Optimization of Max-Norm Objective Functions in Image Processing and Computer Vision

Krzysztof Chris Ciesielski

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

Joint work with Filip Malmberg and Robin Strand

MIPG Seminar, UPenn, March 7, 2019

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

Optimization in image processing

- Many fundamental problems in image processing and computer vision, such as image filtering, segmentation, registration, and stereo vision, can naturally be formulated as optimization problems.
- Often, these optimization problems can be described as *labeling* problems, in which we wish to assign to each image element (pixel) an element from some finite set of labels.
- We identify each image with a vertex weighted graph $\mathcal{G} = (V, \mathcal{E}, f)$, with vertices V being image voxels, edges $\mathcal{E}$ being pairs $\{s, t\}$ of adjacent voxels, and $f(s)$ image intensity at s. Its labeling is a map $\ell \colon V \to \{0, \ldots, n-1\}$, with $n \geq 2$.

Optimization in image processing

- Many fundamental problems in image processing and computer vision, such as image filtering, segmentation, registration, and stereo vision, can naturally be formulated as optimization problems.
- Often, these optimization problems can be described as *labeling* problems, in which we wish to assign to each image element (pixel) an element from some finite set of labels.
- We identify each image with a vertex weighted graph $\mathcal{G} = (V, \mathcal{E}, f)$, with vertices V being image voxels, edges $\mathcal{E}$ being pairs $\{s, t\}$ of adjacent voxels, and $f(s)$ image intensity at s. Its labeling is a map $\ell\colon V \to \{0, \ldots, n-1\}$, with $n \geq 2$.

Optimization in image processing

- Many fundamental problems in image processing and computer vision, such as image filtering, segmentation, registration, and stereo vision, can naturally be formulated as optimization problems.
- Often, these optimization problems can be described as *labeling* problems, in which we wish to assign to each image element (pixel) an element from some finite set of labels.
- We identify each image with a vertex weighted graph $\mathcal{G} = (V, \mathcal{E}, f)$, with vertices V being image voxels, edges $\mathcal{E}$ being pairs $\{s, t\}$ of adjacent voxels, and $f(s)$ image intensity at s. Its labeling is a map $\ell \colon V \to \{0, \ldots, n-1\}$, with $n \geq 2$.

Optimization in image processing

- Many fundamental problems in image processing and computer vision, such as image filtering, segmentation, registration, and stereo vision, can naturally be formulated as optimization problems.
- Often, these optimization problems can be described as *labeling* problems, in which we wish to assign to each image element (pixel) an element from some finite set of labels.
- We identify each image with a vertex weighted graph $\mathcal{G} = (V, \mathcal{E}, f)$, with vertices V being image voxels, edges $\mathcal{E}$ being pairs $\{s, t\}$ of adjacent voxels, and $f(s)$ image intensity at s. Its labeling is a map $\ell \colon V \to \{0, \ldots, n-1\}$, with $n \geq 2$.

L_p energies: the case of L_1

With any image n-labeling ℓ we associate local cost map $\phi_\ell \colon V \cup \mathcal{E} \to [0, \infty]$ consisting of

- unary terms $\phi_\ell(s) = \phi_s(\ell(s))$, depending on $s \in V$, its label $\ell(s)$, and image intensity;
- pairwise terms $\phi_\ell(s,t) = \phi_{s,t}(\ell(s), \ell(t))$, depending on $\{s,t\} \in \mathcal{E}$ and their labeling. They reflect desirability of smoothness/regularity of labeling.

L_1 (graph cut) energy is defined as

$$E_1(\ell) := \|\phi_\ell\|_1 = \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)),$$

often represented as (with x_i denoting label of vertex i)

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{i,j \in \mathcal{E}} \phi_{ij}(x_i, x_j).$$

L_p energies: the case of L_1

With any image n-labeling ℓ we associate local cost map $\phi_\ell \colon V \cup \mathcal{E} \to [0, \infty]$ consisting of

- unary terms $\phi_\ell(s) = \phi_s(\ell(s))$, depending on $s \in V$, its label $\ell(s)$, and image intensity;
- pairwise terms $\phi_\ell(s, t) = \phi_{s,t}(\ell(s), \ell(t))$, depending on $\{s, t\} \in \mathcal{E}$ and their labeling. They reflect desirability of smoothness/regularity of labeling.

L_1 (graph cut) energy is defined as

$$E_1(\ell) := \|\phi_\ell\|_1 = \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)),$$

often represented as (with x_i denoting label of vertex i)

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{i,j \in \mathcal{E}} \phi_{ij}(x_i, x_j).$$

L_p energies: the case of L_1

With any image n-labeling ℓ we associate local cost map $\phi_\ell \colon V \cup \mathcal{E} \to [0, \infty]$ consisting of

- unary terms $\phi_\ell(s) = \phi_s(\ell(s))$, depending on $s \in V$, its label $\ell(s)$, and image intensity;
- pairwise terms $\phi_\ell(s,t) = \phi_{s,t}(\ell(s), \ell(t))$, depending on $\{s,t\} \in \mathcal{E}$ and their labeling. They reflect desirability of smoothness/regularity of labeling.

