IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Graph Cut - Image Segmentation

BERCY Victor
DE CHARENTENAY Stanislas

SUMMARY

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

I - RECALL THE PROBLEM

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

- Min-Cut/Max-Flow Algorithms useful for exact or approximate energy minimization in low-level vision

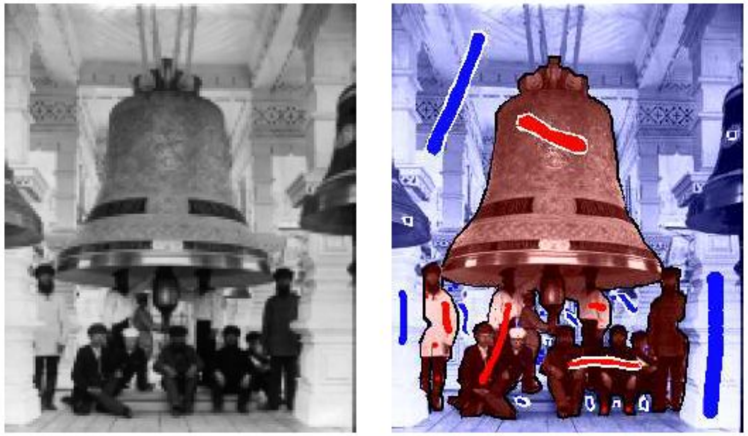
- Energy for graph-based methods :

$$E(L) = \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q)$$

with $L = \{L_P \mid p \in P\}$: labeling of image P
D_P : data penalty function
$V_{p,q}$: interaction potential
$\mathcal{N}$: set of all pairs of neighbouring pixels

- Many applications: image segmentation, restoration, stereo, augmented reality…

Example of Image Segmentation



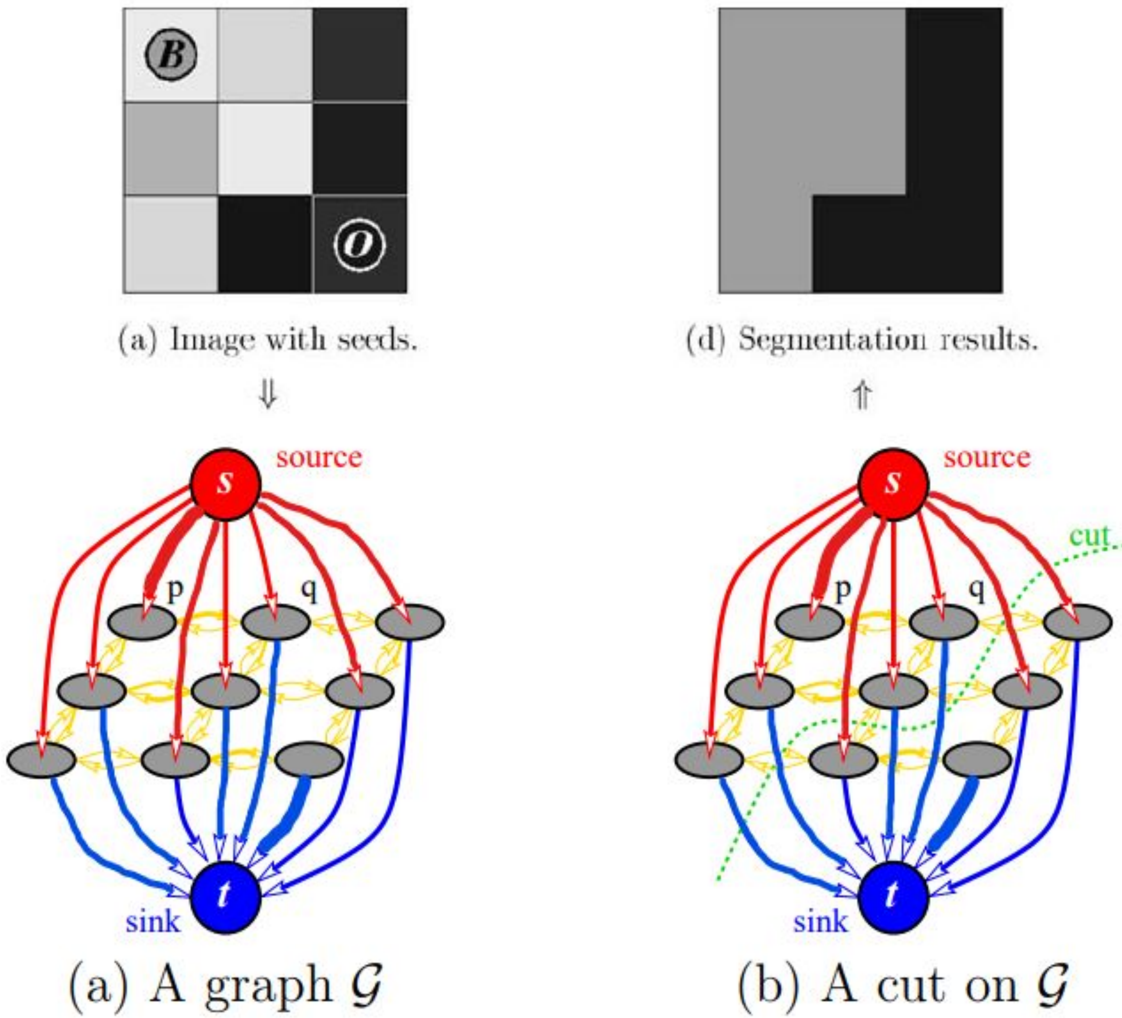
(a) Bell Photo (b) Bell Segmentation

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

I - RECALL THE PROBLEM

1.2 Application to segmentation

Complete process



Wikipedia

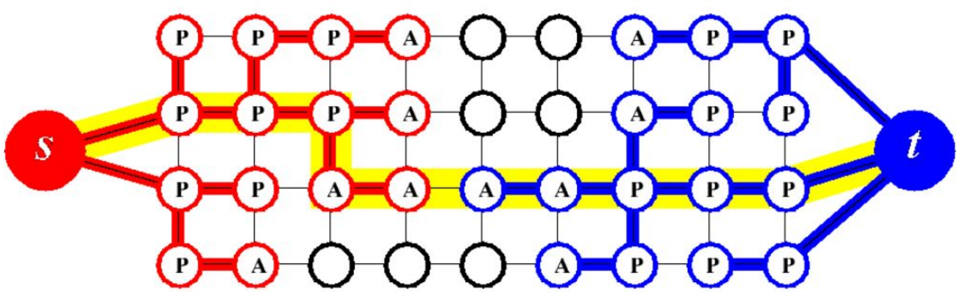
I - RECALL THE PROBLEM

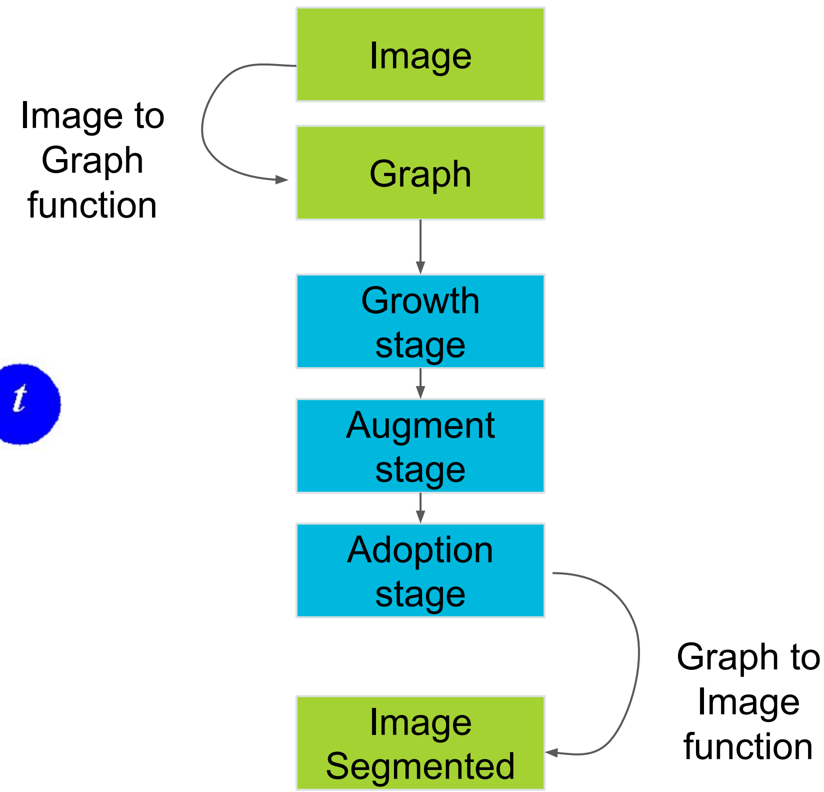
1.3 Dinic and new Min-cut Max-flow algorithms

Dinic's algorithm based on a new **breadth-first search** after each iteration → **very costly** search especially on large graphs computed from images with a large number of pixels.

Aim of the algorithm : keep as much of the information we compute at each search
→ presented algorithm : 2 search trees, one for the source and one for the sink.

