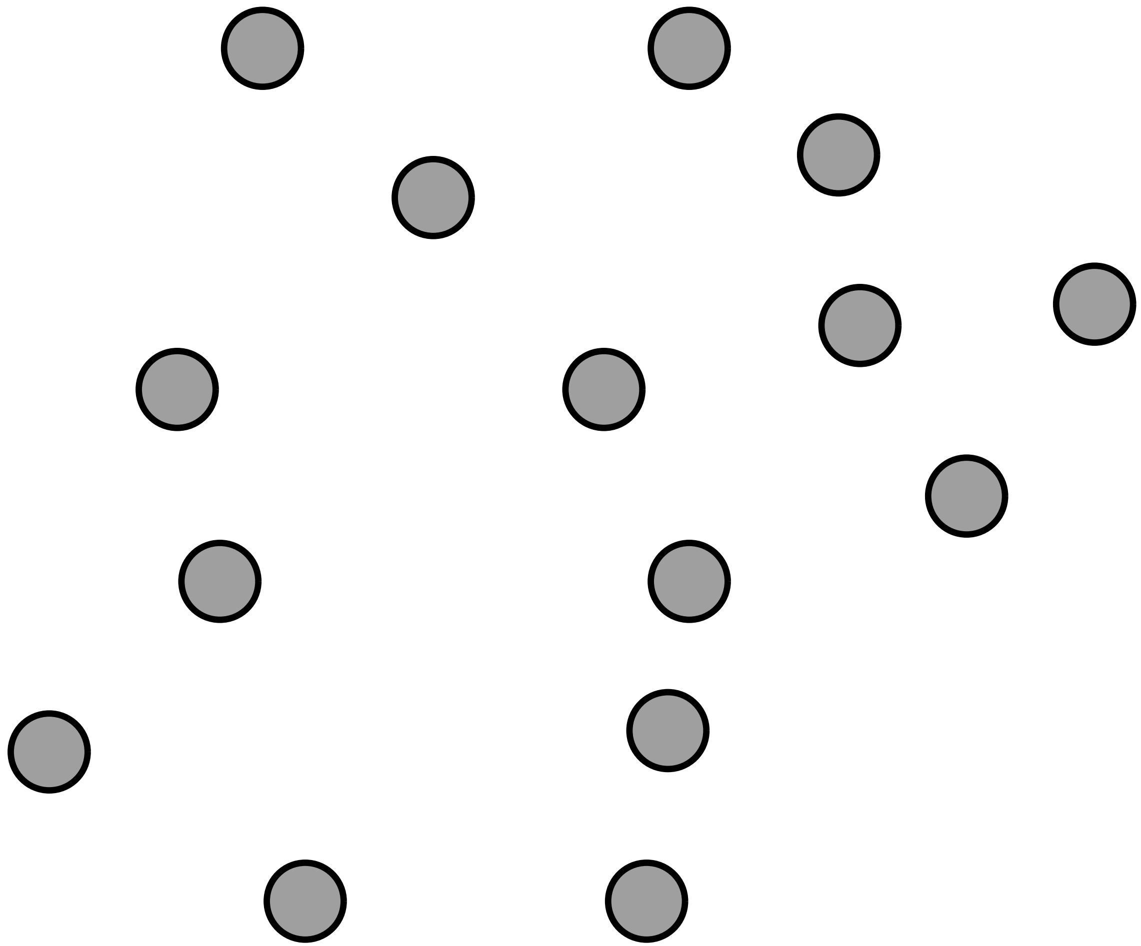
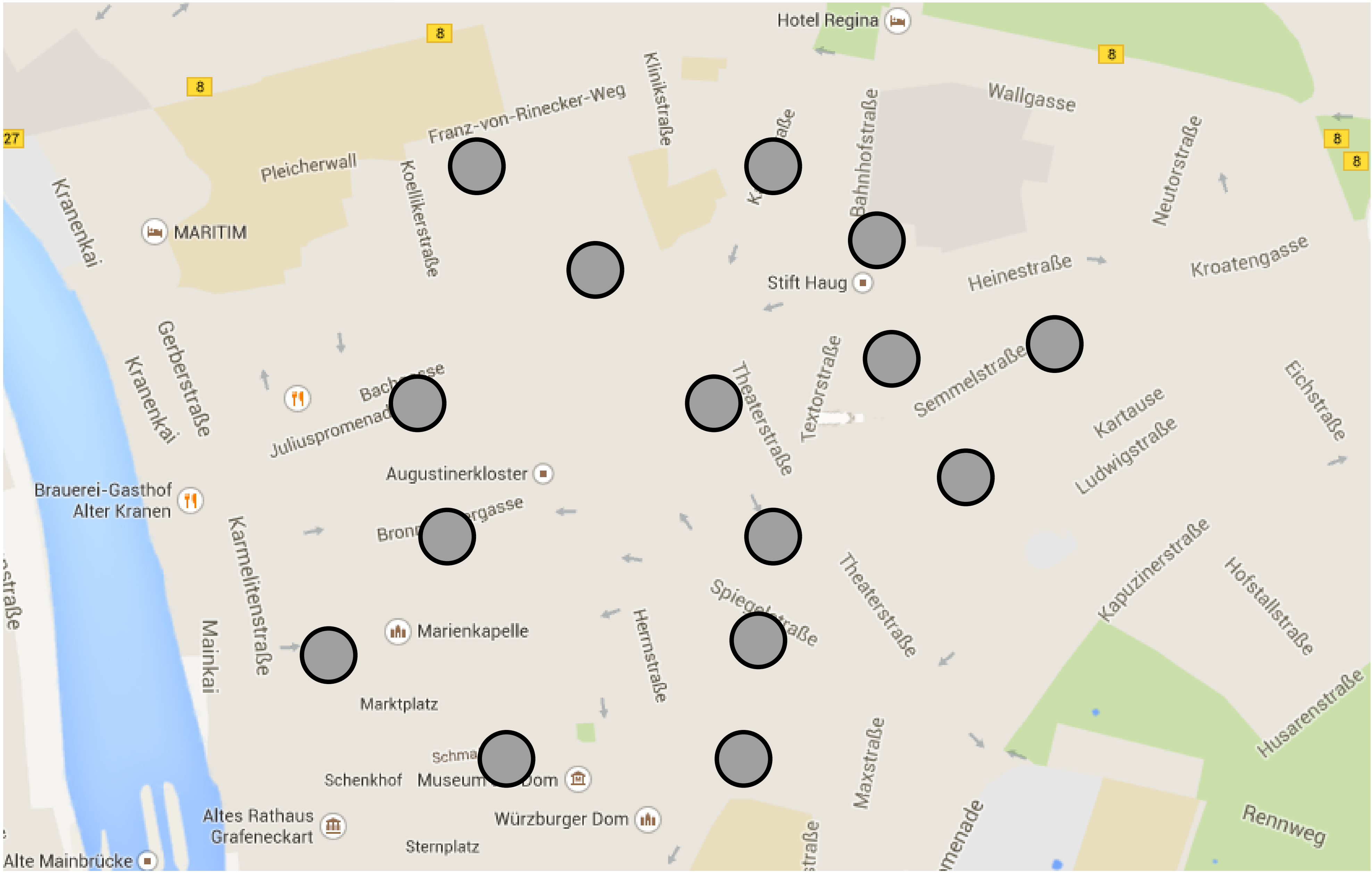
THE UNIVERSITY OF ARIZONA®

