Delineating objects in images via minimization of ℓ_p energies; spanning forests via Dijkstra's and Kruskal's algorithms

Krzysztof Chris Ciesielski

Department of Mathematics, West Virginia University
and
MIPG, Departmentof Radiology, University of Pennsylvania

Based on a joint work with J.K. Udupa, A.X. Falcão, and P.A.V. Miranda

CCNY joint CS-Math Colloquium, May 8, 2013

Outline

1. The problem of object delineation in a digital image: translating intuition to energy minimization setup
2. Object cost as a function of object boundary; ℓ_p cost
3. Delineation algorithms associated with ℓ_p energies
4. Comparison of GC and FC image segmentations
5. Spanning forests, Dijkstra algorithm, IRFC and PW objects
6. Relation between MSF vs OPF: proof

Outline

1. The problem of object delineation in a digital image: translating intuition to energy minimization setup
2. Object cost as a function of object boundary; ℓ_p cost
3. Delineation algorithms associated with ℓ_p energies
4. Comparison of GC and FC image segmentations
5. Spanning forests, Dijkstra algorithm, IRFC and PW objects
6. Relation between MSF vs OPF: proof

Example 1, 2D, of object segmentation/delineation

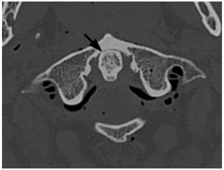
An image of peppers

Delineation version 1

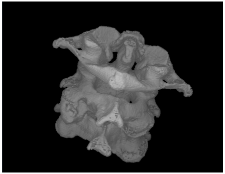
Delineation version 2

Delineation version 3

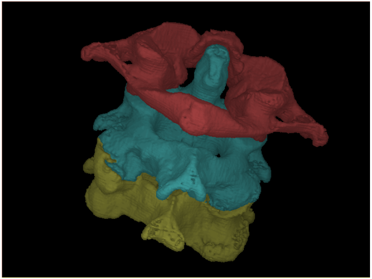
Example 2, 3D: a CT image of patient's cervical spine



A slice of an original 3D image

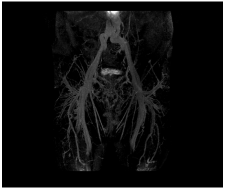


Surface rendition of segmented three vertebrae, together



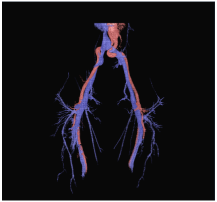
Color surface rendition of the segmented three vertebra

Example 3: An MR angiography image of the body region from belly to knee.



Rendition of an original 3D, contrast enhanced, image

A surface rendition of the entire vascular tree



Color surface rendition of segmented arterial (red) and veinous (blue) trees

Image segmentation — formal setting

- An *(n-dimensional) image* is a map f from $C \subset \mathbb{R}^n$ into $\mathbb{R}^k$

  The value $f(c)$ represents image intensity at c, a k-dimensional vector each component of which indicates a measure of some aspect of the signal, like color.

- *Segmentation problem:* Given an image $f\colon C \to \mathbb{R}^k$,

  find a "desired" family $\mathcal{S}(f) = \{P_1, \ldots, P_m\}$ of subsets of C.

- *Delineation problem* (on which we concentrate)

  is when $m = 1$, i.e., when $\mathcal{S}(f) = P \subset C$.

Delineation of an image $f\colon C \to \mathbb{R}^k$ — formal setting

How to express that $\mathcal{S}(f) = P \subset C$ is desired?

There is no magic formulation that expresses all desires!

Several practical “solutions” exist. We use here the following:

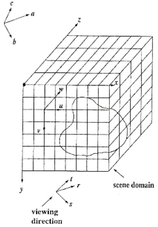
- Fix seed sets:
  S indicating the foreground object P, (i.e., $S \subset P$), and
  T indicating the background (i.e., $T \cap P = \emptyset$).

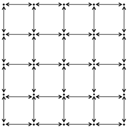
  This restricts the search space for $\mathcal{S}(f)$ to the family
  $\mathcal{P}(S,T) = \{P \subset C \setminus T \colon S \subset P\}$
- Define an energy/cost function $\varepsilon\colon \mathcal{P}(\emptyset,\emptyset) \to [0,\infty)$
- Declare $\mathcal{S}(f)$ to be desired when it minimizes ε on $\mathcal{P}(S,T)$.