Other more efficient algorithms (e.g. : based on Ford-Fulkerson style "augmenting paths")





II - EXPERIMENT

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

II - EXPERIMENT

2.1 Dataset used

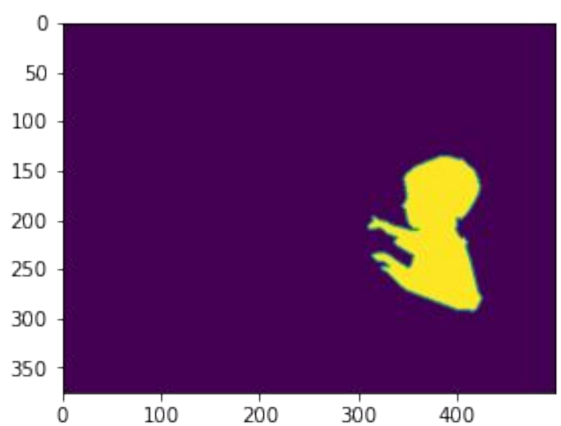
Most work done on graph cut : before 2010

→ hard to find a relevant database for **interactive binary segmentation**

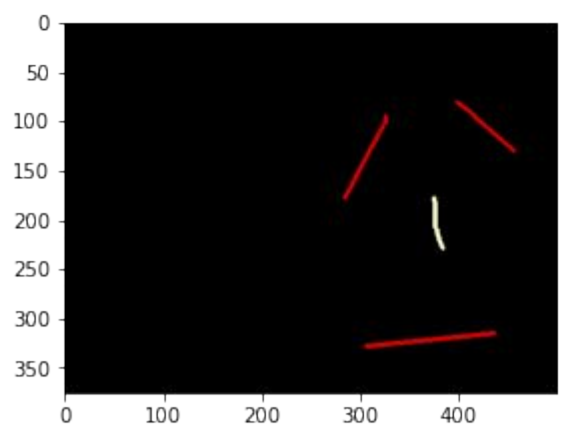
Three set of 151 images :

- Raw images
- Segmentation masks
- Initialisation

Image



Binary segmentation



Initialisation

[1] - Varun Gulshan , Carsten Rother , Antonio Criminisi , Andrew Blake and Andrew Zisserman, *Geodesic Star Convexity for Interactive Image Segmentation*

Metrics used in the dataset paper* : **Average effort** (average number of strokes to get a 98% accuracy with intersection over union metric)
→ specific and add computational time

Intersection over Union (IoU) metric : usual metric used to measure segmentation efficiency

→ Compare dice score between **several graph-cut methods** results (interactive segmentation algorithms, our own algorithm) using the previous database

$$IoU = \frac{A \cap B}{A \cup B}$$

* Visual Geometry Group - University of Oxford

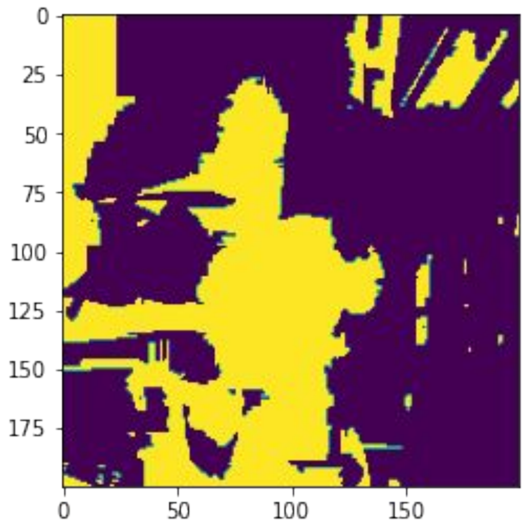
III - RESULTS

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

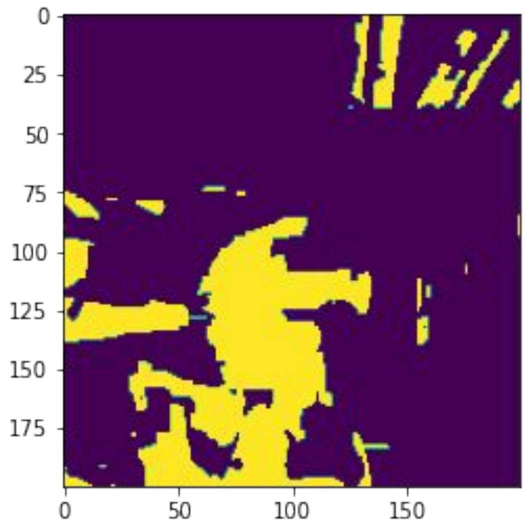
2 methods for graph-cut : **Dinic** and **Boykov-Kolmogorov** (paper) algorithms

Comparison between 2 algorithms :

- Our implementations of the Boykov-Kolmogorov algorithms (**left**)
- The NetworkX functions of an inspired Dinic algorithms (**right**)



V.S.



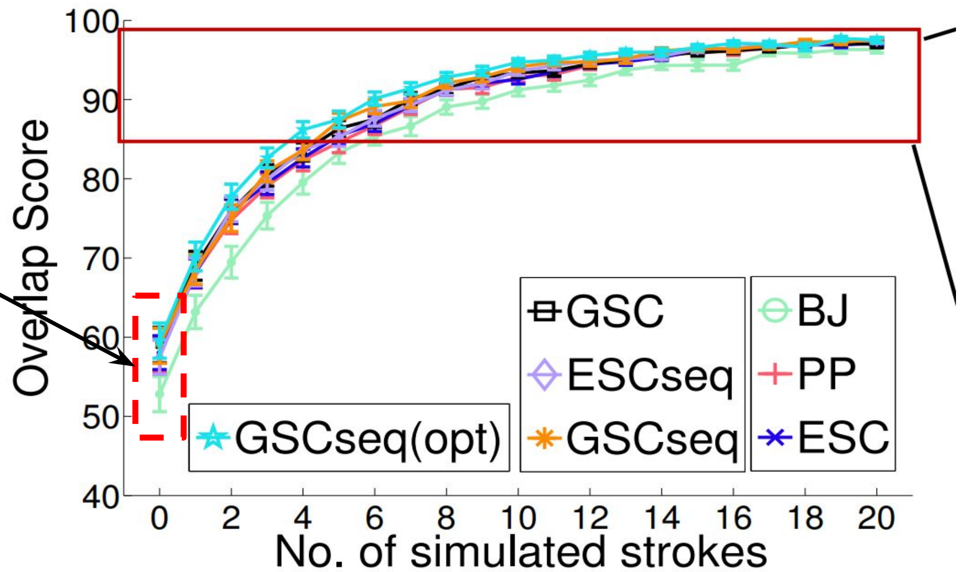
Networkx Dinic inspired :

- Networkx implementation : 13s for a 200x200 image → IoU = 46%

Our Boykov-Kolmogorov :

- Our implementation : iteration 24200, ~6 min for a 200x200 image → IoU = 43%

50 - 60% of accuracy with the IoU metric for the first iteration of the **Average effort method**

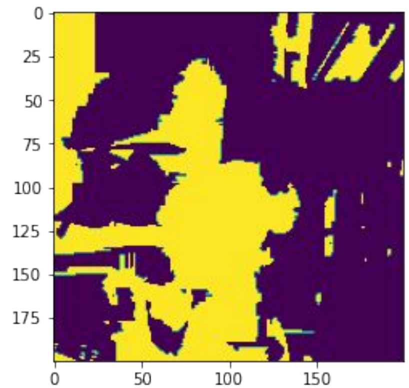


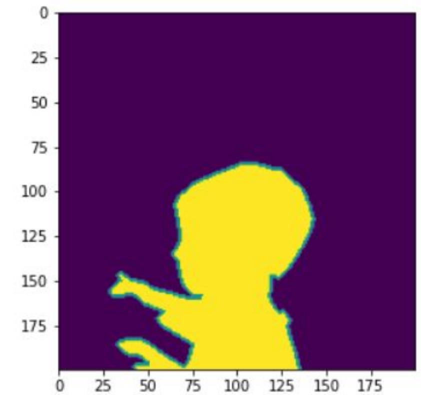
CONCLUSION

Both implementations coded from scratch using the Graph class of Networkx
→ implementation of the Boykov-Kolmogorov performs better than the Dinic

Major issues :

- our implementation of the function that creates the graph from the image → more specifically : implementation of the regularization term
- High computational time (on our computers)





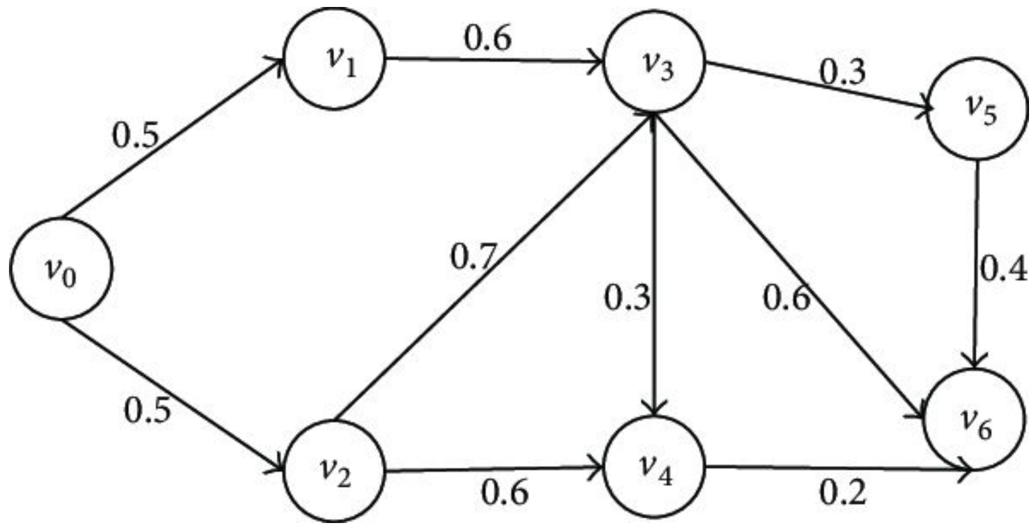
Perspectives :

- Use a robot that iterates the segmentation and changes the initialization points based on the obtained result at each iteration
- Get rid of networkx and its Graph class

Thank you for your attention

Any questions ?