MapSets: Visualizing Embedded and Clustered Graphs

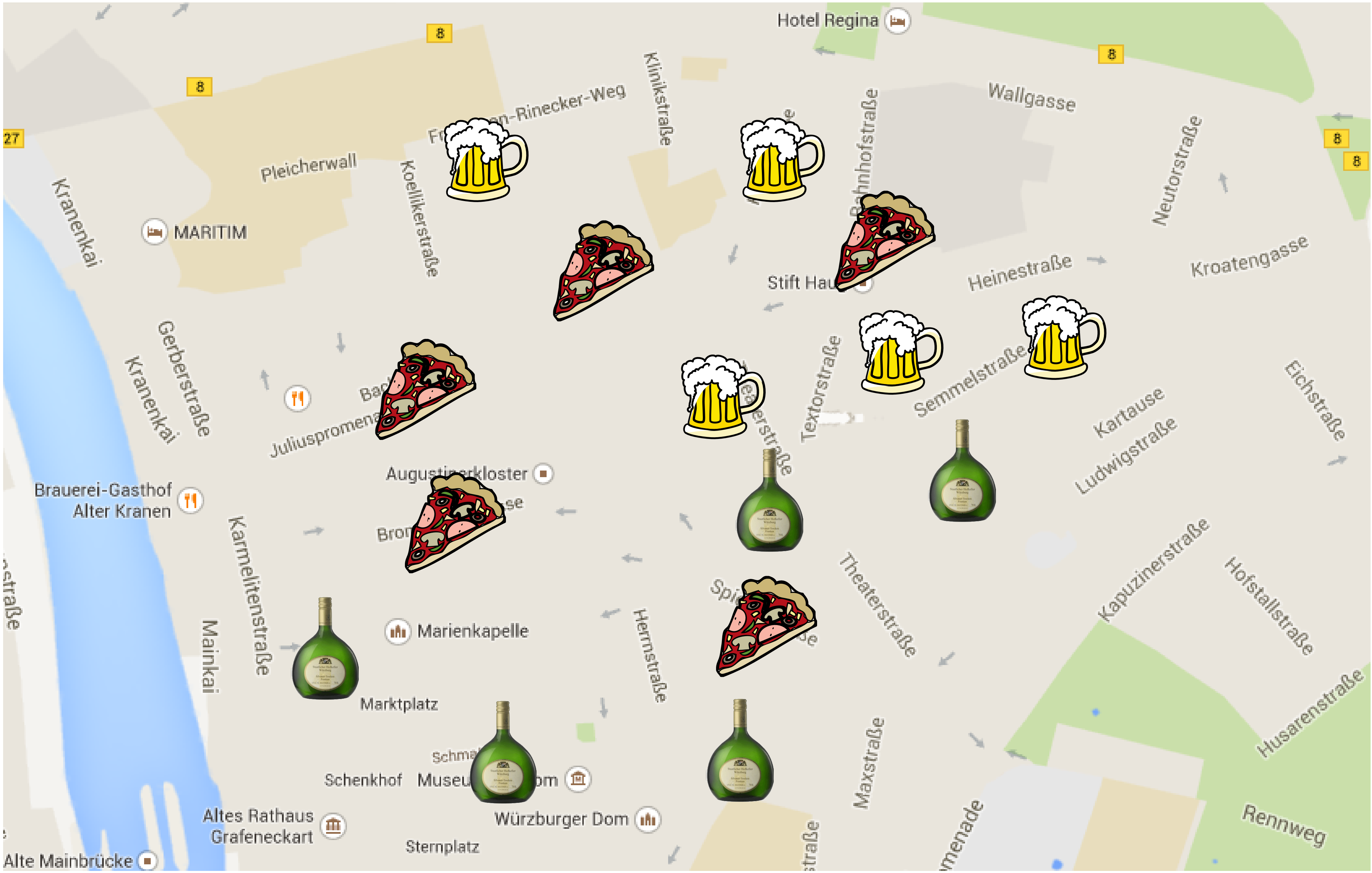
Sergey Pupyrev
University of Arizona

Joint work with Alon Efrat, Yifan Hu and Stephen Kobourov



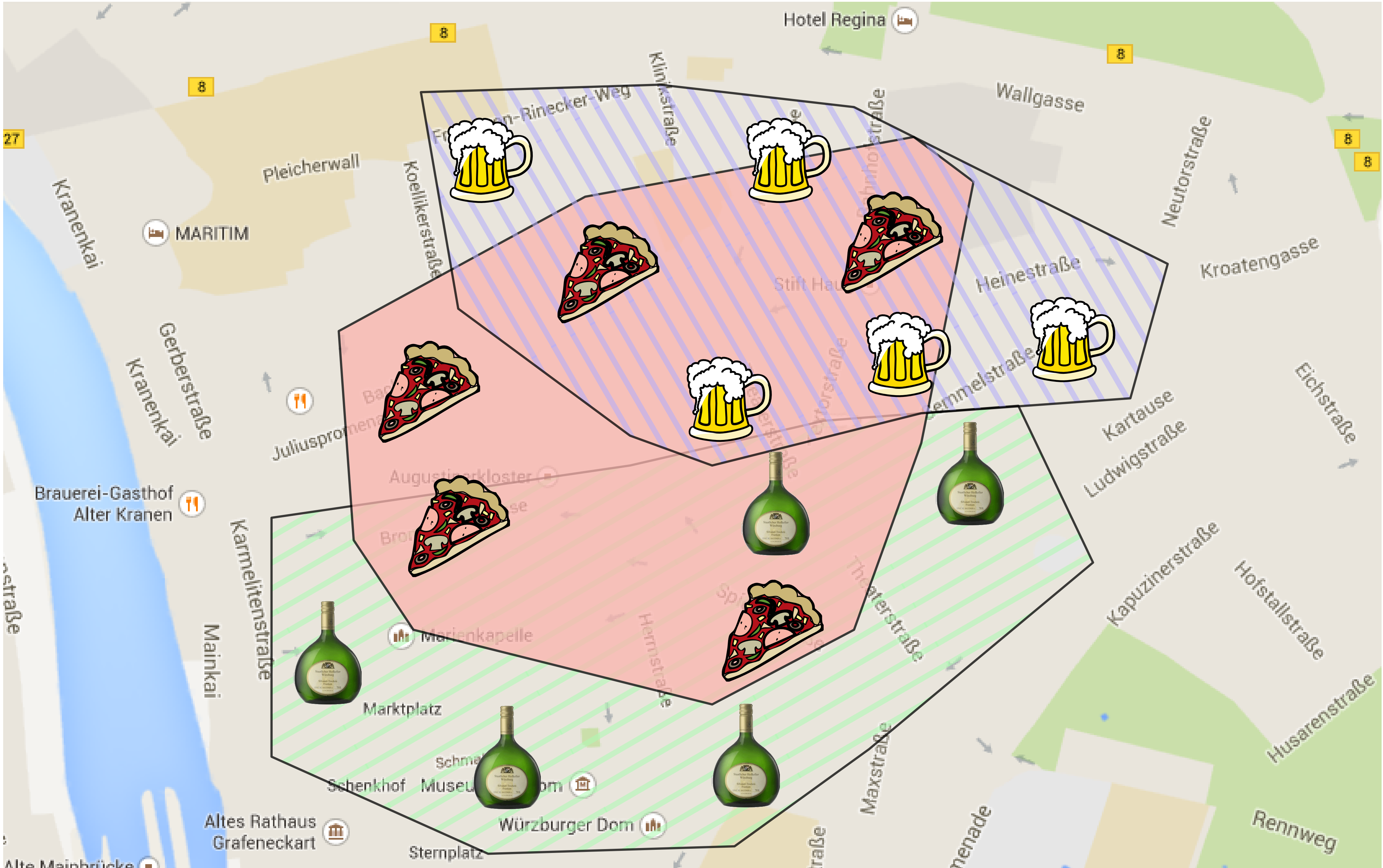


Hotel Regina
Wallgasse
Franz-von-Rinecker-Weg
Klinikstraße
Bahnhofstraße
Pleicherwall
Koellikerstraße
Kranenkai
MARITIM
Neutorstraße
Stift Haug
Heinestraße
Kroatengasse
Gerberstraße
Kranenkai
Textorstraße
Theaterstraße
Semmelstraße
