,
- $f(s,i) =$ number questions asked to i ($\geq c_i$).
- $f(j,t) =$ number of customers who were asked about j ($\geq d_j$),
- $f(t,s) =$ total number of questions.
- easy to verify that f is a circulation in $\mathcal{N}$

If there is an integral circulation in $\mathcal{N}$:

- if $f(i,j) = 1$ then i will be asked about j,
- the constrains will be satisfied by the capacity rule. □

Arrodoniment amb restriccions

Considerem una matriu $A = (a_{ij})$ amb dimensions $n \times n$, on cada $a_{ij} \in \mathbb{R} \cup \{0\}$ i on la suma de cada fila i columna de A és un enter. Volem arrodonir cada valor a_{ij} per $\lfloor a_{ij} \rfloor$ o $\lceil a_{ij} \rceil$ sense modificar la suma de les files/columnes.

Arrodoniment amb restriccions

Considerem una matriu $A = (a_{ij})$ amb dimensions $n \times n$, on cada $a_{ij} \in \mathbb{R} \cup \{0\}$ i on la suma de cada fila i columna de A és un enter. Volem arrodonir cada valor a_{ij} per $\lfloor a_{ij} \rfloor$ o $\lceil a_{ij} \rceil$ sense modificar la suma de les files/columnes.

$$\begin{pmatrix} 10.9 & 2.5 & 1.3 & 9.3 \\ 3.8 & 9.2 & 2.2 & 11.8 \\ 7.9 & 5.2 & 7.3 & 0.6 \\ 3.4 & 13.1 & 1.2 & 6.3 \end{pmatrix} \rightarrow \begin{pmatrix} 11 & 3 & 1 & 9 \\ 4 & 9 & 2 & 12 \\ 7 & 5 & 8 & 1 \\ 4 & 13 & 2 & 6 \end{pmatrix}$$

Arrodoniment amb restriccions

El problema consisteix en donada una matriu A, produir un algorisme eficient per a determinar si es possible arrodonir A i, si es possible, produir la matriu arrodonida.

Demostreu la correctesa del vostre algorisme. Quina es la complexitat del vostre algorisme?.

Notem que

- Els elements d'A que són enters no s'han de modificar.
- Sigui $r_i = \sum_{j=1}^{n}(a_{ij} - \lfloor a_{ij} \rfloor)$ i $c_j = \sum_{i=1}^{n}(a_{ij} - \lfloor a_{ij} \rfloor)$
- si les files i columnes d'A sumen un enter, aleshores r_i i c_j son enters.
- A més $\sum_i r_i = \sum_j c_j$.

Per a resoldre el problema farem una reducció d'aquest problema a un problema de circulació.
Una unitat de flux la podem interpretar com una part decimal que s'arrodoneix a 1.

Construir una xarxa amb demandes $\mathcal{N} = (V, E, c, d)$ on:

- **Vertexs:** $V = \{x_i, y_i | 1 \leq i \leq n\}$. Els vèrtexs x representen les files i els y les columnes.
- **Arestes:** $E = \{(x_i, y_j) | 1 \leq i, j \leq n \text{ i } a_{i,j} \notin \mathbb{Z}\}$
- **Capacitats:** $c(x_i, y_j) = 1$.
- **Demandes:** $d(x_i) = -r_i$, $1 \leq i \leq n$, i $d(y_j) = c_j$, $1 \leq j \leq n$.

$\mathcal{N}$ té $O(n)$ vèrtexs i $O(n^2)$ arestes.

Si existeix un arrodoniment d'A, $\mathcal{N}$ té una circulació amb valors enters (0,1).

- Sigi B un arrodoniment d'A, definim una nova matriu D on

$$d_{ij} = \begin{cases} 1 & \text{si } b_{ij} > a_{ij} \\ 0 & \text{altrament} \end{cases}$$

- Com B es un arrodoniment, $\sum_j d_{ij} = r_i$ i $\sum_i d_{ij} = c_i$.
- Llavors, el flux $f(i,j) = d_{ij}$ és una circulació a $\mathcal{N}$.

Si $\mathcal{N}$ té una circulació amb valors enters (0,1), existeix un arrodoniment d'A.