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



Graph $\mathscr{G}$ = (V, E)
→ V : set of vertices linked by edges contained in E

E = {(x,y) | (x,y)∈V², x≠y} as a set of tuples of vertices

Definitions :

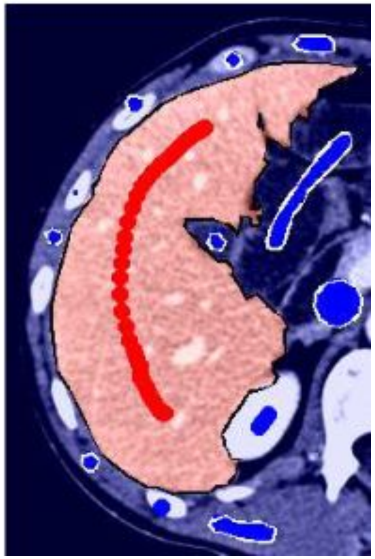
- Neighbor : two vertices x, y ∈ V are neighbors if (x, y) ∈ E
- Neighborhood of a vertex : Ɲ(x) = {y | (x, y) ∈ E}
  → symmetric relation : x ∈ Ɲ(y) ⇔ y ∈ Ɲ(x)

Properties :

- Directed vs. Undirected
- Weighted vs. Unweighted

Segmentation : Classification Problem, which pixel belongs to which class

<u>Example:</u>
Liver Segmentation



Here : Binary Segmentation : differentiate 2 classes in the image

II - CONSIDERED ISSUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

- Min-Cut/Max-Flow Algorithms useful for exact or approximate energy minimization in low-level vision

- Energy for graph-based methods :

$$E(L) = \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q)$$

with $L = \{L_P \mid p \in P\}$: labeling of image P
D_P : data penalty function
$V_{p,q}$: interaction potential
$\mathcal{N}$: set of all pairs of neighbouring pixels

- Experimental comparison of the efficiency of minimum cut/maximum flow algorithms
- Many applications: image segmentation, restoration, stereo, augmented reality…

Example of Image Segmentation

(a) Bell Photo

(b) Bell Segmentation

Example of Image Restoration

Original *Bell Quad*

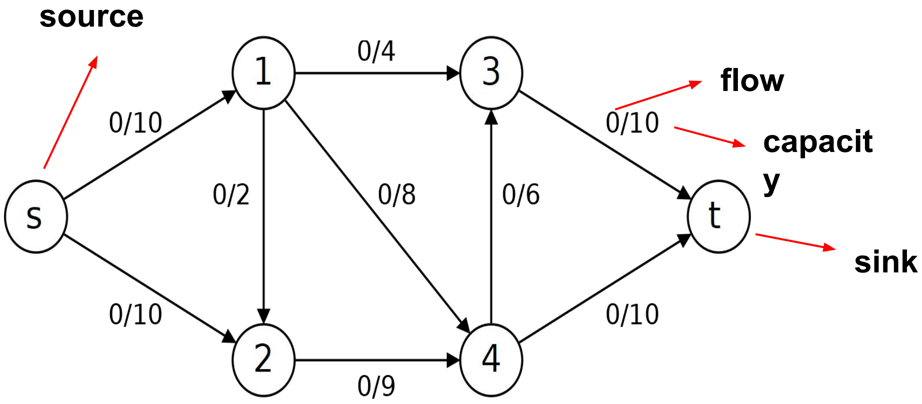
"Restored" *Bell Quad*

III - PROPOSED MODEL

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

3.1 Max-Flow Min-Cut theorem

Let $\mathscr{G}$ = (V, E) be a directed weighted (capacitated) graph consisted of a set of nodes V and a set of directed edges E that connect them



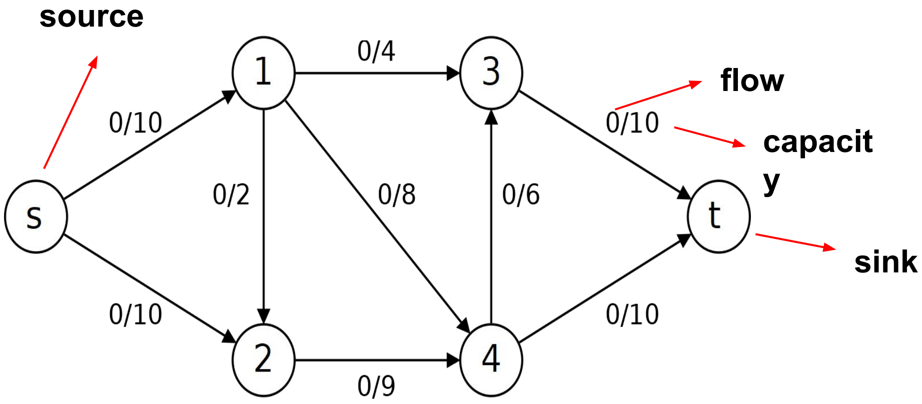
Maximum flow problem :

- **Flow :** function $f : E \to \mathbb{R}^+$ denoted by $f_{u,v}$ or f(u,v) ((u,v) ∈ E) subject to a capacity constraint ($f_{u,v} \leq c_{u,v}$) and a conservation of flow ($\forall$ v ∈ V, $\Sigma_{u\in V} f_{u,v} = \Sigma_{w \in V} f_{v,w}$)
- **Source (s) :** vertex that has no incoming flow
- **Sink (t) :** vertex that has no outgoing flow
- **Network :** set of a directed weighted graph, a source, a sink and a capacity function $c: E \to \mathbb{R}^+$

➢ **Maximum flow pb. :** route as much as possible flow as possible from s to t, or find the flow f_{max} with maximum value

3.1 Max-Flow Min-Cut theorem

Let $\mathscr{G}$ = (V, E) be a directed weighted (capacitated) graph consisted of a set of nodes V and a set of directed edges E that connect them

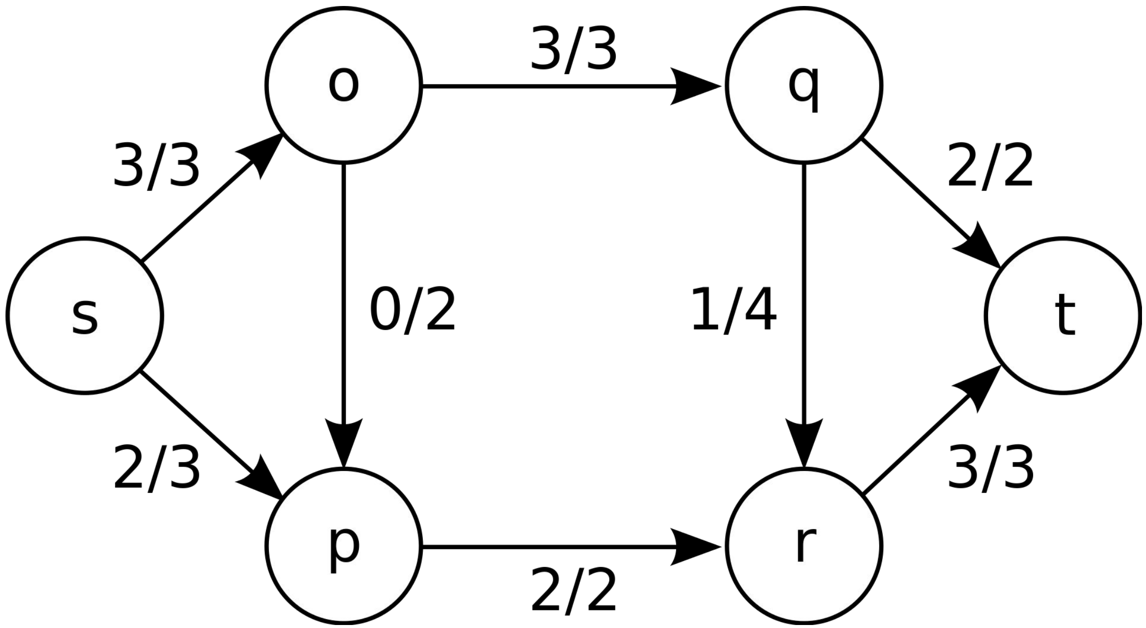


Minimum s-t cut problem :

- **Cut :** partition of the vertices of a graph into two disjoint subsets
  → s-t cut C=(S,T) => s in the S part and t in the S one
- **Cut-set (X_C = (SxT)∩E) :** set of the edges that connect the source part of the cut to the sink part
- **Capacity of a cut (c(S,T)) :** sum of the capacities of of the edges in its cut-set

➢ **Minimum s-t cut pb. :** minimize c(S,T), that is determine S and T such that the capacity of the s-t cut is minimum

Max-Flow Min-Cut theorem :
The maximum value of an s-t flow is equal to the minimum capacity over all s-t cuts



→ the maximum flow/minimum s-t cut capacity value is 5, several minimal *s-t* cuts with capacity 5 (e.g. : *S*={*s*,*p*} and *T*={*o*, *q*, *r*, *t*})

Dinic's algorithm or Dinitz's algorithm is a **strongly polynomial algorithm** for **computing** the **maximum flow** in a flow network. The worst case complexity is $O(mn^2|C|)$.

Dinic's Algorithm:

Input: A network $G = ((V, E), c, s, t)$.

Output: An s-t flow f of maximum value.

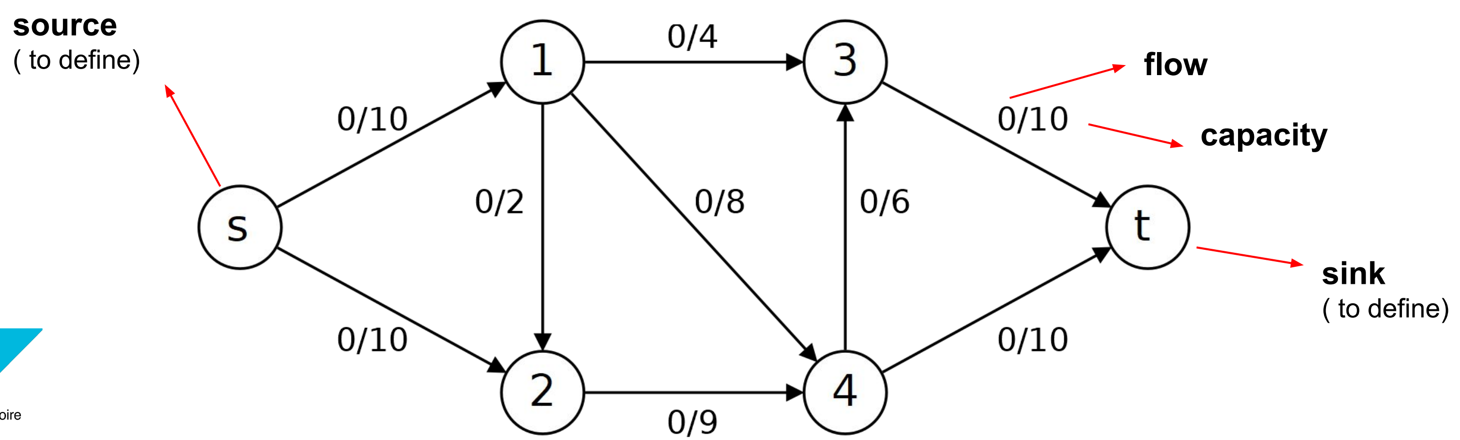
1. Set $f(e) = 0$ for each $e \in E$.