L_1 (graph cut) energy is defined as

$$E_1(\ell) := \|\phi_\ell\|_1 = \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)),$$

often represented as (with x_i denoting label of vertex i)

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{i,j \in \mathcal{E}} \phi_{ij}(x_i, x_j).$$

L_p energies: the case of L_1

With any image n-labeling ℓ we associate local cost map $\phi_\ell \colon V \cup \mathcal{E} \to [0, \infty]$ consisting of

- unary terms $\phi_\ell(s) = \phi_s(\ell(s))$, depending on $s \in V$, its label $\ell(s)$, and image intensity;
- pairwise terms $\phi_\ell(s,t) = \phi_{s,t}(\ell(s), \ell(t))$, depending on $\{s,t\} \in \mathcal{E}$ and their labeling. They reflect desirability of smoothness/regularity of labeling.

L_1 (graph cut) energy is defined as

$$E_1(\ell) := \|\phi_\ell\|_1 = \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)),$$

often represented as (with x_i denoting label of vertex i)

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{i,j \in \mathcal{E}} \phi_{ij}(x_i, x_j).$$

L_p energies: the case of L_1

With any image n-labeling ℓ we associate local cost map $\phi_\ell \colon V \cup \mathcal{E} \to [0, \infty]$ consisting of

- unary terms $\phi_\ell(s) = \phi_s(\ell(s))$, depending on $s \in V$, its label $\ell(s)$, and image intensity;
- pairwise terms $\phi_\ell(s,t) = \phi_{s,t}(\ell(s), \ell(t))$, depending on $\{s,t\} \in \mathcal{E}$ and their labeling. They reflect desirability of smoothness/regularity of labeling.

L_1 (graph cut) energy is defined as

$$E_1(\ell) := \|\phi_\ell\|_1 = \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)),$$

often represented as (with x_i denoting label of vertex i)

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{i,j \in \mathcal{E}} \phi_{ij}(x_i, x_j).$$

L_p energies: the case of L_1

With any image n-labeling ℓ we associate local cost map $\phi_\ell\colon V \cup \mathcal{E} \to [0,\infty]$ consisting of

- unary terms $\phi_\ell(s) = \phi_s(\ell(s))$, depending on $s \in V$, its label $\ell(s)$, and image intensity;
- pairwise terms $\phi_\ell(s,t) = \phi_{s,t}(\ell(s),\ell(t))$, depending on $\{s,t\} \in \mathcal{E}$ and their labeling. They reflect desirability of smoothness/regularity of labeling.

L_1 (graph cut) energy is defined as

$$E_1(\ell) := \|\phi_\ell\|_1 = \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t)),$$

often represented as (with x_i denoting label of vertex i)

$$E(\mathbf{x}) = \sum_{i\in\mathcal{V}} \phi_i(x_i) + \sum_{i,j\in\mathcal{E}} \phi_{ij}(x_i,x_j).$$

L_p energies: the cases of $p \in (1, \infty]$

For $p \in [1, \infty)$:

$$E_p(\ell) := \|\phi_\ell\|_p = \left(\sum_{s \in V} (\phi_s(\ell(s)))^p + \sum_{\{s,t\} \in \mathcal{E}} (\phi_{st}(\ell(s), \ell(t)))^p \right)^{1/p}$$

For $p = \infty$ (of main interest here)

$$E_\infty(\ell) := \|\phi_\ell\|_\infty = \max \left\{ \max_{s \in V} \phi_s(\ell(s)), \max_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)) \right\}$$

Standard analysis fact: $E_p(\ell) \nearrow_{p \to \infty} E_\infty(\ell)$.

L_p energies: the cases of $p \in (1, \infty]$

For $p \in [1, \infty)$:

$$E_p(\ell) := \|\phi_\ell\|_p = \left(\sum_{s \in V} (\phi_s(\ell(s)))^p + \sum_{\{s,t\} \in \mathcal{E}} (\phi_{st}(\ell(s), \ell(t)))^p \right)^{1/p}$$

For $p = \infty$ (of main interest here)

$$E_\infty(\ell) := \|\phi_\ell\|_\infty = \max \left\{ \max_{s \in V} \phi_s(\ell(s)), \max_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)) \right\}$$

Standard analysis fact: $E_p(\ell) \nearrow_{p \to \infty} E_\infty(\ell)$.

L_p energies: the cases of $p \in (1, \infty]$

For $p \in [1, \infty)$:

$$E_p(\ell) := \|\phi_\ell\|_p = \left(\sum_{s \in V} (\phi_s(\ell(s)))^p + \sum_{\{s,t\} \in \mathcal{E}} (\phi_{st}(\ell(s), \ell(t)))^p \right)^{1/p}$$

For $p = \infty$ (of main interest here)

$$E_\infty(\ell) := \|\phi_\ell\|_\infty = \max \left\{ \max_{s \in V} \phi_s(\ell(s)), \max_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t)) \right\}$$

Standard analysis fact: $E_p(\ell) \nearrow_{p \to \infty} E_\infty(\ell)$.