Digital vs "continuous" image $f\colon C \to \mathbb{R}^k$

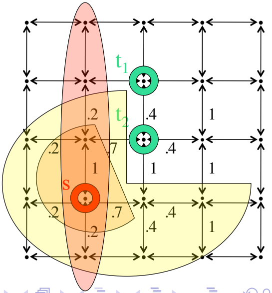
The above set-up makes sense and was studied for the images with scene $C \subset \mathbb{R}^n$ being open bounded region.

We discuss only digital images, with finite rectangular scenes:





Example of sets in $\mathcal{P}(S, T)$
with $S = \{s\}$, $T = \{t_1, t_2\}$

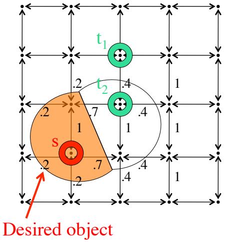


Outline

Heuristic and the definition of boundary

Heuristic: The objects boundary areas should be identifiable in the image, as the areas of sharp image intensity change.

What constitutes boundary $\mathrm{bd}(P)$ of P?



Need graph (or topological) structure $G = \langle V, E \rangle$ on C:

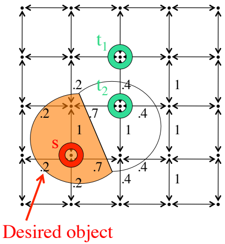
- Pixels $c \in C$ are its vertices, $V = C$;
- Edges $\{c, d\} \in E$ are "nearby" vertices (e.g. as in figure).

$\mathrm{bd}(P)$ is the set of all edges $\{c, d\} \in E$ with $c \in P$ and $d \notin P$

Weighted graphs and ℓ_p cost functions, $1 \leq p \leq \infty$

Assume that with every edge $e = \{c, d\} \in E$ of an image f we have associated its weight/cost $w(e) \geq 0$, which is low, for big $\|f(c) - f(d)\|$.

Typically, $w(e) = e^{-\|f(c)-f(d)\|/\sigma^2}$, see fig.



If $F_P\colon E \to [0, \infty)$, $F_P(e) = w(e)$ for $e \in \mathrm{bd}(P)$ and $F_P(e) = 0$ for $e \notin \mathrm{bd}(P)$, then ℓ_p cost is defined as

$$\varepsilon_p(P) \stackrel{\text{def}}{=} \|F_P\|_p = \begin{cases} \left(\sum_{e \in \mathrm{bd}(P)} w(e)^p\right)^{1/p} & \text{if } p < \infty \\ \max_{e \in \mathrm{bd}(P)} w(e) & \text{if } p = \infty. \end{cases}$$

Outline

1. The problem of object delineation in a digital image: translating intuition to energy minimization setup
2. Object cost as a function of object boundary; ℓ_p cost
3. Delineation algorithms associated with ℓ_p energies
4. Comparison of GC and FC image segmentations
5. Spanning forests, Dijkstra algorithm, IRFC and PW objects
6. Relation between MSF vs OPF: proof

FC and GC algorithms as minimizers of ε_p

$$\varepsilon_p(P) \stackrel{\text{def}}{=} \|F_P\|_p = \begin{cases} \left(\sum_{e \in \text{bd}(P)} w(e)^p\right)^{1/p} & \text{if } p < \infty \\ \max_{e \in \text{bd}(P)} w(e) & \text{if } p = \infty. \end{cases}$$

$p = 1$: $\varepsilon_1(P) = \sum_{e \in \text{bd}(P)} w(e)$;

Optimization solved by classic min-cut/max-flow algorithm.

Graph Cut, GC, delineation algorithm optimizes ε_1.

$p = \infty$: $\varepsilon_\infty(P) = \max_{e \in \text{bd}(P)} w(e)$;
Optimization solved by (versions of) Dijkstra algorithm.

ε_∞ optimized objects are returned by the algorithms: Relative Fuzzy Connectedness, RFC, Iterative RFC, IRFC, and Power Watershed, PW [C. Couprie *et al*, 2011].

$p = 2$: related to Random Walker, RW, algorithm [Grady, 2006], see next slides.

Fuzzy sets

A map $x\colon C \to [0,1]$ (i.e., $x \in [0,1]^C$) can be considered as a *fuzzy set*, with $x(c)$ giving the degree of membership of c in it.

A hard set $P \subset C$ is identified with a fuzzy set (binary image) $\chi_P \in \{0,1\}^C \subset [0,1]^C$, $\chi_P(c) = 1$ iff $c \in P$.