2. Construct G_L from G_f of G. If $dist(t) = \infty$, stop and output f.

3. Find a blocking flow f' in G_L.

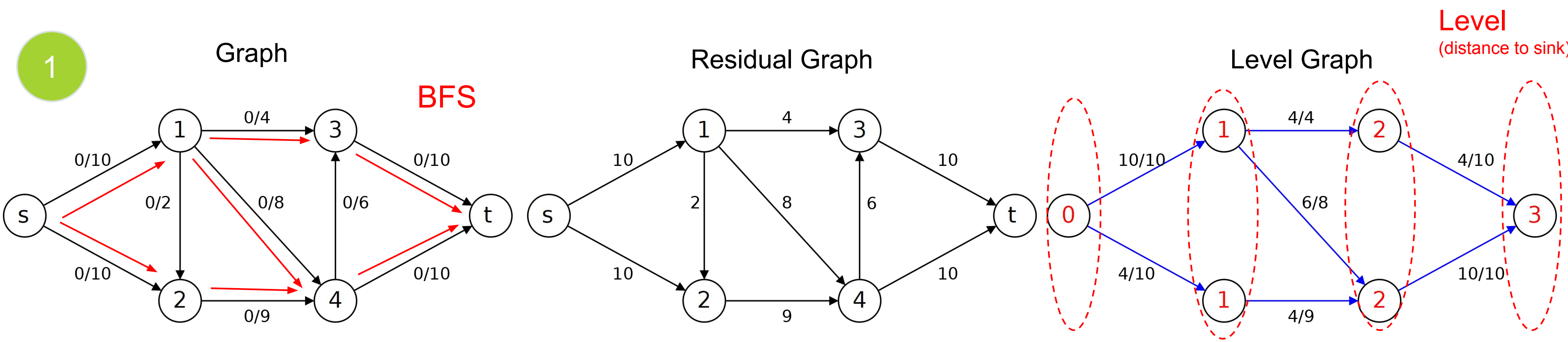
4. Add augment flow f by f, and go back to step 2.

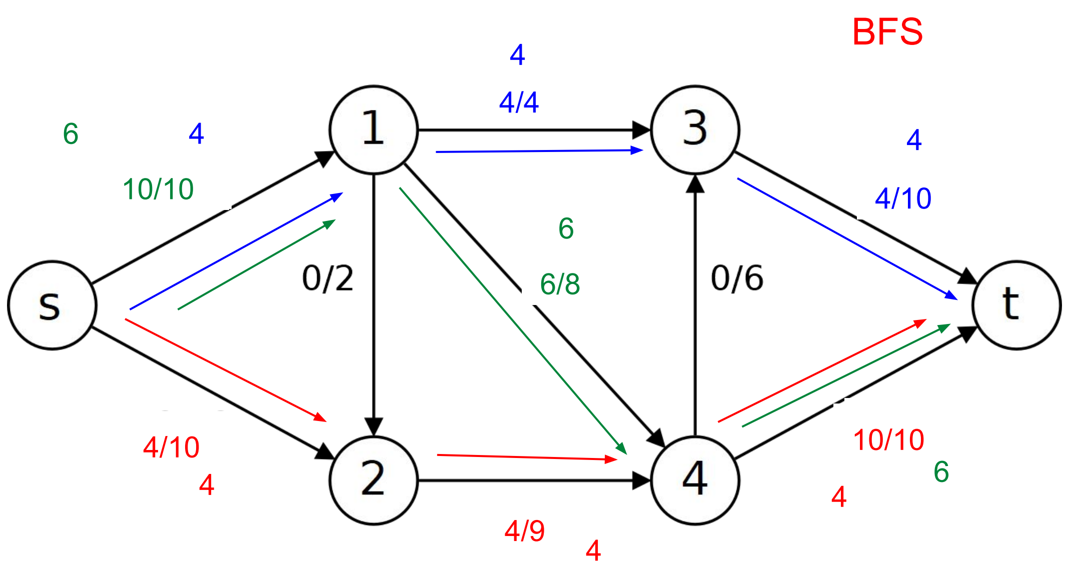
G_L : Level Graph
G_f : Residual Graph



3.2 Dinic's Algorithm - Max Flow Search - Example

1



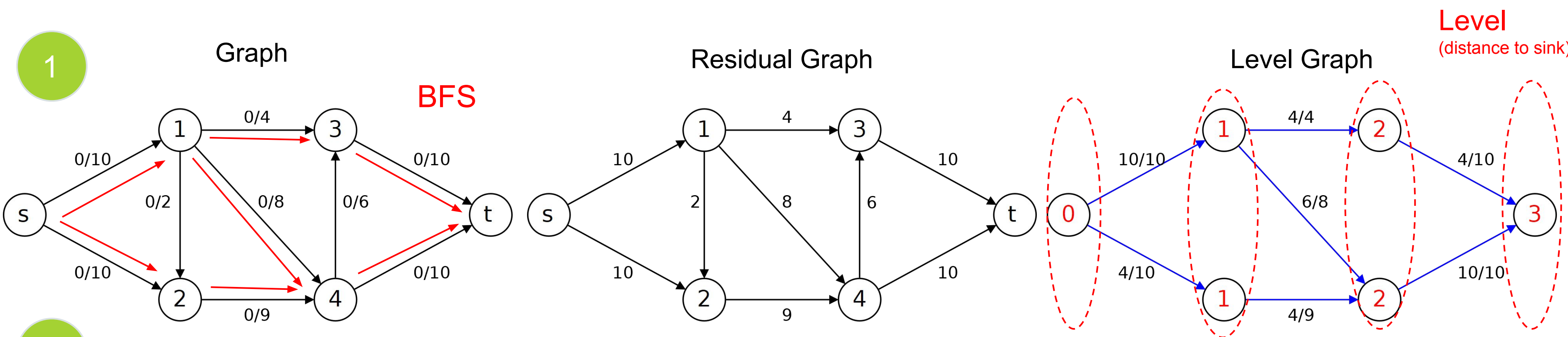


1 : min(capacities) = 4

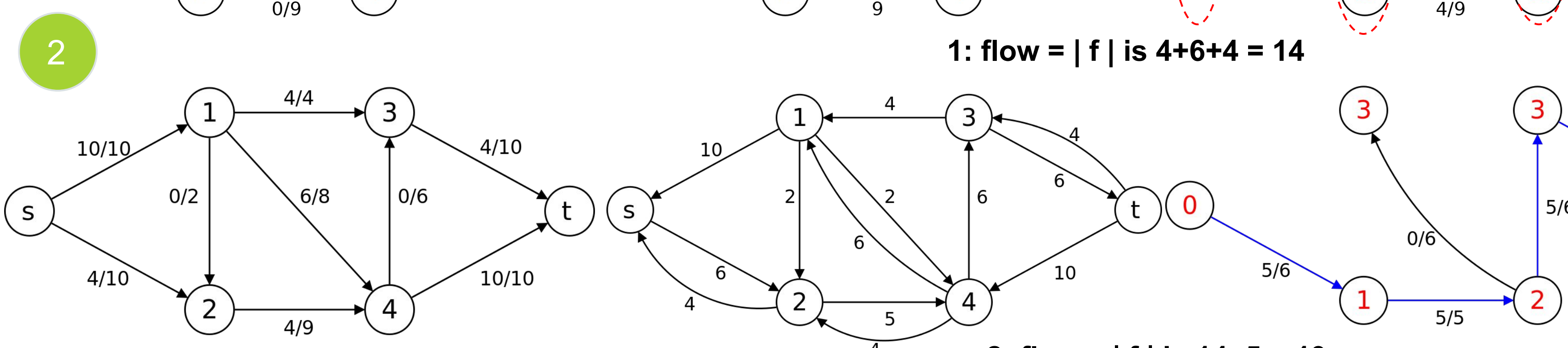
2 : min(capacities) = 6

3 : min(capacities) = 4

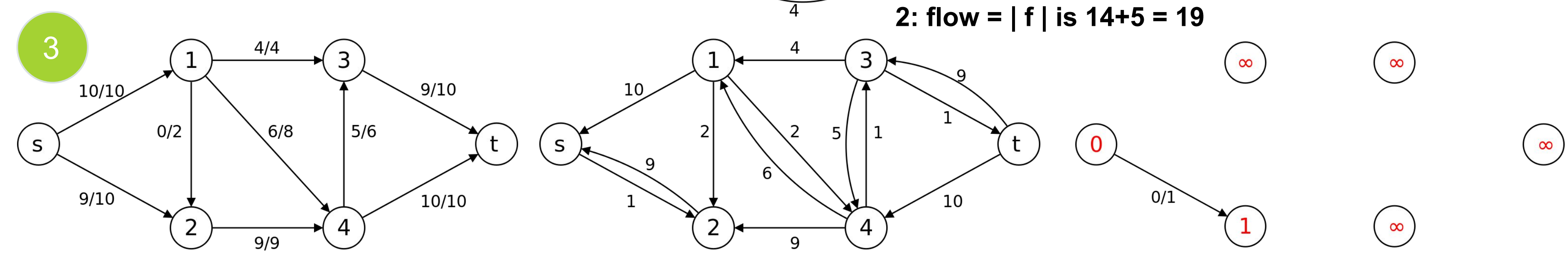
3.2 Dinic's Algorithm - Max Flow Search - Example