What is the effect of p?

- The value p can be seen as a parameter controlling the balance between minimizing the overall cost $E_p(\ell)$ versus minimizing the magnitude of the individual terms $\phi_s(\ell(s))$ and $\phi_{st}(\ell(s), \ell(t))$.
- For $p = 1$, the optimal labeling may contain (few) arbitrarily large individual terms as long as the sum of the terms is small.
- As p increases, a larger penalty is assigned to solutions containing large individual terms. This forces local errors to be distributed more evenly across the image domain.

What is the effect of p?

- The value p can be seen as a parameter controlling the balance between minimizing the overall cost $E_p(\ell)$ versus minimizing the magnitude of the individual terms $\phi_s(\ell(s))$ and $\phi_{st}(\ell(s), \ell(t))$.
- For $p = 1$, the optimal labeling may contain (few) arbitrarily large individual terms as long as the sum of the terms is small.
- As p increases, a larger penalty is assigned to solutions containing large individual terms. This forces local errors to be distributed more evenly across the image domain.

What is the effect of p?

- The value p can be seen as a parameter controlling the balance between minimizing the overall cost $E_p(\ell)$ versus minimizing the magnitude of the individual terms $\phi_s(\ell(s))$ and $\phi_{st}(\ell(s), \ell(t))$.
- For $p = 1$, the optimal labeling may contain (few) arbitrarily large individual terms as long as the sum of the terms is small.
- As p increases, a larger penalty is assigned to solutions containing large individual terms. This forces local errors to be distributed more evenly across the image domain.

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

$p = 1$: Graph Cut segmentation via min-cut/max-flow

$$E_1(\ell) := \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds);
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s), \ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Min-cut/max-flow (polynomial time) algorithm returns optimized labeling **for 2-labeling**.

Optimization is NP-hard **for** $\geq$ **3-labeling**.

$p = 1$: Graph Cut segmentation via min-cut/max-flow

$$E_1(\ell) := \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))$$

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s)=\ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s)\neq\ell(t)$.

Min-cut/max-flow (polynomial time) algorithm returns optimized labeling **for 2-labeling**.

Optimization is NP-hard **for** $\geq$ **3-labeling**.

$p = 1$: Graph Cut segmentation via min-cut/max-flow

$$E_1(\ell) := \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))$$

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Min-cut/max-flow (polynomial time) algorithm returns optimized labeling **for 2-labeling**.

Optimization is NP-hard **for** $\geq$ **3-labeling**.

$p = 1$: Graph Cut segmentation via min-cut/max-flow

$$E_1(\ell) := \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds);
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s), \ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Min-cut/max-flow (polynomial time) algorithm returns optimized labeling **for 2-labeling**.

Optimization is NP-hard **for** $\geq$ **3-labeling**.

$p = 1$: Graph Cut segmentation via min-cut/max-flow

$$E_1(\ell) := \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds);
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s), \ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Min-cut/max-flow (polynomial time) algorithm returns optimized labeling **for 2-labeling**.

Optimization is NP-hard **for** $\geq$ **3-labeling**.

$p = 1$: Graph Cut segmentation via min-cut/max-flow

$$E_1(\ell) := \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds);
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s), \ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Min-cut/max-flow (polynomial time) algorithm returns optimized labeling **for 2-labeling**.

Optimization is NP-hard **for** $\geq$ **3-labeling**.

2-labeling for general $E_1(\ell)$-optimization

$$E_1(\ell) := \sum_{s \in V} \phi_s(\ell(s)) + \sum_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

E_1 (for 2-labeling) is submodular provided, for every $\{s, t\} \in \mathcal{E}$,

$$\phi_{st}(0,0) + \phi_{st}(1,1) \le \phi_{st}(0,1) + \phi_{st}(1,0).$$

Theorem (Kolmogorov & Zabih 2004)

- *If E_1 is submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_1 is **NOT** submodular, then minimizing E_1 is NP-hard.*

2-labeling for general $E_1(\ell)$-optimization

$$E_1(\ell) := \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

E_1 (for 2-labeling) is submodular provided, for every $\{s,t\} \in \mathcal{E}$,

$$\phi_{st}(0,0) + \phi_{st}(1,1) \le \phi_{st}(0,1) + \phi_{st}(1,0).$$

Theorem (Kolmogorov & Zabih 2004)

- *If E_1 is submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_1 is **NOT** submodular, then minimizing E_1 is NP-hard.*

2-labeling for general $E_1(\ell)$-optimization

$$E_1(\ell) := \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s), \ell(t))$$