For $x \in [0,1]^C$ let $\hat{\varepsilon}_p(x) = \|F_x\|_p$, where $F_x\colon E \to [0,\infty)$,

$F_x(\{c,d\}) = |x(c) - x(d)|w(\{c,d\})$ for $\{c,d\} \in E$.

Then $\varepsilon_p(P) = \hat{\varepsilon}_p(\chi_P)$. We can minimize $\hat{\varepsilon}_p$ on

$\hat{\mathcal{P}}(S,T) = \{x\colon x(c) = 1 \text{ for } c \in S \;\&\; x(c) = 0 \text{ for } c \in T\}$

instead of ε_p on $\mathcal{P}(S,T) = \hat{\mathcal{P}}(S,T) \cap \{0,1\}^C$.

Random Walker, RW, algorithm

- RW finds (the unique) $\hat{\varepsilon}_2$ minimizer on $\hat{\mathcal{P}}(S, T)$.
- Defines its output as $P = \{c \colon x(c) \geq .5\}$.

Problems with RW:

1. Output need not be connected (even when S and T are).
2. P need not minimize ε_2 on $\mathcal{P}(S, T)$.

Neither of this happens for ε_1 (i.e. GC) or ε_∞ (i.e. RFC or PW):

Thm: For $p \in \{1, \infty\}$, any minimizer of $\hat{\varepsilon}_p$ on $\hat{\mathcal{P}}(S, T)$ actually belongs to $\mathcal{P}(S, T)$.

(Non)-uniquness of the minimizers for ε_1 and ε_∞

Let $\mathcal{P}_p(S,T) = \{P \in \mathcal{P}(S,T) \colon P \text{ minimizes } \varepsilon_p \text{ on } \mathcal{P}(S,T)\}$.

Both $\mathcal{P}_1(S,T)$ and $\mathcal{P}_\infty(S,T)$ may have more than one element.

However, the outputs of the standard versions of the algorithms:

- GC, from $\mathcal{P}_1(S,T)$,
- RFC, from $\mathcal{P}_\infty(S,T)$, and
- IRFC, from $\mathcal{P}_\infty(S,T)$

are unique in the sense of the next theorem.

GC & FC segmentations — comparison theorem 1

Theorem (Argument minimality)

For $p \in \{1, \infty\}$, $\mathcal{P}_\varepsilon(S, T)$ contains the $\subset$-smallest object.

- *GC algorithm returns the smallest set in $\mathcal{P}_1(S, T)$.*
- *RFC algorithm returns the smallest set in $\mathcal{P}_\infty(S, T)$.*
- *IRFC algorithm returns the smallest set in a refinement $\mathcal{P}^*_\infty(S, T)$ of $\mathcal{P}_\infty(S, T)$.*

Moreover, if n is the size of the image (scene), then

- *GC runs in time of order $O(n^3)$ (the best known algorithm) or $O(n^{2.5})$ (the fastest currently known algorithm)*
- *Both RFC and IRFC run in time of order $O(n)$ (for standard medical images — the intensity range size not too big) or $O(n \ln n)$ (the worst case scenario)*

GC & FC — asymptotic equivalence

Theorem (Asymptotic equivalence of GC and FC)

Let $\mathcal{P}_p^m(S,T)$ be the family $\mathcal{P}_p(S,T)$ for the edge weight function w replaced by its m-th power w^m. Then

- $\mathcal{P}_\infty^m(S,T) = \mathcal{P}_\infty(S,T)$ *and similarly for IRFC algorithm.*

  So, the outputs of RFC and IRFC are unchanged by m.
- $\mathcal{P}_1^m(S,T) \subseteq \mathcal{P}_\infty(S,T)$ *for m large enough.*

In particular, if $\mathcal{P}_\infty(S,T)$ has only one element, then

the output of GC coincides with the outputs of RFC and IRFC

for m large enough.

Outline

1. The problem of object delineation in a digital image: translating intuition to energy minimization setup
2. Object cost as a function of object boundary; ℓ_p cost
3. Delineation algorithms associated with ℓ_p energies
4. Comparison of GC and FC image segmentations
5. Spanning forests, Dijkstra algorithm, IRFC and PW objects
6. Relation between MSF vs OPF: proof

Advantages of FC over GC — theoretical angle

Speed: FC algorithms run a lot faster than GC algorithms: $O(n)$ (or $O(n \ln n)$) versus $O(n^3)$ (or $O(n^{2.5})$).

Robustness: RFC & IRFC are unaffected by small seed changes. GC is sensitive for even small seed changes.