Juliuspromenade
Kartause
Ludwigstraße
Eichstraße
Augustinerkloster
Brauerei-Gasthof
Alter Kranen
Karmelitenstraße
Kapuzinerstraße
Hofstallstraße
Theaterstraße
Herrnstraße
Marienkapelle
Mainkai
Marktplatz
Husarenstraße
Maxstraße
Schenkhof
Würzburger Dom
Altes Rathaus
Grafeneckart
Sternplatz
Rennweg
Alte Mainbrücke


Hotel Regina
Wallgasse
Pleicherwall
Klinikstraße
Bahnhofstraße
Neutorstraße
Kranenkai
MARITIM
Koellikerstraße
Stift Hau
Heinestraße
Kroatengasse
Gerberstraße
Kranenkai
Juliuspromena
Textorstraße
Semmelstraße
Eichstraße
Kartause
Ludwigstraße
Augustinerkloster
Brauerei-Gasthof
Alter Kranen
Karmelitenstraße
Kapuzinerstraße
Hofstallstraße
Theaterstraße
Marienkapelle
Mainkai
Herrnstraße
Husarenstraße
Marktplatz
Maxstraße
Schenkhof
Würzburger Dom
Altes Rathaus
Grafeneckart
Sternplatz
Rennweg
Alte Mainbrücke

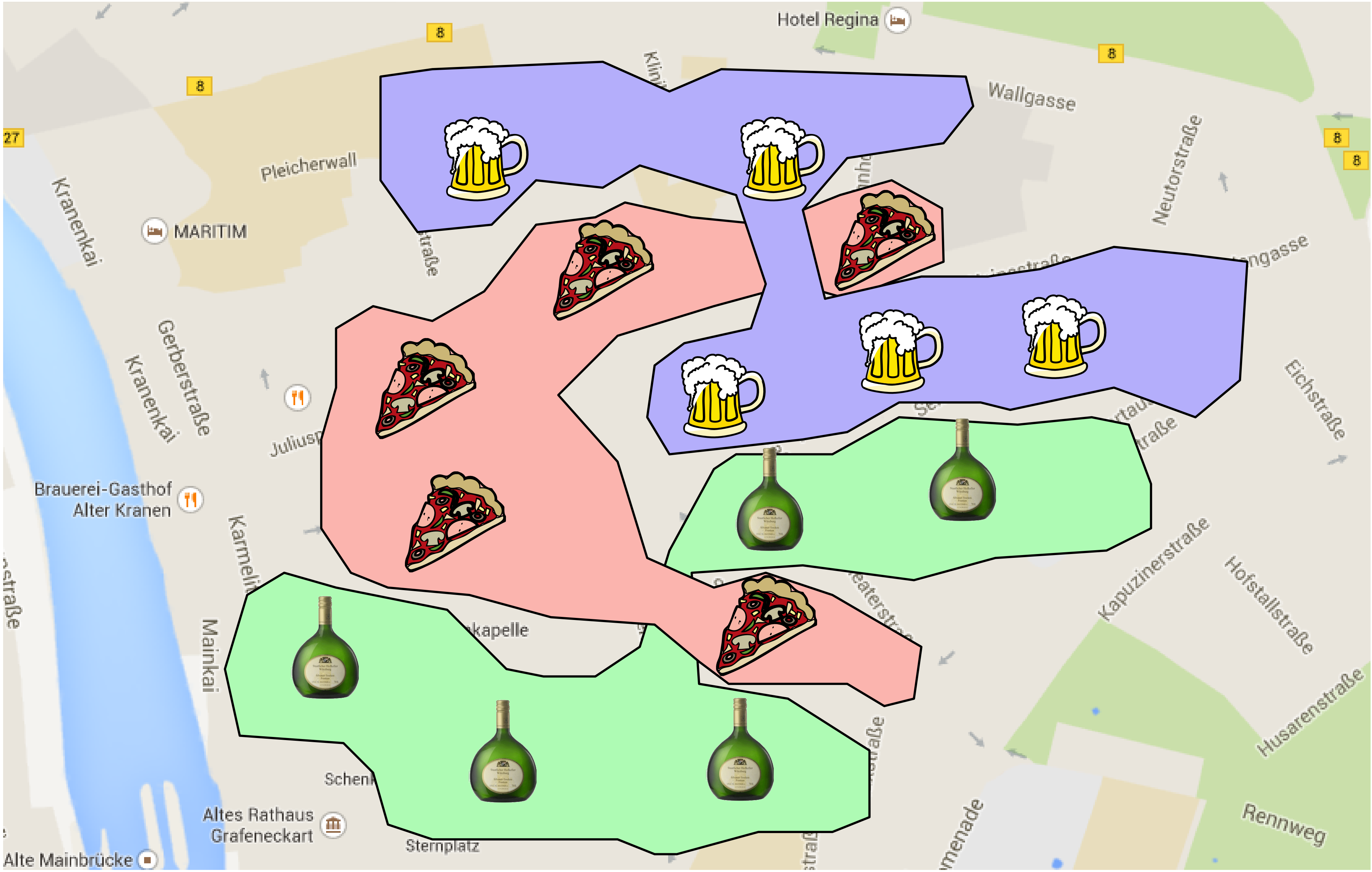
Euler diagrams

[Simonetto Auber Archambault, CGF'09]