E_1 (for 2-labeling) is submodular provided, for every $\{s,t\} \in \mathcal{E}$,

$$\phi_{st}(0,0) + \phi_{st}(1,1) \leq \phi_{st}(0,1) + \phi_{st}(1,0).$$

Theorem (Kolmogorov & Zabih 2004)

- *If E_1 is submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_1 is **NOT** submodular, then minimizing E_1 is NP-hard.*

2-labeling for general $E_1(\ell)$-optimization

$$E_1(\ell) := \sum_{s\in V} \phi_s(\ell(s)) + \sum_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))$$

E_1 (for 2-labeling) is submodular provided, for every $\{s,t\}\in\mathcal{E}$,

$$\phi_{st}(0,0) + \phi_{st}(1,1) \le \phi_{st}(0,1) + \phi_{st}(1,0).$$

Theorem (Kolmogorov & Zabih 2004)

- *If E_1 is submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_1 is* **NOT** *submodular, then minimizing E_1 is NP-hard.*

$E_p(\ell)$ with $1 \le p < \infty$ is as $E_1(\ell)$

$$(E_p(\ell))^p := \sum_{s\in V} (\phi_s(\ell(s)))^p + \sum_{\{s,t\}\in\mathcal{E}} (\phi_{st}(\ell(s),\ell(t)))^p$$

E_p is *p*-submodular provided, for every $\{s,t\} \in \mathcal{E}$,

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \le \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

Corollary (Obvious, Malmberg & Strand, IWCIA 2018)

- *If E_p is p-submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_p is* **NOT** *p-submodular, then minimizing E_p is NP-hard.*

$E_p(\ell)$ with $1 \leq p < \infty$ is as $E_1(\ell)$

$$(E_p(\ell))^p := \sum_{s\in V}(\phi_s(\ell(s)))^p + \sum_{\{s,t\}\in\mathcal{E}}(\phi_{st}(\ell(s),\ell(t)))^p$$

E_p is *p*-submodular provided, for every $\{s,t\} \in \mathcal{E}$,

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \leq \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

Corollary (Obvious, Malmberg & Strand, IWCIA 2018)

- *If E_p is p-submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_p is **NOT** p-submodular, then minimizing E_p is NP-hard.*

$E_p(\ell)$ with $1 \le p < \infty$ is as $E_1(\ell)$

$$(E_p(\ell))^p := \sum_{s\in V} (\phi_s(\ell(s)))^p + \sum_{\{s,t\}\in\mathcal{E}} (\phi_{st}(\ell(s),\ell(t)))^p$$

E_p is *p*-submodular provided, for every $\{s,t\} \in \mathcal{E}$,

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \le \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

Corollary (Obvious, Malmberg & Strand, IWCIA 2018)

- *If E_p is p-submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_p is* **NOT** *p-submodular, then minimizing E_p is NP-hard.*

$E_p(\ell)$ with $1 \le p < \infty$ is as $E_1(\ell)$

$$\left(E_p(\ell)\right)^p := \sum_{s\in V}\left(\phi_s(\ell(s))\right)^p + \sum_{\{s,t\}\in\mathcal{E}}\left(\phi_{st}(\ell(s),\ell(t))\right)^p$$

E_p is p-submodular provided, for every $\{s,t\} \in \mathcal{E}$,

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \le \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

Corollary (Obvious, Malmberg & Strand, IWCIA 2018)

- *If E_p is p-submodular, then min-cut/max-flow algorithm returns optimized labeling.*
- *If E_p is* **NOT** *p-submodular, then minimizing E_p is NP-hard.*

$E_p(\ell)$ with $1 \le p < \infty$ vs $E_\infty(\ell)$

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \le \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

p-submodular for every $p < \infty$ implies ∞-submodularity:

$$\max\{\phi_{st}(0,0), \phi_{st}(1,1)\} \le \max\{\phi_{st}(1,0), \phi_{st}(0,1)\}.$$

Theorem (Malmberg & Strand, IWCIA 2018)

1- and ∞-submodularity imply p-submodularity for all p. In such case min-cut/max-flow algorithm optimizes E_p for every $p < \infty$.

$E_p(\ell)$ with $1 \leq p < \infty$ vs $E_\infty(\ell)$

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \leq \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

p-submodular for every $p < \infty$ implies ∞-submodularity:

$$\max\{\phi_{st}(0,0), \phi_{st}(1,1)\} \leq \max\{\phi_{st}(1,0), \phi_{st}(0,1)\}.$$

Theorem (Malmberg & Strand, IWCIA 2018)

1- and ∞-submodularity imply p-submodularity for all p. In such case min-cut/max-flow algorithm optimizes E_p for every $p < \infty$.

$E_p(\ell)$ with $1 \le p < \infty$ vs $E_\infty(\ell)$

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \le \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

p-submodular for every $p < \infty$ implies ∞-submodularity:

$$\max\{\phi_{st}(0,0), \phi_{st}(1,1)\} \le \max\{\phi_{st}(1,0), \phi_{st}(0,1)\}.$$

Theorem (Malmberg & Strand, IWCIA 2018)

1- and ∞-submodularity imply p-submodularity for all p. In such case min-cut/max-flow algorithm optimizes E_p for every $p < \infty$.