Shrinking: GC chooses objects with small size boundary (often with edges with high weights); No such problem for RFC & IRFC

Multiple objects: FC framework handles easily the segmentation of multiple objects, same running time and robustness. GC in such setting leads to NP-hard problem, so (for precise delineation) it runs in exponential time

Iterative approach: RFC has an iterative approach refinement; No such refinement methods exist for GC at present.

Advantages of GC over FC

Boundary smoothness: GC chooses small boudary, so it naturally smooths it; in many (but not all) medically important delineations, this is a desirable feature.

Basic FC framework has no boundary smoothing; if desirable, smoothing requires post processing

Combining image homogeneity info with known object intensity: GC naturally combines information on image homogeneity (binary relation on voxels) with information on expected object intensity (unary relation on voxels);

Combining such informations is difficult to achieve in the FC framework.

Setup of experiments:

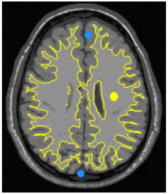
- In each experiment we used 20 MR BrainWeb phantom images (simulated T1 acquisition); graphs show averages.
- Sets of seeds were generated, from known true binary segmentations, by applying erosion operation:
  the bigger erosion radius, the smaller the seed sets.
- The weight map $w(c,d)$, same for FC and GC, was defined from the image intensity function f as
  $w(c,d) = -|G(f(c)) - G(f(d))|$,
  where G is an appropriate Gaussian.

Setup of experiments:

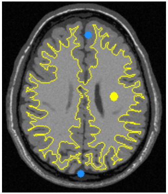
Data parameters: the simulated T1 acquisition were as follows: spoiled FLASH sequence with TR=22ms and TE=9.2ms, flip angle = 30°, voxel size $= 1 \times 1 \times 1\text{mm}^3$, noise = 3%, and background non-uniformity = 20%.

Computer: Experiments were run on PC with an AMD Athlon 64 X2 Dual-Core Processor TK-57, 1.9 GHz, 2×256 KB L2 cache, and 2 GB DDR2 of RAM.

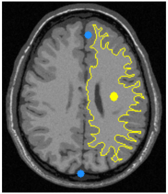
Robustness & shrinking for FC & GC: White Matter



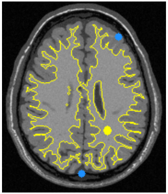
(a) RFC



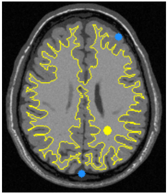
(b) IRFC



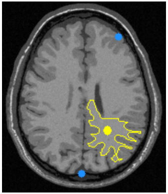
(c) GC



(d) RFC



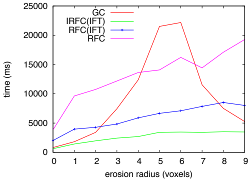
(e) IRFC



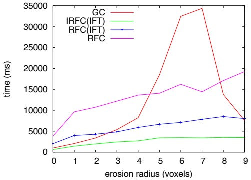
(f) GC

Figure: (a)&(d) and (b)&(e): same outputs for different seeds; (c)&(f) GC: dramatic change of output; seeds choice same as in the FC case

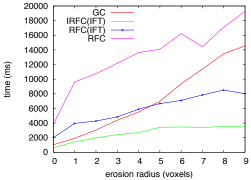
Time & accuracy of FC & GC: segmentation of WM



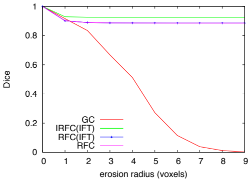
(a) Time for w^1



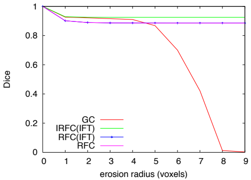
(b) Time for w^5



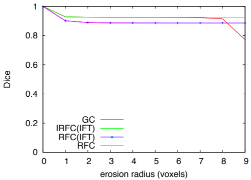
(c) Time for w^{30}



(d) Accuracy for w^1



(e) Accuracy for w^5



(f) Accuracy for w^{30}

FC vs GC: Conclusions

- FC and GC quite similar,
  yet FC has many advantages over GC:
  - FC runs considerably faster than GC
  - FC is robust (seed), while GC has shrinkage problem
  - FC, unlike GC, easily handles multiple-object segmentation
- unless the application requires, in an essential way, the simultaneous use of
  - homogeneity (binary) info on image intensity;
  - expected object intensity (unary) info on image intensity;

  it makes sense to use FC (more precisely IRFC) segmentation algorithm, rather than GC algorithm

Outline

Forests: the powerhouse behind Dijkstra algorithm

Fix weighted graph $G = \langle C, E, w \rangle$ and $\emptyset \neq W \subset C$.