1: flow = | f | is 4+6+4 = 14



2: flow = | f | is 14+5 = 19

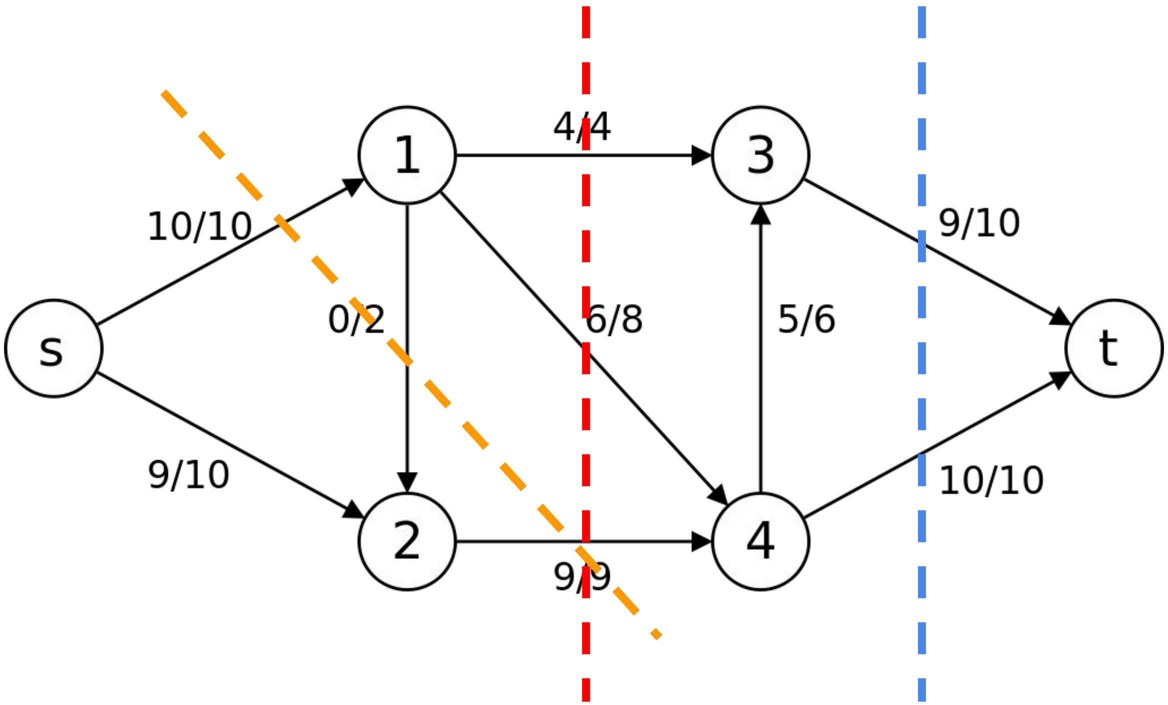


3 : flow = | f | is 19 (no path left)

Wikipedia

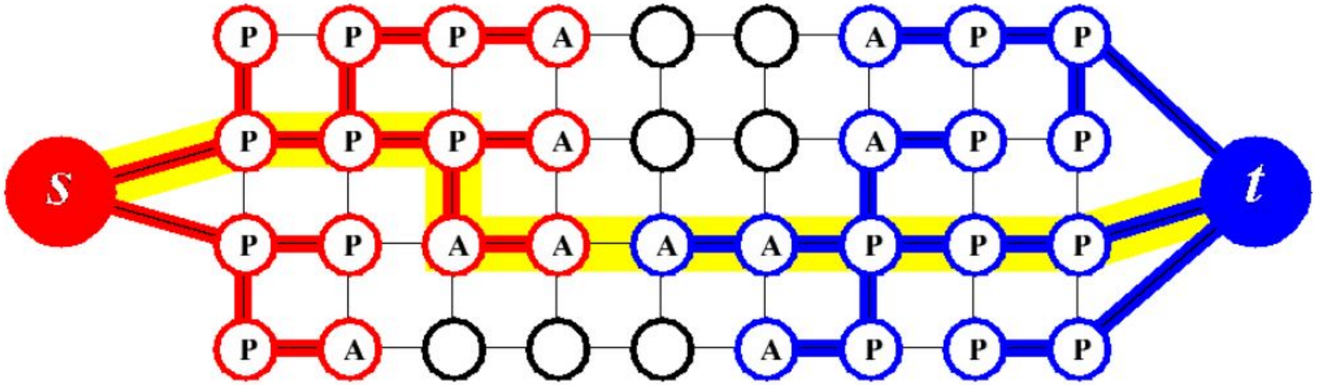
Our Case: examples of min-cut

cost of a cut =
max flow = 19



Dinic's algorithm is based on a new breadth-first search after each iteration. This search is very costly especially on large graphs computed from images with a large number of pixels.

The aim of the algorithm is to keep as much of the information we compute at each search. In the presented algorithm, we have two search trees. One for the source and one for the sink.



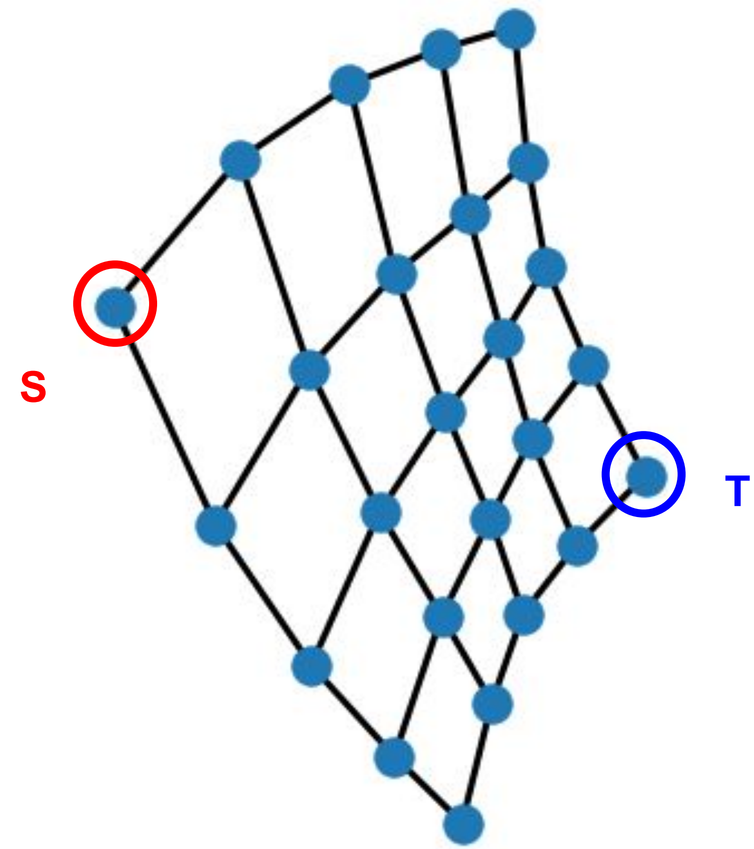
III - PROPOSED MODEL

3.3 New Min-Cut/Max-Flow Algorithm - example

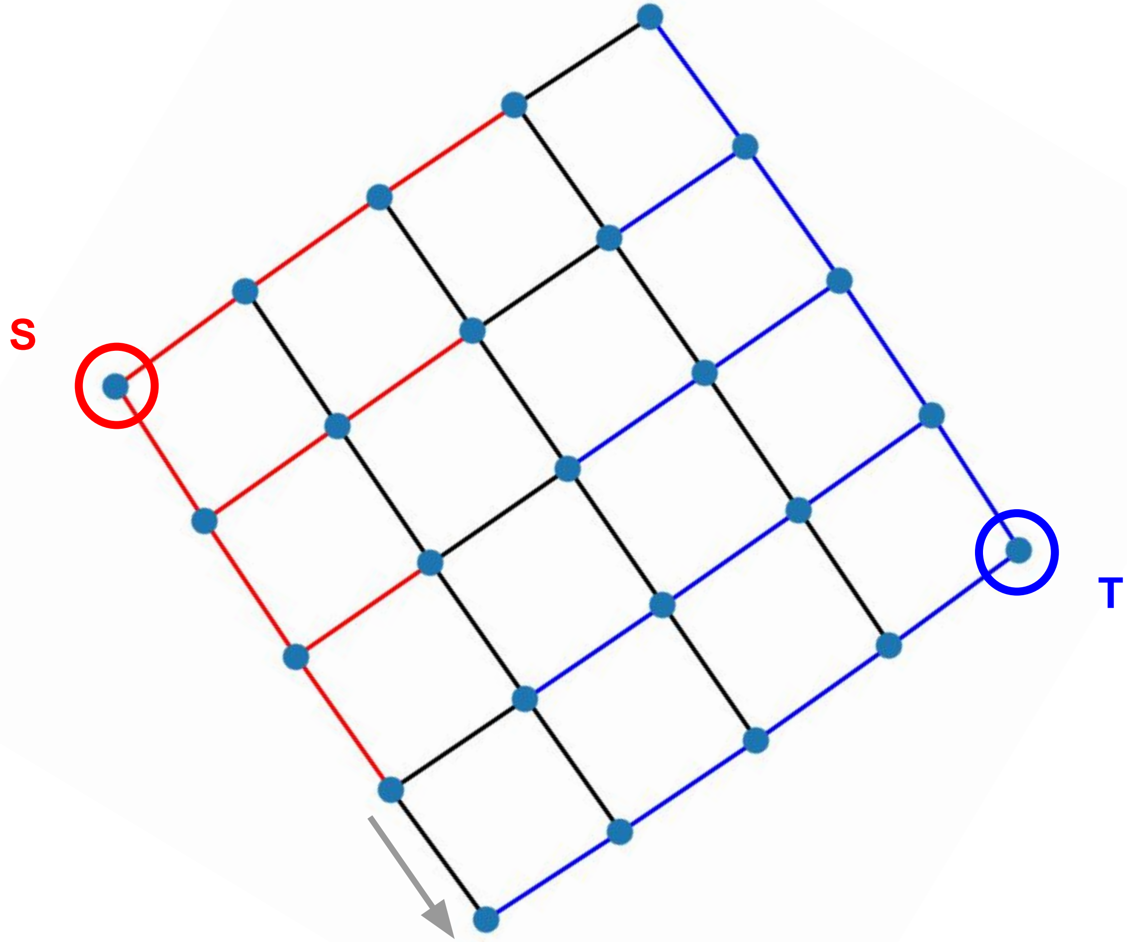
Our Algorithm is in fact the repetition of three steps:

- Growth stage
- Augmentation stage
- Adoption stage

We will illustrate these steps on the following graph:

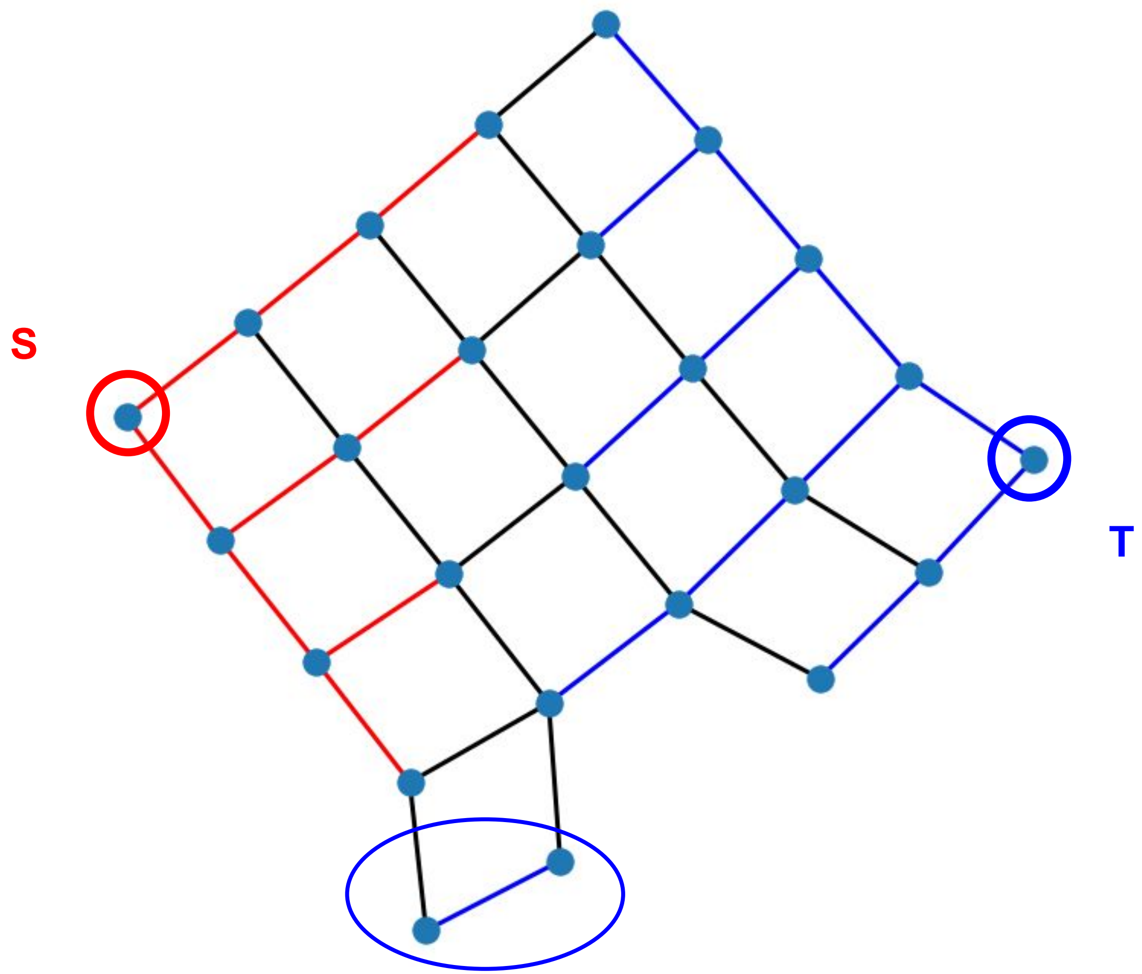


During the growth stage, both search trees grow until they touch

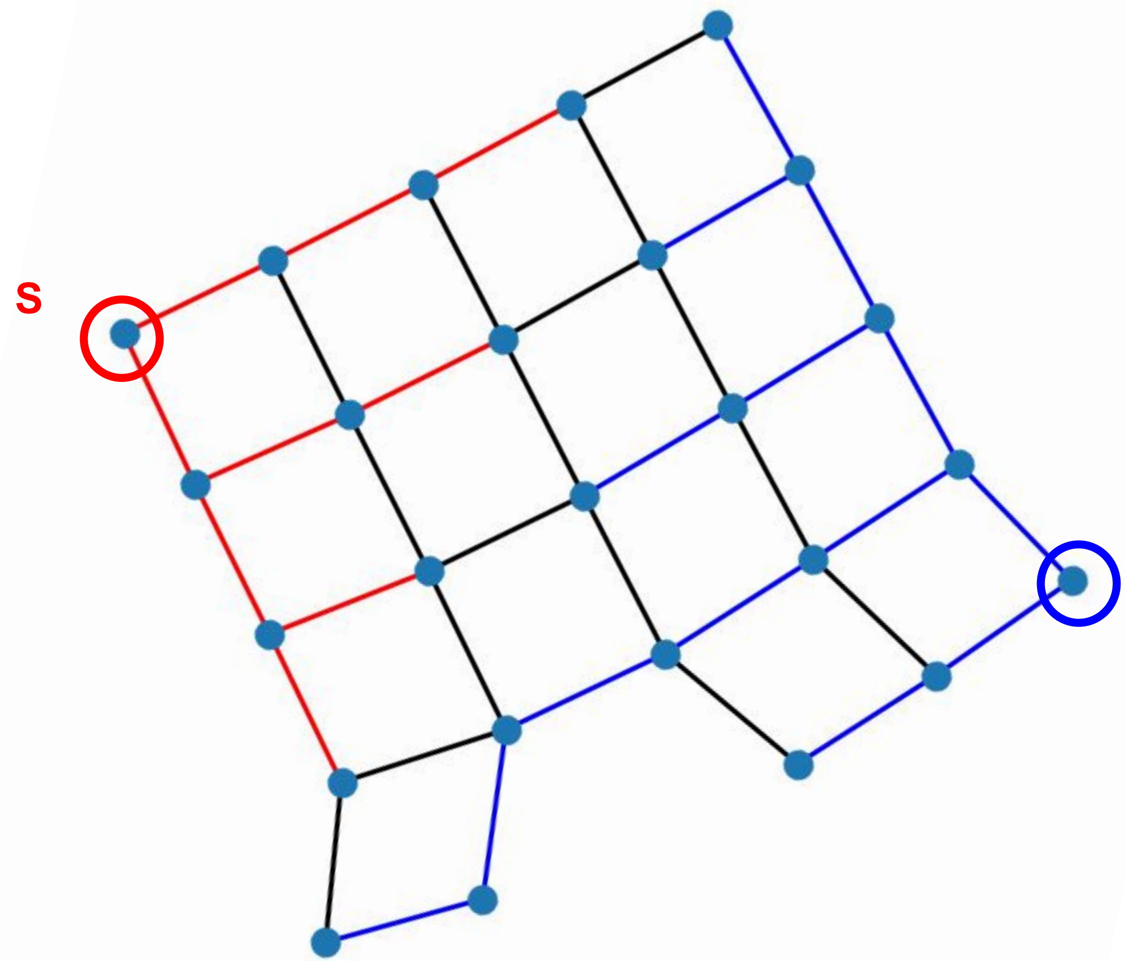


IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

During the augmentation stage, the path is augmented. This process can creates orphans.



During the adoption stage, each orphans try to reattach to the main tree. Orphans that cannot reattach are removed.



The process is repeated until no path is found during the growth stage. The resulting cut correspond to the two trees S and T.

Worst case complexity is $O(mn^2|C|)$ with m the number of edges, n the number of nodes and C the cost of minimum cut. This is worst than Dinic but in practice it is said that it is faster in average.

IV - APPLICATION TO SEGMENTATION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

We consider a **binary segmentation** problem where a given **object** has to be accurately separated from its **background** with **hard constraints**.