$E_p(\ell)$ with $1 \leq p < \infty$ vs $E_\infty(\ell)$

$$\phi_{st}(0,0)^p + \phi_{st}(1,1)^p \leq \phi_{st}(0,1)^p + \phi_{st}(1,0)^p.$$

p-submodular for every $p < \infty$ implies ∞-submodularity:

$$\max\{\phi_{st}(0,0), \phi_{st}(1,1)\} \leq \max\{\phi_{st}(1,0), \phi_{st}(0,1)\}.$$

Theorem (Malmberg & Strand, IWCIA 2018)

1- and ∞-submodularity imply p-submodularity for all p. In such case min-cut/max-flow algorithm optimizes E_p for every $p < \infty$.

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $= \infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**
Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $=\infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s)=\ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**
Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $= \infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**
Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $= \infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**
Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max\left\{\max_{s\in V} \phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))\right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $= \infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**

Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $= \infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**
Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

FC segmentations are E_∞ optimized segmentations

$$E_\infty(\ell) := \max \left\{ \max_{s\in V} \phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t)) \right\}$$

We get FC segmentations (as minimization of cut),

- $\phi_s(\ell(s)) = 0$ in all cases (except seeds, when $= \infty$);
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) = \ell(t)$;
- Cost of cut: $\phi_{st}(\ell(s),\ell(t)) > 0$ (depending of $f(s)$, $f(t)$) when $\ell(s) \neq \ell(t)$.

Dijkstra (quasi-linear time) algorithm returns optimized labeling for n-labeling **for arbitrary large n!**
Better than for $E_1(\ell)$ (i.e., GC) segmentations.

Q. For what other E_∞s are there efficient optimizing algorithms?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.

This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Efficient algorithm for 2-labeling ∞-submodular E_∞?

YES!

Theorem (Malmberg, Ciesielski, Strand, DGCI 2019)

There is an algorithm, quasi-linear with respect to $n = |V \cup \mathcal{E}|$, returning minimal 2-labeling for any ∞-submodular energy E_∞.

The algorithm is NOT Dijkstra-like! More on this latter.
This is all that is in the DGCI 2019 paper.

Natural questions, towards post DGCI 2019 work:

Q1: Is ∞-submodularity assumption essential in the thm?

Q2: Is there efficient algorithm for $\geq$ 3-labelings?

Optimal 2-labeling for $E_\infty(\ell)$ with no ∞-submodularity

Full answer to Q1:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

There is an algorithm,
quasi-linear with respect to $n = |V \cup \mathcal{E}|$,
returning minimal 2-labeling for any $E_\infty(\ell)$:

$$\max\left\{\max_{s\in V} \phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s), \ell(t))\right\}.$$

More about the algorithm latter.

Optimal 2-labeling for $E_\infty(\ell)$ with no ∞-submodularity

Full answer to Q1:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

There is an algorithm,
quasi-linear with respect to $n = |V \cup \mathcal{E}|$,
returning minimal 2-labeling for any $E_\infty(\ell)$:

$$\max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}.$$

More about the algorithm latter.

Optimal 2-labeling for $E_\infty(\ell)$ with no ∞-submodularity

Full answer to Q1:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

There is an algorithm,
quasi-linear with respect to $n = |V \cup \mathcal{E}|$,
returning minimal 2-labeling for any $E_\infty(\ell)$:

$$\max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}.$$

More about the algorithm latter.

Optimal 2-labeling for $E_\infty(\ell)$ with no ∞-submodularity

Full answer to Q1:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

There is an algorithm,
quasi-linear with respect to $n = |V \cup \mathcal{E}|$,
returning minimal 2-labeling for any $E_\infty(\ell)$:

$$\max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}.$$

More about the algorithm latter.

Optimal 2-labeling for $E_\infty(\ell)$ with no ∞-submodularity

Full answer to Q1:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

There is an algorithm,
quasi-linear with respect to $n = |V \cup \mathcal{E}|$,
returning minimal 2-labeling for any $E_\infty(\ell)$:

$$\max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}.$$

More about the algorithm latter.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Q2: What about optimal $\geq$ 3-labeling for $E_\infty(\ell)$?

Partial answer to Q2:

Theorem (Malmberg, Ciesielski, Strand; 2019 ???)

Optimization problem of the general form of E_∞ energy for more than 2 labels is NP-hard.

Remaining version of Q2:

Q: Under what conditions there exists an efficient (polynomial-time) algorithm for optimization of E_∞ energy for 3 or more labels?