Definition (Spanning Forest w.r.t. W)

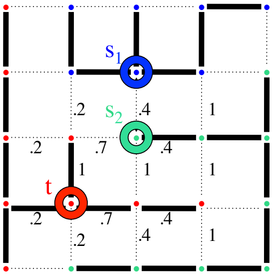
A *forest* for G is any subgraph $\mathbb{F} = \langle C, E' \rangle$ of G free of cycles. $\mathbb{F} = \langle C, E' \rangle$ is *spanning with respect to W* when any connected component of $\mathbb{F}$ contains precisely one element of W.

Example of a spanning

forest w.r.t. $W = \{s_1, s_2, t\}$

Each component

marked by different color



Forest-generated (IRFC and PW) objects

$G = \langle C, E, w \rangle$ – weighted graph, $\emptyset \neq W \subset C$, $S \subset W$

Definition (Forest-generated object)

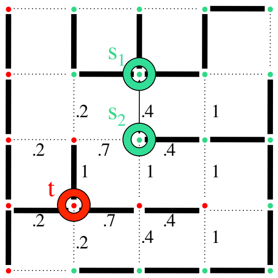
For a spanning forest $\mathbb{F}$ w.r.t. W and $S \subset W$,
$P(S, \mathbb{F})$ is a union of all components of $\mathbb{F}$ intersecting S.
Note that $P(S, \mathbb{F}) \in \mathcal{P}(S, T)$ for $T = W \setminus S$.

Example (green vertices) of

$P(S, \mathbb{F})$ with $S = \{s_1, s_2\}$.

Outputs of the algorithms we will discuss, GC_{sum} and PW,

are in the $P(S, \mathbb{F})$ format.

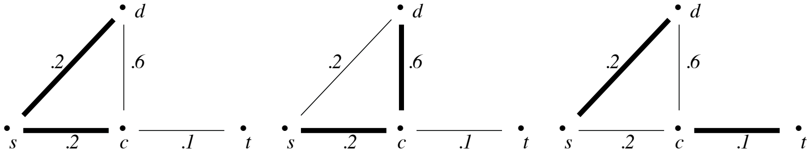


Optimal Path Forest, OPF

Definition (Optimal Path Forest, OPF)

For a path $p = \langle c_1, \ldots, c_k \rangle$ in G let $\mu(p) = \min_{i<k} w(\{c_k, c_{k+1}\})$, the *weakest link* of p.
A forest $\mathbb{F}$ w.r.t. W is *path-optimal* provided for every $c \in C$, the unique path p_c in $\mathbb{F}$ from W to c is μ-optimal in G, i.e., $\mu(p_c) \geq \mu(p)$ for any path p in G from W to c.

For OPF $\mathbb{F}$ w.r.t. W, $\mu(p_c) = \mu^C(c, W)$ for every $c \in C$
(with μ^C in the Fuzzy Connectedness sense)



(g) OPF, $W = \{s, t\}$ (h) another OPF (i) not OPF

$\text{GC}_{\max}$ algorithm and IRFC

Theorem ([KC *et al.*] OPF object minimizing ε_∞)

There exists the smallest $P_{\min} \in \mathcal{P}(S, T)$ in form $P(S, \mathbb{F})$, where $\mathbb{F}$ is an OPF w.r.t. $S \cup T$.

$\mathbb{F}$ is found by $\text{GC}_{\max}$, a version of Dijkstra's shortest path algorithm, in a linear time w.r.t. $|C| + M$, where M is the size of the range of w.

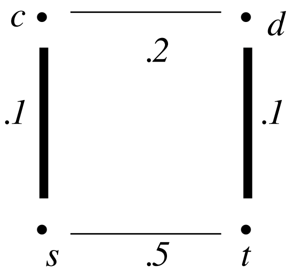
In practice, $O(|C| + M) = O(|C|)$.

The object $P_{\min}$, returned by $\text{GC}_{\max}$, coincides with the Iterative Relative Fuzzy Connectedness, IRFC, object.