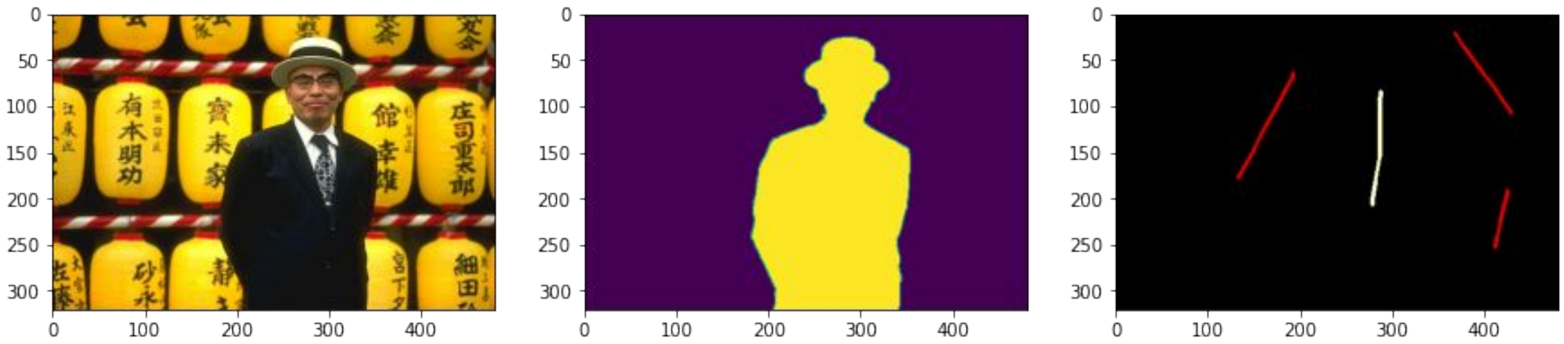
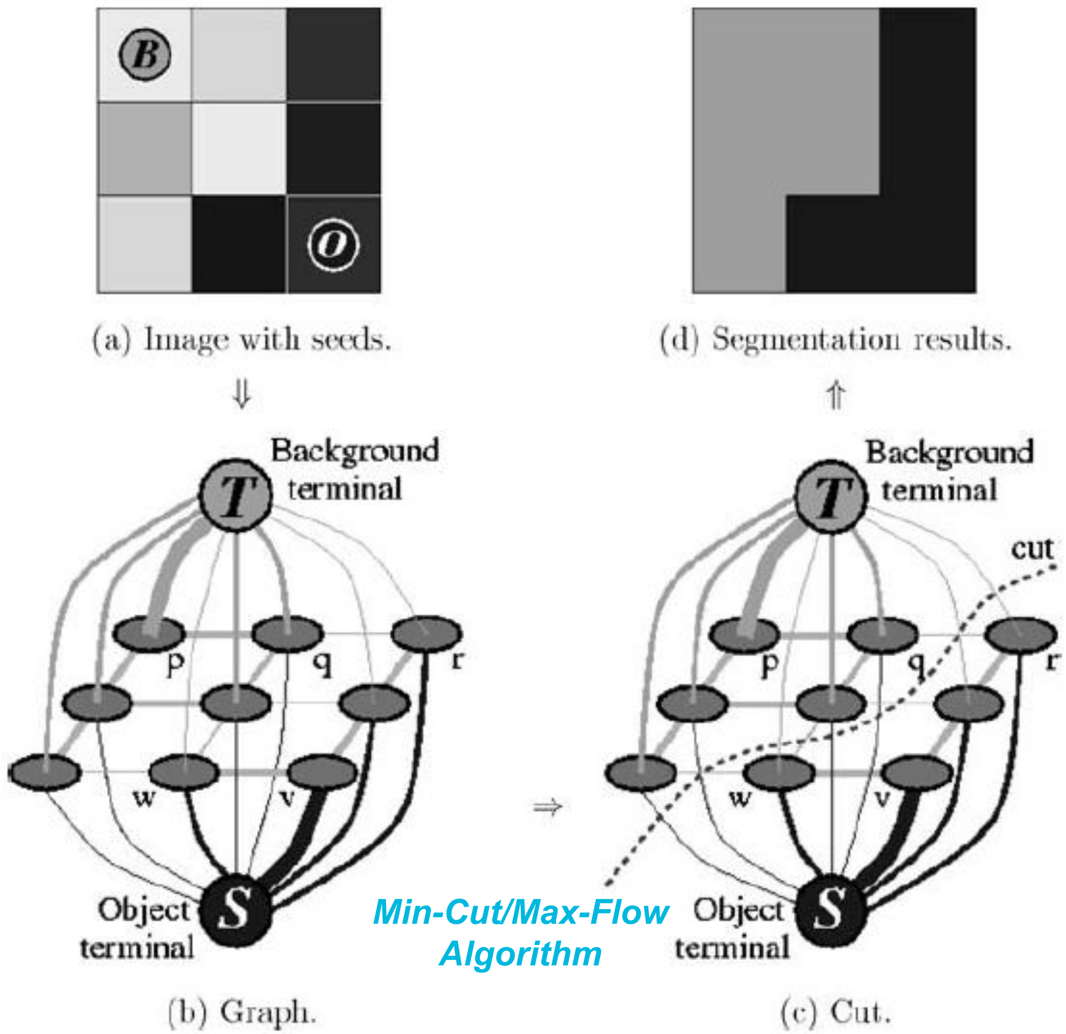


Image *Binary segmentation* *Hard constraints*

Wikipedia

Complete process



Min-Cut/Max-Flow Algorithm

Wikipedia

$$\boxed{E(A) = \lambda \cdot R(A) + B(A)}$$

Segmentation Energy

where

$$R(A) = \sum_{p \in \mathcal{P}} R_p(A_p) \quad (\textit{regional term})$$

$$B(A) = \sum_{\{p,q\} \in \mathcal{N}} B_{p,q} \cdot \delta_{A_p \neq A_q} \quad (\textit{boundary term})$$

and

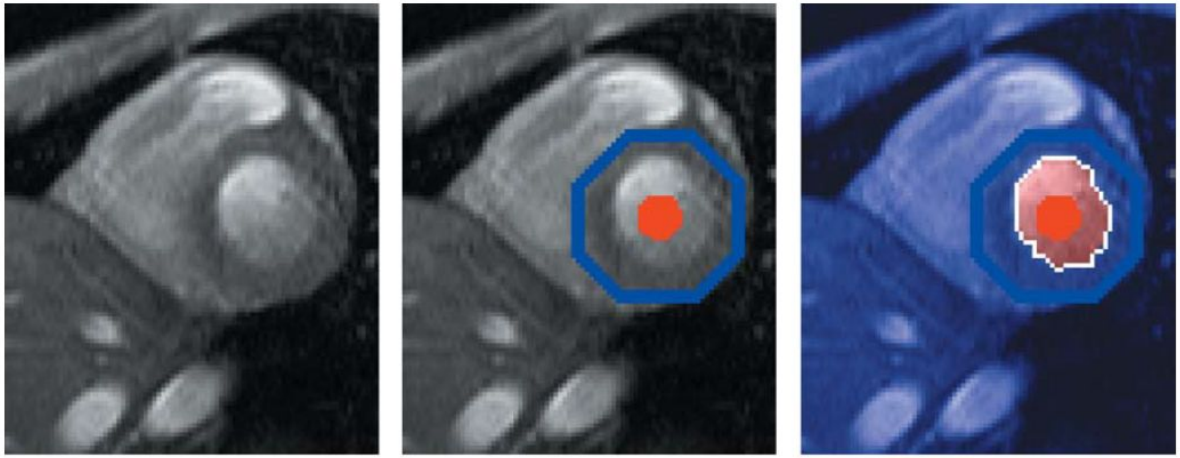
$A = (A_1, \ldots, A_p, \ldots, A_{|\mathcal{P}|})$ a binary vector whose components A_p specify assignments to pixels p in $\mathcal{P}$, either “obj” or “bkg”.

Wikipedia

Hard constraints

$$\forall p \in \mathcal{O}: \quad A_p = \text{“obj”} \tag{8}$$

$$\forall p \in \mathcal{B}: \quad A_p = \text{“bkg”}. \tag{9}$$



(a) Original image (b) Initialization (c) Segmentation

Wikipedia

Theorem 1. *The segmentation $\hat{A} = A(\hat{C})$ defined by the minimum cut $\hat{C}$ as in (10) minimizes (2) among all segmentations satisfying constraints (8, 9).*

$$E(A) = \lambda \cdot R(A) + B(A) \qquad (2)$$

Segmentation Energy

$$\forall p \in \mathcal{O}: \quad A_p = \text{“obj”} \qquad (8)$$

$$\forall p \in \mathcal{B}: \quad A_p = \text{“bkg”}. \qquad (9)$$

Hard Constraints

For any feasible cut $C \in \mathcal{F}$ we can define a unique corresponding segmentation $A(C)$ such that

$$A_p(C) = \begin{cases} \text{“obj”}, & \text{if } \{p, T\} \in C \\ \text{“bkg”}, & \text{if } \{p, S\} \in C. \end{cases} \qquad (10)$$

Nodes: $\mathcal{V} = \mathcal{P} \cup \{S, T\}$

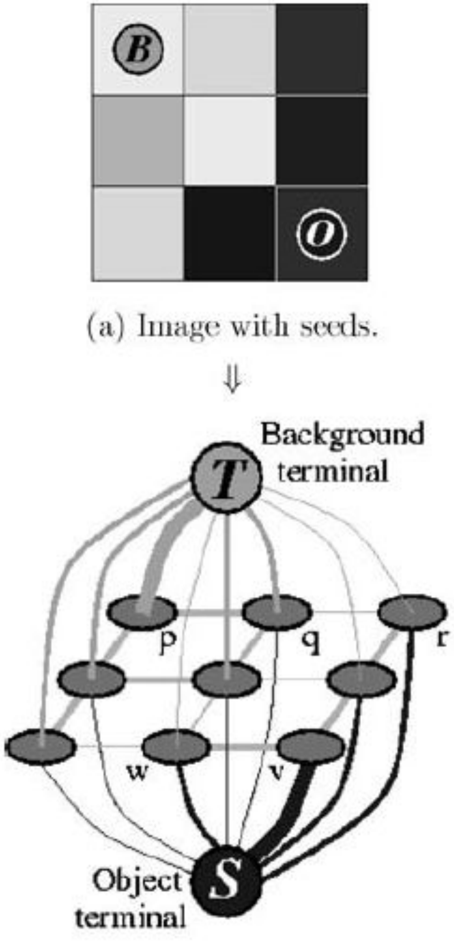
Edges: $\mathcal{E} = \mathcal{N} \bigcup_{p \in \mathcal{P}} \{\{p, S\}, \{p, T\}\}$

Edge values for the graph:

edge	weight (cost)	for
$\{p, q\}$	$B_{p,q}$	$\{p, q\} \in \mathcal{N}$
$\{p, S\}$	$\lambda \cdot R_p(\text{"bkg"})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p, T\}$	$\lambda \cdot R_p(\text{"obj"})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

where

$$K = 1 + \max_{p \in \mathcal{P}} \sum_{q:\ \{p,q\} \in \mathcal{N}} B_{p,q}.$$



(b) Graph.