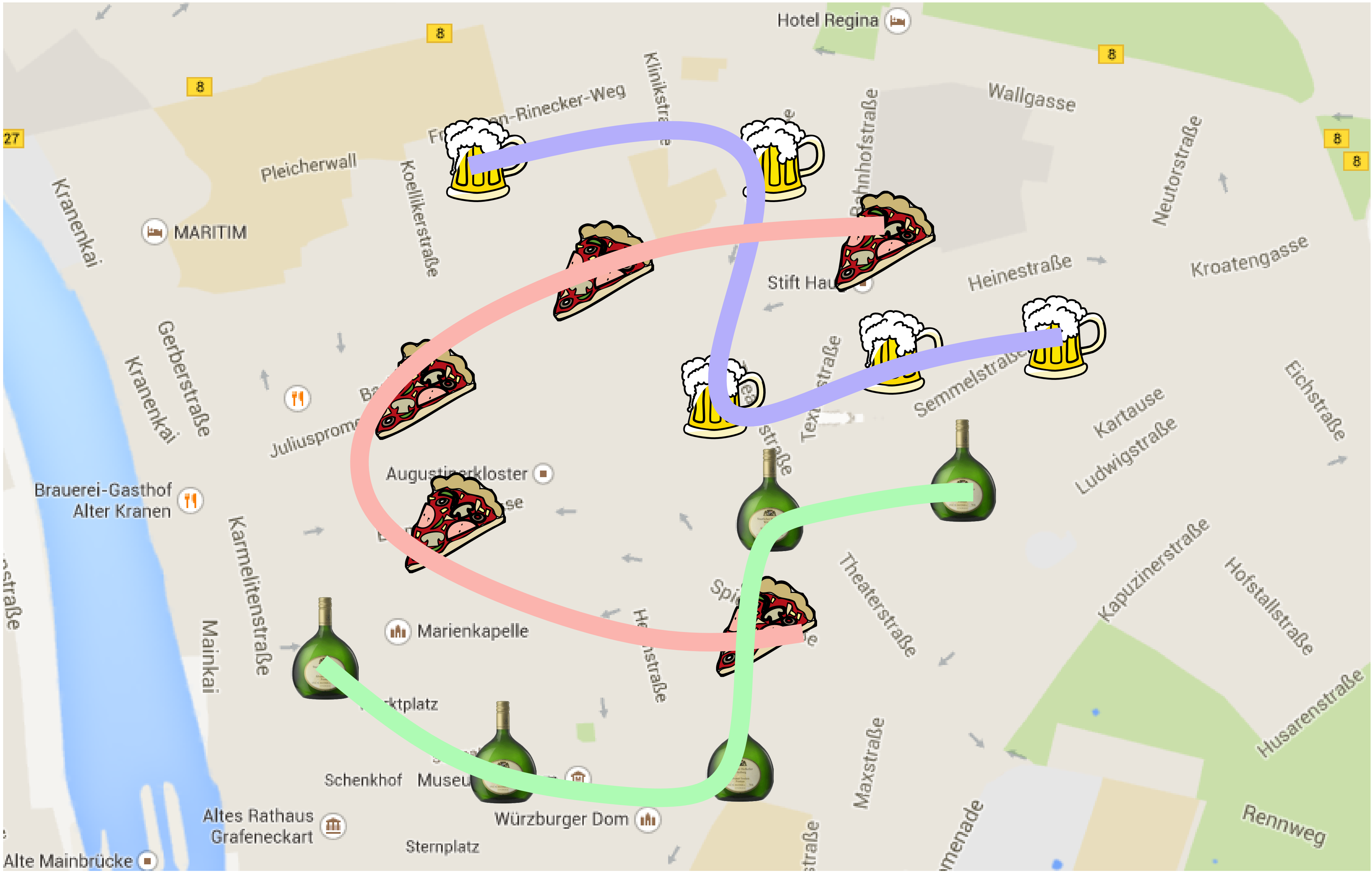
BubbleSets

[Collins Penn Carpendale, TVCG'09]