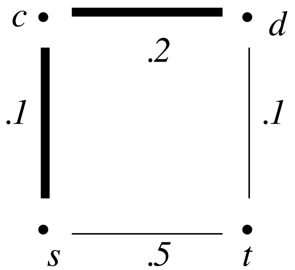
Maximal Spanning Forest, MSF

Definition (Maximal Spanning Forest, MSF)

A forest $\mathbb{F} = \langle C, E' \rangle$ w.r.t. W is *maximal spanning* provided $\sum_{e \in E'} w(e) \geq \sum_{e \in \hat{E}'} w(e)$ for every forest $\hat{\mathbb{F}} = \langle C, \hat{E}' \rangle$ w.r.t. W



(j) OPF w.r.t. $\{s, t\}$, not MSF



(k) MSF and OPF

Theorem ([Audigier & Lotufo], [Cousty *et al.*])

Every MSF is OPF, but not the other way around.

MSF and Power Watershed, PW, algorithm

Theorem ([C. Couprie *et al.*] PW output as MSF)

PW algorithm returns $P(S, \mathbb{F})$ for a MSF $\mathbb{F}$ w.r.t. $S \cup T$.

$\mathbb{F}$ is found by PW via a complicated version of Kruskal's algorithm and, locally, Random Walker algorithm.

Since

- IRFC object is indicated by OPF,
- PW object is indicated by MSF, and
- every MSF is OPF

What is the relation between IRFC and PW objects?

New results on GC_{max}, MSF, and OPF

Theorem ([KC *et al.*] MSF vs OPF)

If $P_{\min}$ is the output of $GC_{\max}$ (the smallest $P(S,\mathbb{F})$, with with $\mathbb{F}$ is being OPF w.r.t. $S \cup T$), then $P_{\min} = P(S,\hat{\mathbb{F}})$ for some MSF $\hat{\mathbb{F}}$.

If $\mathbb{F}$ is a MSF w.r.t. $S \cup T$, then $P(S,\mathbb{F})$ minimizes energy ε_∞ (in $\mathcal{P}(S,T)$).

$P(S,\mathbb{F})$, with $\mathbb{F}$ being OPF w.r.t. $S \cup T$, need not minimize ε_∞.

In other words

$$P_{\min} \in \mathcal{P}_{MSF}(S,T) \subset \mathcal{P}_{OPF}(S,T) \cap \mathcal{P}_{\varepsilon_\infty}(S,T),$$

where $\mathcal{P}_{MSF}(S,T) = \{P(S,\mathbb{F}) \colon \mathbb{F} \text{ is MSF}\}$, similarly for OPF, and $\mathcal{P}_{\varepsilon_\infty}(S,T)$ is the set of all ε_∞-optimizers.

Outline

1. The problem of object delineation in a digital image: translating intuition to energy minimization setup
2. Object cost as a function of object boundary; ℓ_p cost
3. Delineation algorithms associated with ℓ_p energies
4. Comparison of GC and FC image segmentations
5. Spanning forests, Dijkstra algorithm, IRFC and PW objects
6. Relation between MSF vs OPF: proof

Outline of the proof of Main Theorem

- Describe Dijkstra's algorithm that gives OPF $\mathbb{F}$ with $P_{\min} = \mathcal{P}(S, \mathbb{F})$. Notice, it is the smallest set in $\mathcal{P}_{OPF}(S, T)$.
- Use Kruskal's algorithm to find MSF $\hat{\mathbb{F}}$ with $P_{\min} = \mathcal{P}(S, \hat{\mathbb{F}})$.
- Show that $\mathcal{P}(S, \hat{\mathbb{F}}) \in \mathcal{P}_{\varepsilon_\infty}(S, T)$ whenever $\hat{\mathbb{F}}$ is MSF. An argument is a variant of a proof that Kruskal's algorithm indeed returns MSF.
- Give examples, showing that no inclusion can be reversed.

Dijkstra's algorithm DA: standard vs our version

$G = \langle C, E, w \rangle$, $\mathbb{F}$ generated forest w.r.t. W, $S \subset W \subset C$
p_c – unique path in $\mathbb{F}$ from W to $c \in C$

- Standard DA "grows" tree from a single source set W.
  We use DA to grow forest with a multiple sources set W.
- In standard DA, path p_c has the smallest length.
  (It optimizes path measure "sum of weights of all links.")
  We use DA to optimize p_c w.r.t. "weakest link measure" μ.
- Newest variation:
  We insure that $P_{\min} = P(\mathbb{F}, S)$ is the smallest possible.
  No control of algorithm's output among $\mathcal{P}_{\varepsilon_\infty}(S, T)$ was insurable before introduction of $GC_{\max}$ (as far as we know).