- Sigi f una circulació a $\mathcal{N}$,
- definim la matriu B com

$$b_{i,j} = \begin{cases} a_{ij} & \text{si } a_{ij} \in \mathbb{Z} \\ \lceil a_{i,j} \rceil & \text{si } a_{ij} \notin \mathbb{Z} \text{ i } f(i,j) = 1 \\ \lfloor a_{i,j} \rfloor & \text{altrament} \end{cases}$$

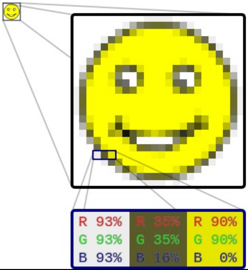
- Com f es una circulació, $\sum_j b_{ij} = \sum_j a_{ij}$ i $\sum_i b_{ij} = \sum_i a_{ij}$.
- Llavors, B és un arrodoniment vàlid d'A.

La construcció de $\mathcal{N}$ té una complexitat de $O(n^2)$.

Ford-Fulkerson funciona en $O(D|E|)$, on D és la suma de les demandes possitives, i.e. $D = \sum r_i = O(n^2)$ i com que $|E| = O(n^2)$, el nombre total de passos és $O(n^4)$.

Pixels and digital image

- In digital imaging, a pixel is the smallest controllable element of a picture represented on the screen.
- Digital images are represented by a raster graphics image, a dot matrix data structure representing rectangular grid of pixels, or points of color
- The address of a pixel corresponds to its physical coordinates.



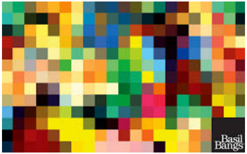
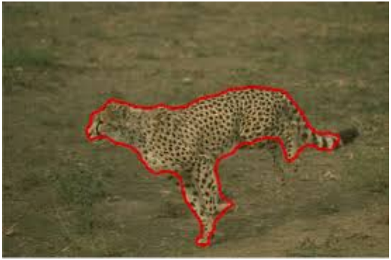
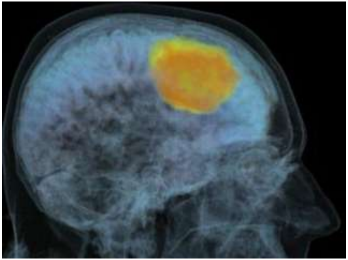


Image segmentation

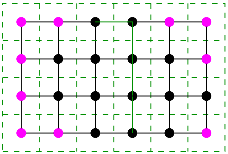
Given a set of pixels classify each pixel as part of the main object or as part of the background.
Important problem in different techniques for image processing.





Foreground/background segmentation

- We aim to label each pixel as belonging to the foreground or the background
- Picture pixels as a grid of dots.
- Define the undirected graph $G = (V, E)$, where, $V =$ set pixels in image, $E =$ pairs of neighbors of pixels (in the grid)



Foreground/background segmentation

Given information:

- For each pixel i, $a_i \geq 0$ is likelihood that i is in the foreground and $b_i \geq 0$ is likelihood that i is in the background.
- For each (i, j) of neighboring pixels, there is a separation penalty $p_{ij} \geq 0$ for placing one in the foreground and the other in the background.

Foreground/background segmentation

Goals:

- For i isolated, if $a_i > b_i$ we prefer to label i as foreground (otherwise we label i as background)
- If many neighbors of i are labeled foreground we prefer to label i as foreground. This makes the labeling smoother by minimizing the amount of foreground/background

Foreground/background segmentation

Goals:

- For i isolated, if $a_i > b_i$ we prefer to label i as foreground (otherwise we label i as background)
- If many neighbors of i are labeled foreground we prefer to label i as foreground. This makes the labeling smoother by minimizing the amount of foreground/background

We want to partition V into A (set of foreground pixels) and B (set of background pixels), such that we maximize the objective function:

$$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}| = 1} p_{ij}$$

Formulate as a min-cut problem

Segmentation has the flavor of a cut problem, but

- it is a maximization different than the min-cut,
- G is undirected,
- it does not have sink s and source t (but we can add them).