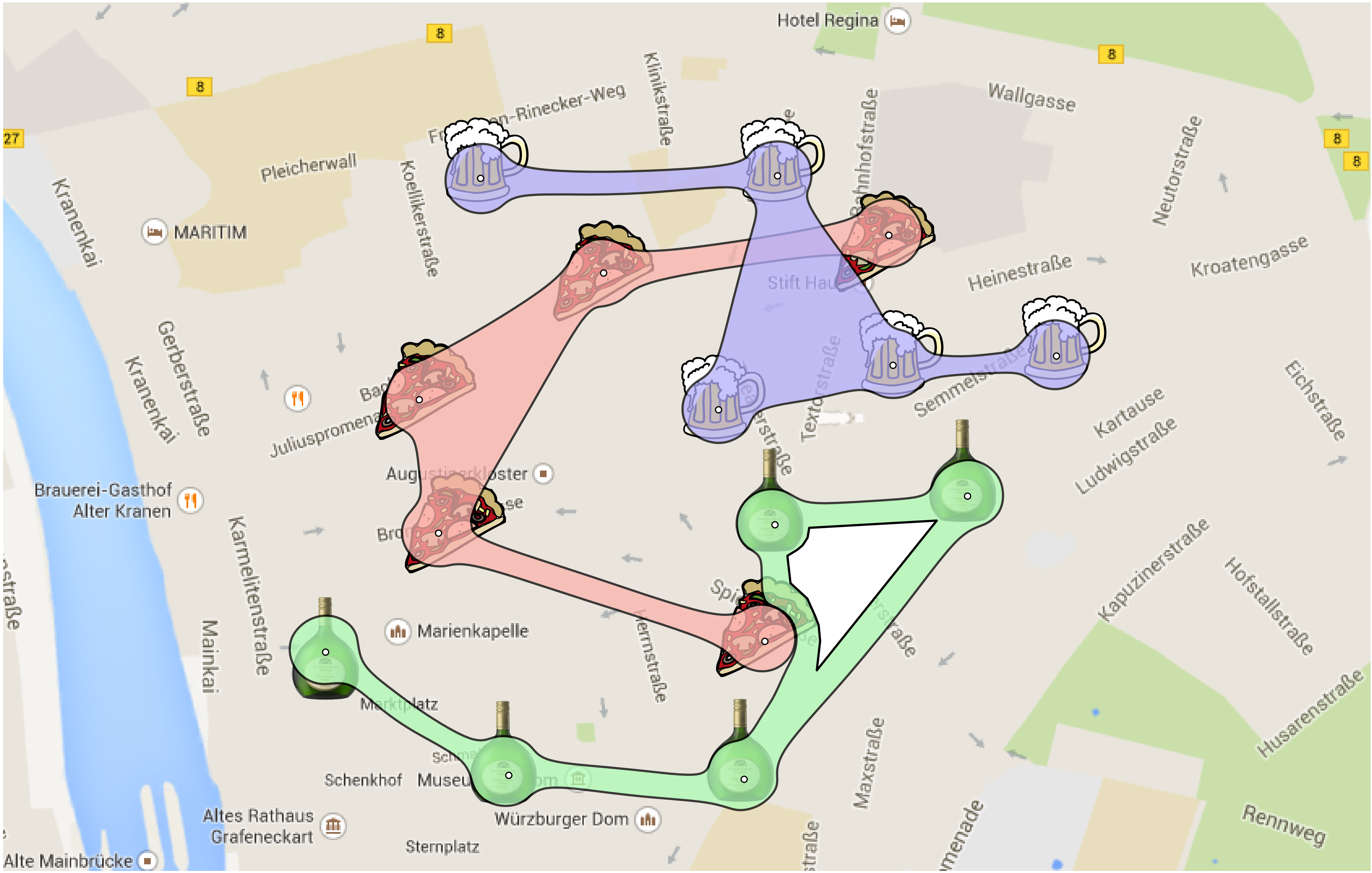
LineSets

[Alper Riche Ramos Czerwinski, TVCG'11]