GC_{max} (i.e., our DA) data structure

- $\mathbb{F}$ is grown from roots, $W = S \cup T$, via adding edges.
- $\mathbb{F}$ is indicated via path-predecessor map Pr: $Pr[W] = \{\emptyset\}$, $Pr(c) =$ *predecessor of c in* p_c for $c \notin W$
- $R(c)$ indicates root of c: the initial $w \in W$ belonging to p_c
- We use preorder relation $\prec$ on $\mathbb{R} \times C$:

$$\langle x, c\rangle \prec \langle y, d\rangle \iff x < y \text{ or } (x = y \,\&\, d \in T \,\&\, c \notin T)$$

- Initialize $\mu(c) = 1$, $R(c) = c$, $Pr(c) = \emptyset$ for $c \in W$
- Initialize $\mu(c) = -1$, $R(c) = c$, $Pr(c) = c$ for $c \in C \setminus W$
- Insert every $c \in C$ into queue Q according to priority $\preceq$

The $\mathrm{GC}_{\max}$ algorithm

begin
1. *while* Q is not empty *do*
2. remove from Q a $\preceq$-maximal spel c;
3. *for* every d with $\{c, d\} \in E$ *do*
4. *if* $\langle \mu(d), R(d) \rangle \prec \langle \min\{\mu(c), w_{\{c,d\}}\}, R(c) \rangle$ *then*
5. set $\mu(d) = \min\{\mu(c), w_{\{c,d\}}\}$;
6. set $R(d) = R(c)$ and $Pr(d) = c$;
7. remove temporarily d from Q;
8. push d to Q with the current values of μ and R;
9. *endif*;
10. *endfor*;
11. *endwhile*;
12. return $\mu(\cdot, W) = \mu(\cdot)$, $\mathbb{F}$ indicated by Pr, $P_{\min} = P(S, \mathbb{F})$;

end

Properties of GC$_{\max}$; correctness

line 2: Each $c \in C$ is removed precisely once from Q

- with $\mu(c) = \mu(c, W)$
- with $\prec$-maximal value of $\langle \mu(c), R(c) \rangle$

Proof: If the above fails for a $c \in C$ and c comes from the fist execution of line 2 when this happens, then, in earlier execution of lines 4-9, the value $\langle \mu(c), R(c) \rangle$ would have been increased.

So, indeed $\mathbb{F}$ is OPF and
$P_{\min} = \mathcal{P}(S, \mathbb{F})$ is the $\subset$-smallest element of $\mathcal{P}_{OPF}(S, T)$.

Next we show that $P_{\min} = P(S, \hat{\mathbb{F}})$ for some MSF $\hat{\mathbb{F}}$

Kruskal's algorithm KA

Kruskal's algorithm creates MSF $\hat{\mathbb{F}}$ for $G = \langle C, E, w \rangle$ as follows:

- it lists all edges of the graph in a queue Q such that their weights form a non-increasing sequence;
- it removes consecutively the edges from Q, adding to $\hat{\mathbb{F}}$ those, which addition creates, in the expanded $\hat{\mathbb{F}}$, neither a cycle nor a path between different vertices from W; other edges are discarded.

This schema has a leeway in choosing the order of edges in Q: those that have the same weight can be ordered arbitrarily.

This leeway will be exploited in the next proof.

Construction of MSF $\hat{\mathbb{F}}$ with $P_{\min} = P(S, \hat{\mathbb{F}})$

Put $B = \mathrm{bd}(P(S, \mathbb{F}))$.

Insert every $e \in E$ into queue Q such that:

- the weights of $e \in Q$ are in a non-increasing order;
- among the edges with the same weight,
  all those from $E \setminus B$ precede all those from B.

Apply Kruskal's algorithm to this Q to get MSF $\hat{\mathbb{F}}$.

$\hat{\mathbb{F}}$ is an MSF by the power of Kruskal's algorithm.

To prove that $P(S, \hat{\mathbb{F}}) = P(S, \mathbb{F})$
it is enough to show that $\hat{\mathbb{F}} \cap B = \emptyset$.

$\hat{\mathbb{F}}$ is disjoint with $B = \mathrm{bd}(P(S, \mathbb{F}))$

Let $e = \{c, d\} \in B = \mathrm{bd}(P(S, \mathbb{F}))$, $c \in P(T, \mathbb{F})$. We show that:

In KA, adding e to $\hat{\mathbb{F}}$ would create a path from S to T.

Let p_c and p_d be the paths in $\mathbb{F}$ from W to c and d. Then

$$\mu(p_c) \geq w_e \text{ and } \mu(p_d) \geq w_e. \tag{1}$$

Proof: If $\mu(p_c) > \mu(p_d)$, then $w_e \leq \mu(p_d)$, since otherwise $\mu(p_d) < \min\{\mu(p_c), w_e\} \leq \mu(d, W)$, contradicting optimality of p_d.