Can be done in FC/Dijkstra setting. Not (NP-hard) in general.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s), \ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V} \phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s),\ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V} \phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s),\ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s), \ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s), \ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V} \phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}} \phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s),\ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s \in V} \phi_s(\ell(s)), \max_{\{s,t\} \in \mathcal{E}} \phi_{st}(\ell(s), \ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s), \ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s), \ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Optimal $\geq$ 3-labeling of $E_\infty(\ell)$ is NP-hard: proof

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

For a graph $\mathcal{G} = (V, \mathcal{E})$ put:

- $\phi_s(\ell(s)) = 0$ in all cases;
- $\phi_{st}(\ell(s),\ell(t)) = 1$ when $\ell(s) = \ell(t)$;
- $\phi_{st}(\ell(s),\ell(t)) = 0$ when $\ell(s) \neq \ell(t)$.

Then, the minimal $E_\infty(\ell)$ is 0 if, and only if, ℓ is a coloring of $\mathcal{G}$

(i.e., no adjacent vertices have came label).

But graph n-coloring problem for any $n \geq 3$ is NP-complete!
It is not for $n = 2$.

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{..}$ and $\phi_.$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\} : s\in V\} \cup \{\{(s,\ell(s)),(t,\ell(t))\} : \{s,t\}\in\mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v\in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j\in\{0,1\})$ for each $\{s,t\}\in\mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{\cdot\cdot}$ and $\phi_{\cdot}$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\} : s\in V\} \cup \{\{(s,\ell(s)),(t,\ell(t))\} : \{s,t\}\in\mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v\in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j\in\{0,1\})$ for each $\{s,t\}\in\mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{\cdot\cdot}$ and $\phi_\cdot$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\} : s\in V\} \cup \{\{(s,\ell(s)),(t,\ell(t))\} : \{s,t\}\in\mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v\in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j\in\{0,1\})$ for each $\{s,t\}\in\mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{..}$ and $\phi_.$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\}: s\in V\}\cup\{\{(s,\ell(s)),(t,\ell(t))\}: \{s,t\}\in\mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v\in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j\in\{0,1\})$ for each $\{s,t\}\in\mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{..}$ and $\phi_.$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\} : s \in V\} \cup \{\{(s,\ell(s)),(t,\ell(t))\} : \{s,t\} \in \mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v \in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j \in \{0,1\})$ for each $\{s,t\} \in \mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{..}$ and $\phi_.$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\} \colon s \in V\} \cup \{\{(s,\ell(s)),(t,\ell(t))\} \colon \{s,t\} \in \mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v \in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j \in \{0,1\})$ for each $\{s,t\} \in \mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{..}$ and $\phi_.$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \{\{(s,\ell(s))\} : s \in V\} \cup \{\{(s,\ell(s)),(t,\ell(t))\} : \{s,t\} \in \mathcal{E}\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v \in V$

binary: four $\{(s,i),(t,j)\}$ $(i,j \in \{0,1\})$ for each $\{s,t\} \in \mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Atoms of E_∞ and their cost

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Atoms $\mathcal{A}(\ell)$ of ℓ: input for $\phi_{..}$ and $\phi_.$ (to calculate $E_\infty(\ell)$), i.e.,

$$\mathcal{A}(\ell) := \left\{\{(s,\ell(s))\} \colon s\in V\right\} \cup \left\{\{(s,\ell(s)),(t,\ell(t))\} \colon \{s,t\}\in\mathcal{E}\right\}$$

Atoms $\mathcal{A}$ of E_∞: all such possible atoms, i.e.,

unary: two $\{(s,0)\}$ and $\{(s,1)\}$ for each $v\in V$

binary: four $\{(s,i),(t,j)\}$ ($i,j\in\{0,1\}$) for each $\{s,t\}\in\mathcal{E}$.

- Cost of a unary atom $\{(s,i)\}$: $\phi_s(i)$
- Cost of a binary atom $\{(s,i),(t,j)\}$: $\phi_{st}(i,j)$

Consistent atoms and minimization of E_∞

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Set $\mathcal{A}' \subset \mathcal{A}$ of atoms is consistent when $\mathcal{A}(\ell) \subset \mathcal{A}'$ for some ℓ

Finding $\min_\ell E_\infty(\ell)$ is equivalent to finding minimal $C \in \mathbb{R}$ with

$\mathcal{A}(C) :=$ all atoms with cost $\leq C$

being consistent.

Consistent atoms and minimization of E_∞

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Set $\mathcal{A}' \subset \mathcal{A}$ of atoms is consistent when $\mathcal{A}(\ell) \subset \mathcal{A}'$ for some ℓ

Finding $\min_\ell E_\infty(\ell)$ is equivalent to finding minimal $C \in \mathbb{R}$ with

$$\mathcal{A}(C) := \text{all atoms with cost } \leq C$$

being consistent.

Consistent atoms and minimization of E_∞

$$E_\infty(\ell) := \max\left\{\max_{s\in V}\phi_s(\ell(s)), \max_{\{s,t\}\in\mathcal{E}}\phi_{st}(\ell(s),\ell(t))\right\}$$

Set $\mathcal{A}' \subset \mathcal{A}$ of atoms is consistent when $\mathcal{A}(\ell) \subset \mathcal{A}'$ for some ℓ

Finding $\min_\ell E_\infty(\ell)$ is equivalent to finding minimal $C \in \mathbb{R}$ with

$$\mathcal{A}(C) := \text{all atoms with cost } \leq C$$

being consistent.