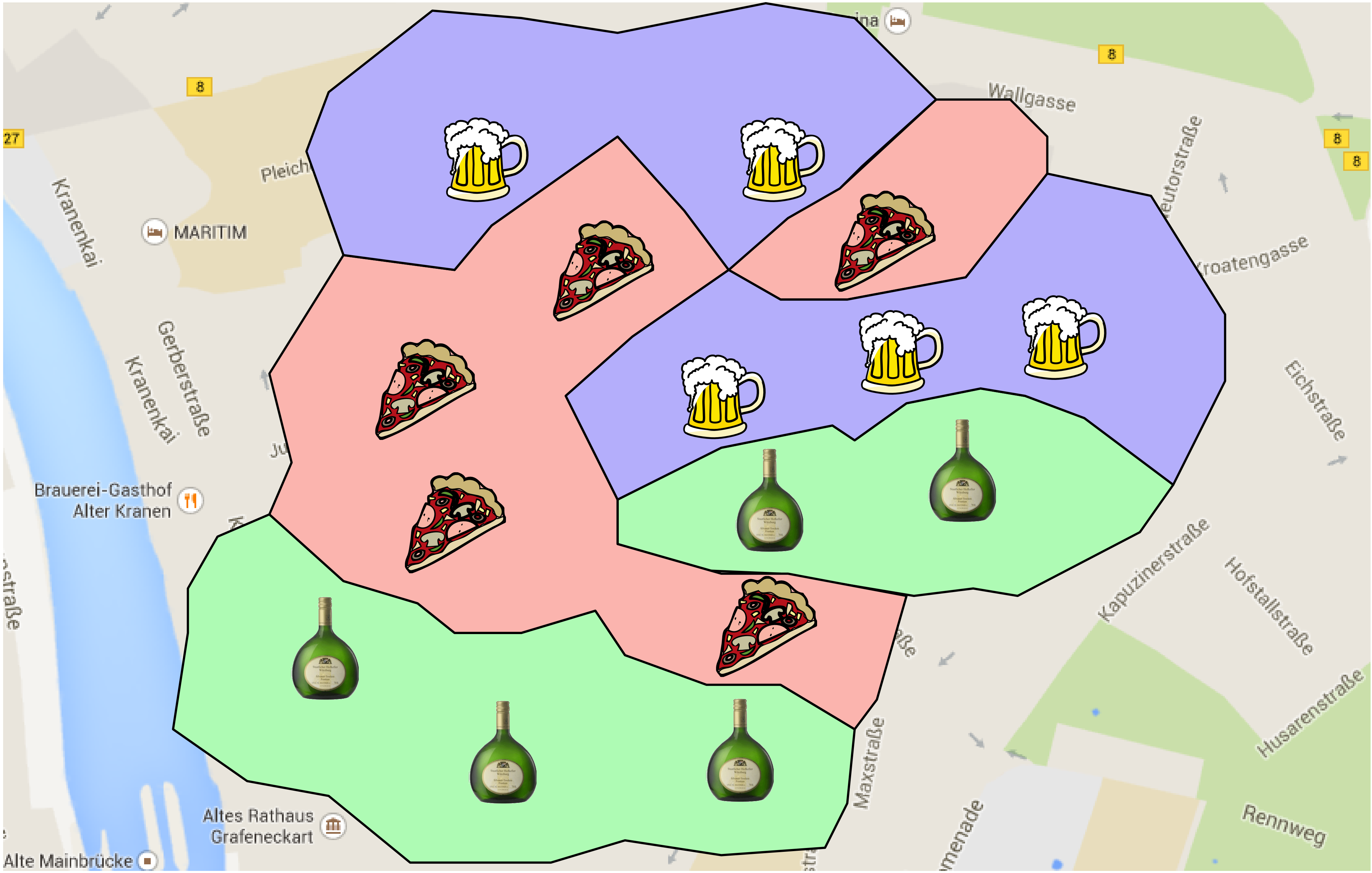


KelpFusion

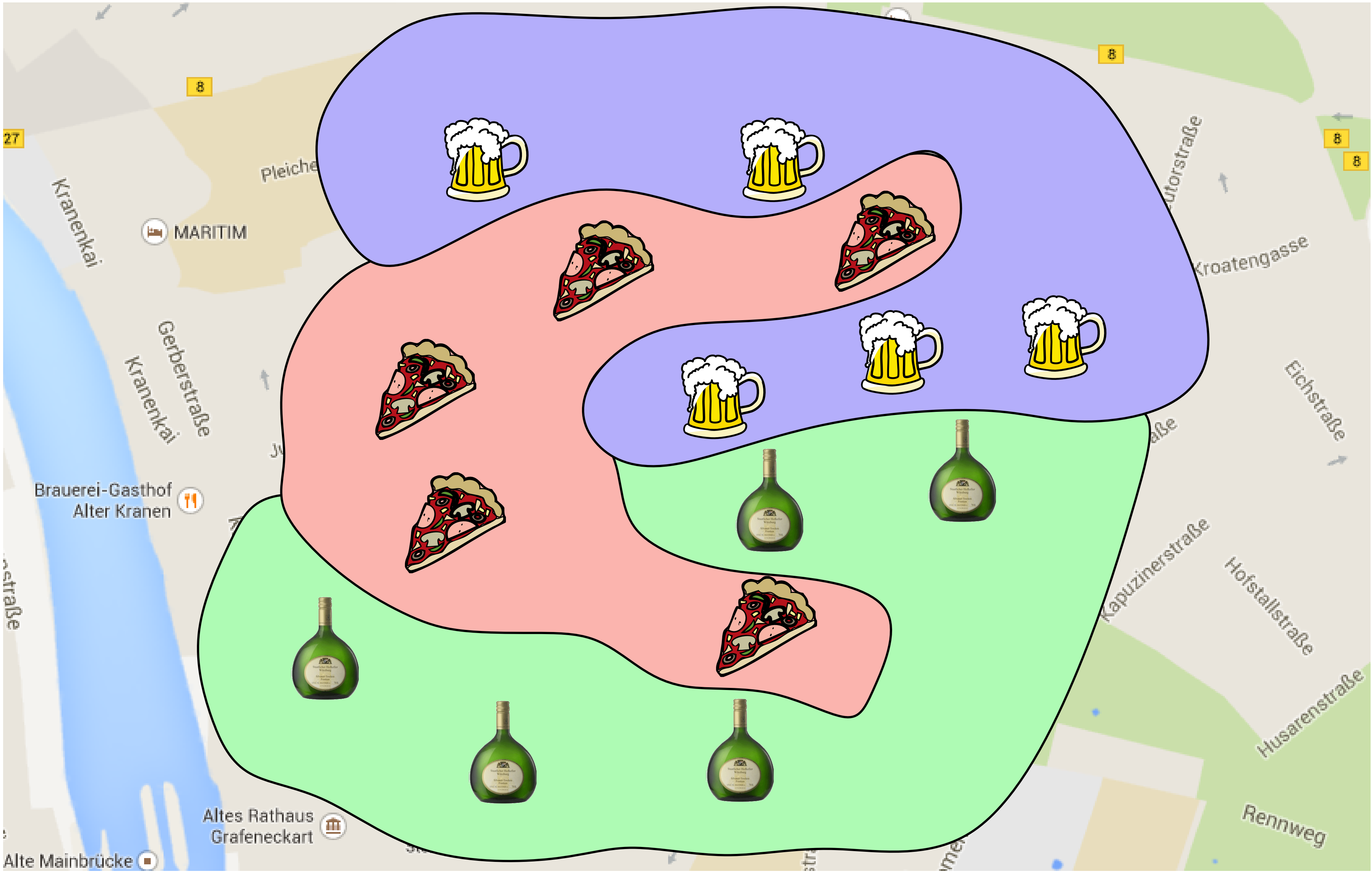
[Meulemans Riche Speckmann Alper Dwyer, TVCG'13]