From maximization to minimization

Recall we want to maximice

$$\underbrace{\sum_{i\in A} a_i + \sum_{j\in B} b_j - \sum_{(i,j)\in E, |A\cap\{i,j\}|=1} p_{ij})}_{(*)}$$

- Let $Q = \sum_{i\in V}(a_i + b_i)$, then

$$\sum_{i\in A} a_i + \sum_{j\in B} b_j = Q - (\sum_{i\in A} b_i + \sum_{j\in B} a_j)$$

$(*) = Q - (\sum_{i\in A} b_i - \sum_{j\in B} a_j) - \sum_{(i,j)\in E, |A\cap\{i,j\}|=1} p_{ij}$

- Therefore, maximizing (*) is equivalent to minimize

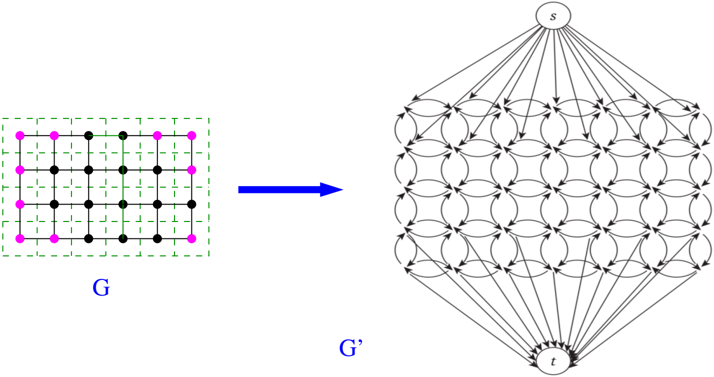
$$\sum_{i\in A} b_i + \sum_{j\in B} a_j + \sum_{(i,j)\in E, |A\cap\{i,j\}|=1} p_{ij}.$$

Transforming G into a network $\mathcal{N}$

We transform $G = (V, E)$ to $\mathcal{N} = (V', E', c, s, t)$ by

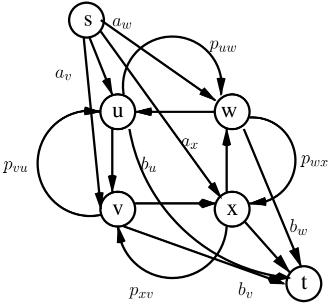
- Add a node s representing the foreground
- Add a node t representing the background
- $V' = V \cup \{s, t\}$
- For each $(v, u) \in E$ create antiparallel directed edges (u, v) and (v, u) in E'
- For each pixel i create directed edges (s, i) and (i, t)
- $E' = \{(s, v) \cup (v, t)\}_{v \in E} \cup \{(u, v) \cup (v, u)\}_{(u,v) \in E}$

The pixel graph G and the graf $G' = (V', E')$



Adding capacities to the edges of $\mathcal{N}$

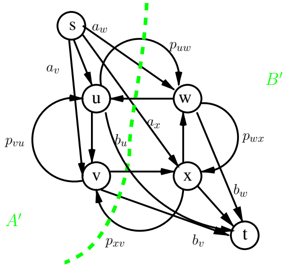
- For each $i \in V$, $c(s,i) = a_i$, $c(i,t) = b_i$
- For each $(i,j) \in E$, $c(i,j) = c(j,i) = p_{ij}$



(s,t)-cuts in $\mathcal{N}$

A (s,t)- cut (A', B') corresponds to a partition of the pixels into (A, B), for $A = A' - \{s\}$, and $B = B' - \{t\}$).

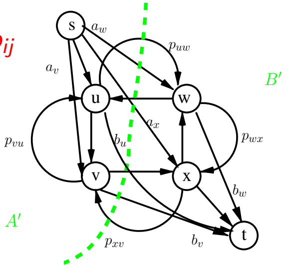
- An edge (s,j) with $j \in B$ contributes with a_j to $c(A', B')$,
- An edge (i,t) where $i \in A$ contributes with b_j to $c(A', B')$,
- An edge (i,j) where $i \in A$ and $j \in B$ contributes with p_{ij} to $c(A', B')$



(s,t)-cuts in $\mathcal{N}$

Therefore,

$$c(A',B') = \sum_{i\in A} b_i + \sum_{j\in B} a_j + \sum_{(i,j)\in E, |A\cap\{i,j\}|=1} p_{ij}$$



We want to find a cut with the min value of the above quantity, which is equivalent to solve the min-cut problem in $\mathcal{N}$