Wikipedia

Examples

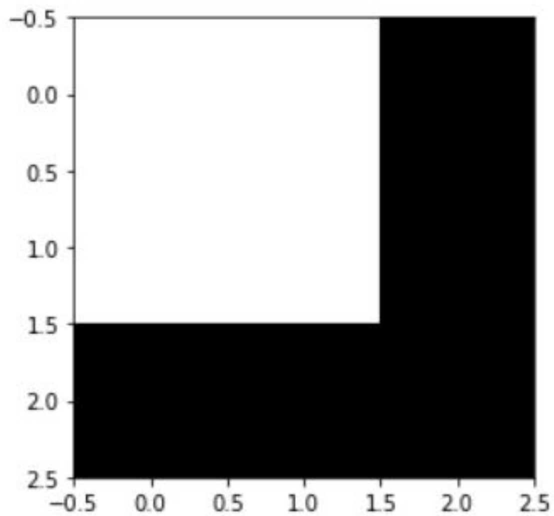
Regional term:

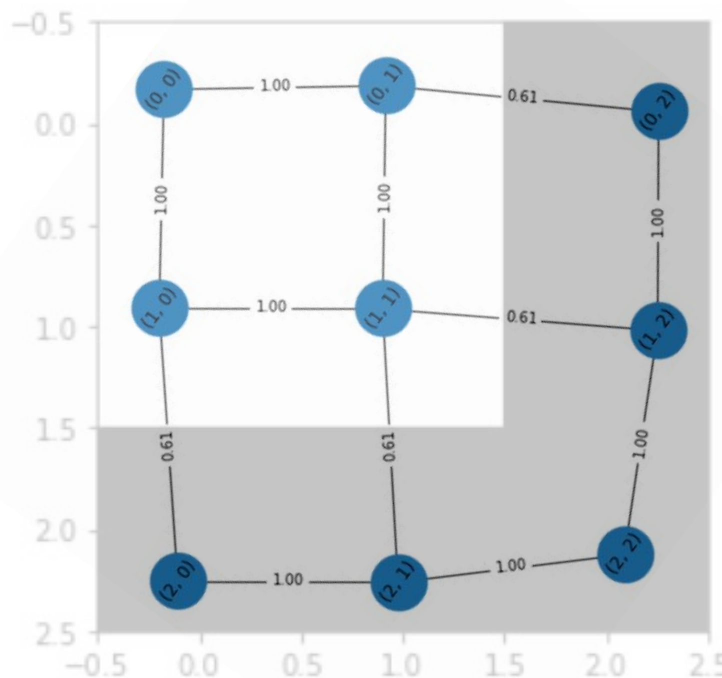
$$R_p(\text{"obj"}) = -\ln \Pr(I_p|\text{"obj"})$$
$$R_p(\text{"bkg"}) = -\ln \Pr(I_p|\text{"bkg"})$$

motivated by the MAP-MRF formulation

Boundary term:

$$B_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$$





4.7 Comparison to other algorithms

(a) Bell Photo

(b) Bell Segmentation

method	2D examples					
	Bell photo (255x313)		Lung CT (409x314)		Liver MR (511x511)	
	N4	N8	N4	N8	N4	N8
DINIC	2.73	3.99	2.91	3.45	6.33	22.86
H_PRF	1.27	1.86	1.00	1.22	1.94	2.59
Q_PRF	1.34	0.83	1.17	0.77	1.72	3.45
Our	0.09	0.17	0.22	0.33	0.20	0.45

Thank you for your attention

Any questions ?

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

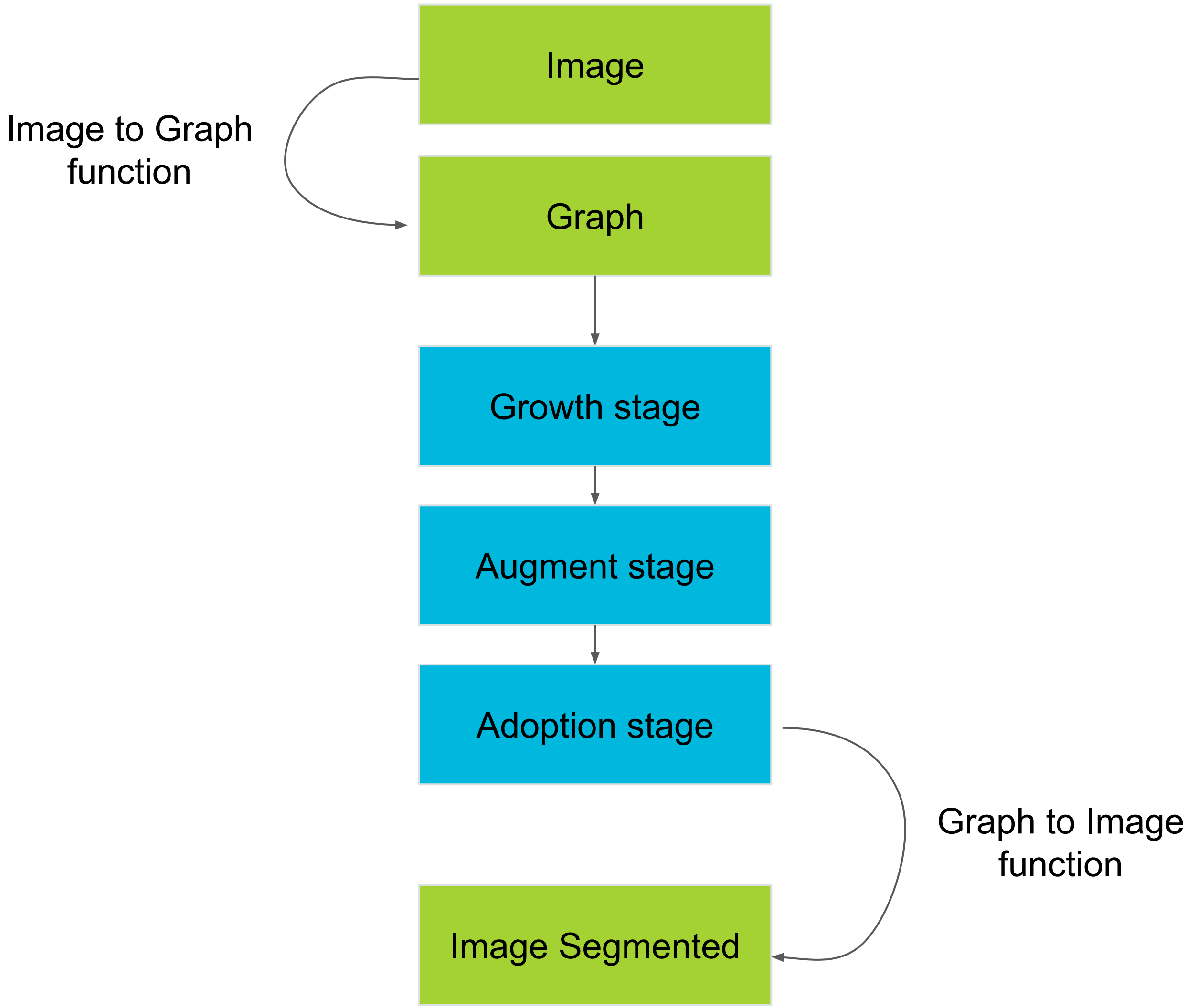
References

- Dinic’s Algorithm - Wikipedia : https://en.wikipedia.org/wiki/Dinic%27s_algorithm

3.3.1. What we will do (Group 1)

3.3.2. What we will do (Group 2)

Use of **NetworkX**



IV- EXPERIMENT

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

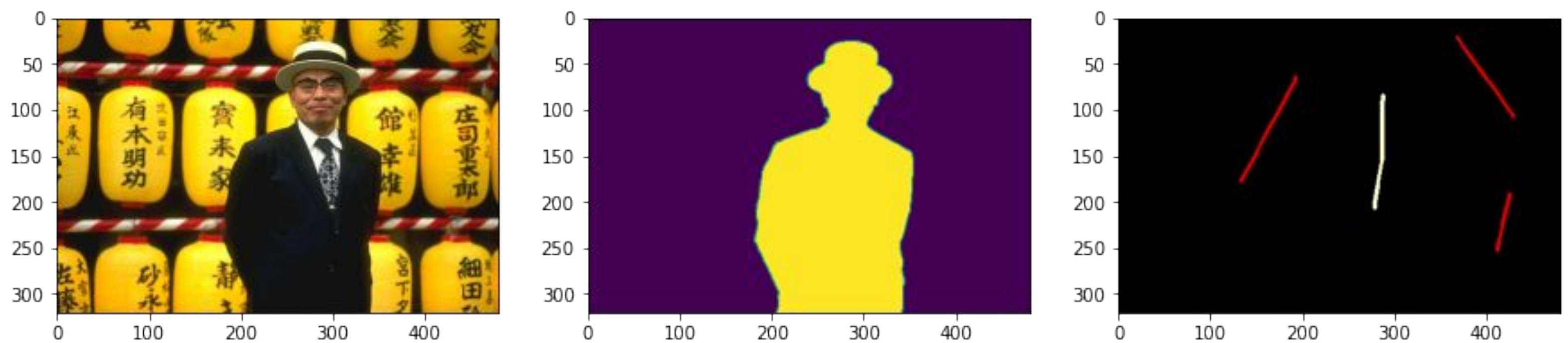
IV - EXPERIMENT

4.1 Dataset used

As most work on graph cut was done before 2010, it was very complicated to find a relevant database. The real task we are doing with our algorithm is interactive segmentation so we found a database [1] dedicated to this task.

Three set of images:

- Raw images
- Segmentation masks
- Initialisation



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

[1] - Varun Gulshan , Carsten Rother , Antonio Criminisi , Andrew Blake and Andrew Zisserman, *Geodesic Star Convexity for Interactive Image Segmentation*

4.2 Metrics used

The usual metric used to measure segmentation efficiency is the dice metric. A first experiment would be to compare the dice score of some graph-cut and other interactive segmentation algorithm with our own, using our database.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$