Towards the algorithm: Main lemma

Lemma

A set $\mathcal{A}'$ of atoms is consistent if, and only if, a naturally associated with it 2-conjunctive formula is satisfiable.

Since satisfiability of such formulas is decidable in a linear time (e.g., by Aspvall, Plass, Tarjan algorithm):

Corollary

There is an algorithm deciding consistency of a set $\mathcal{A}'$ of atoms. It has linear complexity w.r.t. $|\mathcal{A}'|$.

Towards the algorithm: Main lemma

Lemma

A set $\mathcal{A}'$ of atoms is consistent if, and only if, a naturally associated with it 2-conjunctive formula is satisfiable.

Since satisfiability of such formulas is decidable in a linear time (e.g., by Aspvall, Plass, Tarjan algorithm):

Corollary

There is an algorithm deciding consistency of a set $\mathcal{A}'$ of atoms. It has linear complexity w.r.t. $|\mathcal{A}'|$.

Towards the algorithm: Main lemma

Lemma

A set $\mathcal{A}'$ of atoms is consistent if, and only if, a naturally associated with it 2-conjunctive formula is satisfiable.

Since satisfiability of such formulas is decidable in a linear time (e.g., by Aspvall, Plass, Tarjan algorithm):

Corollary

There is an algorithm deciding consistency of a set $\mathcal{A}'$ of atoms. It has linear complexity w.r.t. $|\mathcal{A}'|$.

Towards the algorithm: Main lemma

Lemma

A set $\mathcal{A}'$ of atoms is consistent if, and only if, a naturally associated with it 2-conjunctive formula is satisfiable.

Since satisfiability of such formulas is decidable in a linear time (e.g., by Aspvall, Plass, Tarjan algorithm):

Corollary

There is an algorithm deciding consistency of a set $\mathcal{A}'$ of atoms. It has linear complexity w.r.t. $|\mathcal{A}'|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

The main algorithm

1. For all possible costs C of the atoms in $\mathcal{A}$ decide if $\mathcal{A}(C)$ is consistent.
2. The smallest C with consistent $\mathcal{A}(C)$ is our minimal energy.

Note that

- The algorithm deciding consistency of $\mathcal{A}(C)$ returns also a labeling justifying it.
- It is enough to check the consistency of $\mathcal{A}(C)$ for $\log_2 |\mathcal{A}|$-many values of C.
  So, we get complexity $O(m \ln m)$, with $m = |\mathcal{A}|$.

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

Lexicographical order $\preceq_{lex}$ among labelings

For labeling ℓ let

$\vec{A}(\ell) = \langle c_1^\ell, \ldots, c_k^\ell \rangle$: cost of all atoms $\in \mathcal{A}(\ell)$ in $\geq$-order.

$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff c_j^\ell < c_j^{\ell'}$, where $j = \min\{i \colon c_i^\ell < c_i^{\ell'}\}$

Easy fact: $E_\infty(\ell) < E_\infty(\ell') \implies \vec{A}(\ell) \prec_{lex} \vec{A}(\ell')$

So, $\prec_{lex}$ better distinguishes labelling than E_∞.

Q. Can we efficiently optimize w.r.t. $\prec_{lex}$ rather than E_∞?

Lexicographical order $\preceq_{lex}$ among labelings

For labeling ℓ let

$\vec{A}(\ell) = \langle c_1^\ell, \ldots, c_k^\ell \rangle$: cost of all atoms $\in \mathcal{A}(\ell)$ in $\geq$-order.

$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff c_j^\ell < c_j^{\ell'}$, where $j = \min\{i \colon c_i^\ell < c_i^{\ell'}\}$

Easy fact: $E_\infty(\ell) < E_\infty(\ell') \implies \vec{A}(\ell) \prec_{lex} \vec{A}(\ell')$

So, $\prec_{lex}$ better distinguishes labelling than E_∞.

Q. Can we efficiently optimize w.r.t. $\prec_{lex}$ rather than E_∞?

Lexicographical order $\preceq_{lex}$ among labelings

For labeling ℓ let

$\vec{A}(\ell) = \langle c_1^\ell, \ldots, c_k^\ell \rangle$: cost of all atoms $\in \mathcal{A}(\ell)$ in $\geq$-order.

$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff c_j^\ell < c_j^{\ell'}$, where $j = \min\{i \colon c_i^\ell < c_i^{\ell'}\}$

Easy fact: $E_\infty(\ell) < E_\infty(\ell') \implies \vec{A}(\ell) \prec_{lex} \vec{A}(\ell')$

So, $\prec_{lex}$ better distinguishes labelling than E_∞.