Similarly, $\mu(p_c) < \mu(p_d)$ implies $w_e \leq \mu(p_c)$.

Finally, $\mu(p_c) = \mu(p_d)$ implies $w_e < \mu(p_c) = \mu(p_d)$, since otherwise $\mathrm{GC}_{\max}$ (during the execution of lines 6-8 for c and d) would reassign d to $P(T, \mathbb{F})$, contradicting $d \in P(S, \mathbb{F})$.
So, (1) is proved.

$\hat{\mathbb{F}}$ is disjoint with $B = \mathrm{bd}(P(S, \mathbb{F}))$, continuation

For $e = \{c, d\} \in B = \mathrm{bd}(P(S, \mathbb{F}))$, $c \in V \setminus P(S, \mathbb{F})$, we show:

In KA, adding e to $\hat{\mathbb{F}}$ would create a path from S to T.

For paths p_c and p_d in $\mathbb{F}$ from W to c and d,

$$\mu(p_c) \geq w_e \text{ and } \mu(p_d) \geq w_e.$$

Let $E' = \{e' \in E \colon w_{e'} \geq w_e\} \setminus B$. Then, $\hat{\mathbb{F}} \cap E'$ is already constructed by KA. It is enough to show that

In $\hat{G} = \langle V, \hat{\mathbb{F}} \cap E' \rangle$ there is path $\hat{p}_d$ from S to d and $\hat{p}_c$ from T to c.

Proof. The component $\mathbb{C}$ of d in $\hat{G}$ intersects S, as otherwise there is an $\hat{e} \in p_d \subset E'$ only one vertex of which intersects $\mathbb{C}$ and $\hat{e} \in E'$ would have been added to $\hat{\mathbb{F}}$, but was not. So, indeed, there is $\hat{p}_d$ as claimed. Similarly, for $\hat{p}_c$. QED

If $\mathbb{F}$ is an MSF, then $P(S,\mathbb{F})$ minimizes ε_∞

Let $\mathbb{F}$ be an MSF and $P = P(S,\mathbb{F})$. Note that

$$\varepsilon_{\min} \stackrel{\text{def}}{=} \{\varepsilon_\infty(P)\colon P \in \mathcal{P}(S,T)\} = \max\{\mu(p)\colon p \text{ is from } S \text{ to } T\}$$

We need to show that $\varepsilon_\infty(P) \le \varepsilon_{\min}$. Assume it is not.

Then, there is an $e = \{c,d\} \in E$ with $c \in P = P(S,\mathbb{F}) \cap \text{bd}(P)$ for which $w_e > \varepsilon_{\min}$. Let p_c and p_d be the paths in $\mathbb{F}$ from W to c and d. Then either $\mu(p_c) < w_e$ or $\mu(p_d) < w_e$; otherwise there is path p from S to T with $\mu(p) = w_e > \varepsilon_{\min}$, a contradiction.

Assume that $\mu(p_c) < w_e$. Then $p_c = \langle c_1, \ldots, c_k\rangle$ with $k > 1$ and $e' = \{c_{k-1}, c_k\}$ has weight $\le \mu(p_c) < w_e$. But then $\mathbb{F}' = \mathbb{F} \cup \{e\} \setminus \{e'\}$ is a spanning forest w.r.t. W with $w(\mathbb{F}') = w(\mathbb{F}) + w_e - w_{e'} > w(\mathbb{F})$, contradicting that $\mathbb{F}$ is MSF. QED

Summary

We proved that GC_{max} algorithm returns OPF $\mathbb{F}$ for which $P(S,\mathbb{F})$ minimizes $\varepsilon_\infty(P) \stackrel{\text{def}}{=} \max_{e\in \mathrm{bd}(P)} w(e)$ in $\mathcal{P}(S,T)$.

Moreover,

$$P_{\min} \in \mathcal{P}_{MSF}(S,T) \subset \mathcal{P}_{OPF}(S,T) \cap \mathcal{P}_{\varepsilon_\infty}(S,T),$$

where $\mathcal{P}_{MSF}(S,T) = \{P(S,\mathbb{F})\colon \mathbb{F} \text{ is MSF}\}$, similarly for OPF, and $\mathcal{P}_{\varepsilon_\infty}(S,T)$ is the set of all ε_∞-optimizers.

None of the inclusions can be reversed.

Thank you for your attention!