Q. Can we efficiently optimize w.r.t. $\prec_{lex}$ rather than E_∞?

Lexicographical order $\preceq_{lex}$ among labelings

For labeling ℓ let

$\vec{\mathcal{A}}(\ell) = \langle c_1^\ell, \ldots, c_k^\ell \rangle$: cost of all atoms $\in \mathcal{A}(\ell)$ in $\geq$-order.

$\vec{\mathcal{A}}(\ell) \prec_{lex} \vec{\mathcal{A}}(\ell') \iff c_j^\ell < c_j^{\ell'}$, where $j = \min\{i \colon c_i^\ell < c_i^{\ell'}\}$

Easy fact: $E_\infty(\ell) < E_\infty(\ell') \implies \vec{\mathcal{A}}(\ell) \prec_{lex} \vec{\mathcal{A}}(\ell')$

So, $\prec_{lex}$ better distinguishes labelling than E_∞.

Q. Can we efficiently optimize w.r.t. $\prec_{lex}$ rather than E_∞?

Lexicographical order $\preceq_{lex}$ among labelings

For labeling ℓ let

$\vec{\mathcal{A}}(\ell) = \langle c_1^\ell, \ldots, c_k^\ell \rangle$: cost of all atoms $\in \mathcal{A}(\ell)$ in $\geq$-order.

$\vec{\mathcal{A}}(\ell) \prec_{lex} \vec{\mathcal{A}}(\ell') \iff c_j^\ell < c_j^{\ell'}$, where $j = \min\{i \colon c_i^\ell < c_i^{\ell'}\}$

Easy fact: $E_\infty(\ell) < E_\infty(\ell') \implies \vec{\mathcal{A}}(\ell) \prec_{lex} \vec{\mathcal{A}}(\ell')$

So, $\prec_{lex}$ better distinguishes labelling than E_∞.

Q. Can we efficiently optimize w.r.t. $\prec_{lex}$ rather than E_∞?

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{A}(\ell) \prec_{lex} \vec{A}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{\mathcal{A}}(\ell) \prec_{lex} \vec{\mathcal{A}}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Efficient algorithm for $\preceq_{lex}$-optimization of 2-labelings?

YES when energy E is 1- and ∞-submodular.

By graph cut algorithm, since

Theorem

For any energy E, there is (easily computable) $p > 0$ s.t.

$$\vec{\mathcal{A}}(\ell) \prec_{lex} \vec{\mathcal{A}}(\ell') \iff E_p(\ell) < E_p(\ell')$$

NO when energy is not ∞-submodular.

Such problem is NP-hard: reduces to the problem of finding maximal independent set of vertices in a graph, which is known to be NP-hard.

Outline

1. Energies we will optimize
2. Algorithms for L_p, $p < \infty$; NP-completeness
3. Which max-norm energies E_∞ can be efficiently optimized?
4. Efficient algorithm optimizing E_∞ for 2-labeling
5. Lexicographical order refinement of E_∞
6. Conclusions

Conclusions

- Optimization problems, specifically pixel labeling problems, are frequently occurring in image processing applications.
- We are specifically interested in problems where the objective function is given by the max-norm of the local errors.
- For many such problems, globally optimal solutions can be found very efficiently, in quasi linear time.
- Some max-norm for $\geq$ 3-labeling are NP-hard.
- $\preceq_{lex}$-optimization equivalent to E_p-optimization for large p.

Conclusions

- Optimization problems, specifically pixel labeling problems, are frequently occurring in image processing applications.
- We are specifically interested in problems where the objective function is given by the max-norm of the local errors.
- For many such problems, globally optimal solutions can be found very efficiently, in quasi linear time.
- Some max-norm for $\geq$ 3-labeling are NP-hard.
- $\preceq_{lex}$-optimization equivalent to E_p-optimization for large p.

Conclusions

- Optimization problems, specifically pixel labeling problems, are frequently occurring in image processing applications.
- We are specifically interested in problems where the objective function is given by the max-norm of the local errors.
- For many such problems, globally optimal solutions can be found very efficiently, in quasi linear time.
- Some max-norm for $\geq$ 3-labeling are NP-hard.
- $\preceq_{lex}$-optimization equivalent to E_p-optimization for large p.

Conclusions

- Optimization problems, specifically pixel labeling problems, are frequently occurring in image processing applications.
- We are specifically interested in problems where the objective function is given by the max-norm of the local errors.
- For many such problems, globally optimal solutions can be found very efficiently, in quasi linear time.
- Some max-norm for $\geq$ 3-labeling are NP-hard.
- $\preceq_{lex}$-optimization equivalent to E_p-optimization for large p.

Conclusions

- Optimization problems, specifically pixel labeling problems, are frequently occurring in image processing applications.
- We are specifically interested in problems where the objective function is given by the max-norm of the local errors.
- For many such problems, globally optimal solutions can be found very efficiently, in quasi linear time.
- Some max-norm for $\geq$ 3-labeling are NP-hard.
- $\preceq_{lex}$-optimization equivalent to E_p-optimization for large p.

Thank you for your attention!