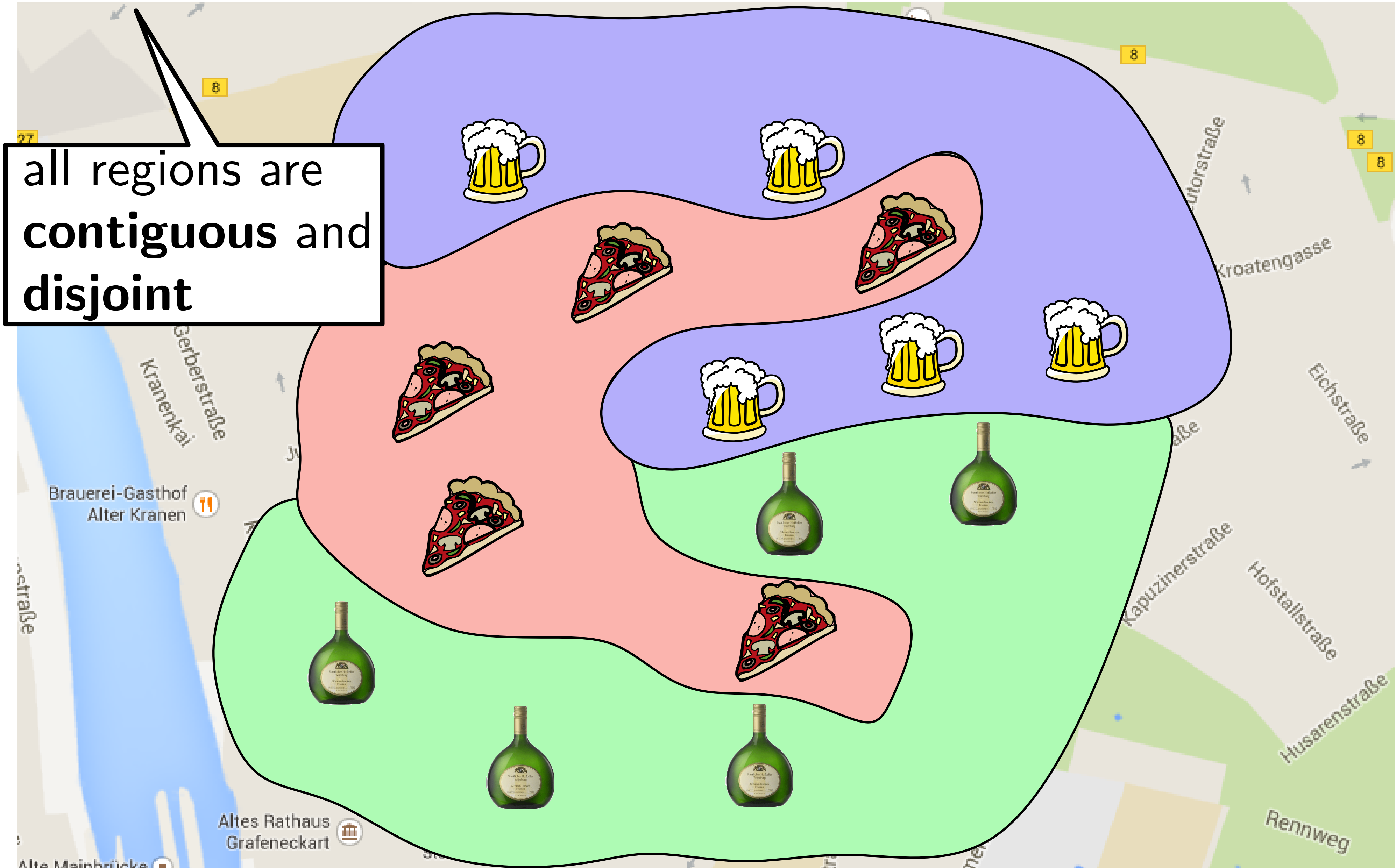
GMap (Graph-to-Map) [Hu Gansner Kobourov, CGA'10]



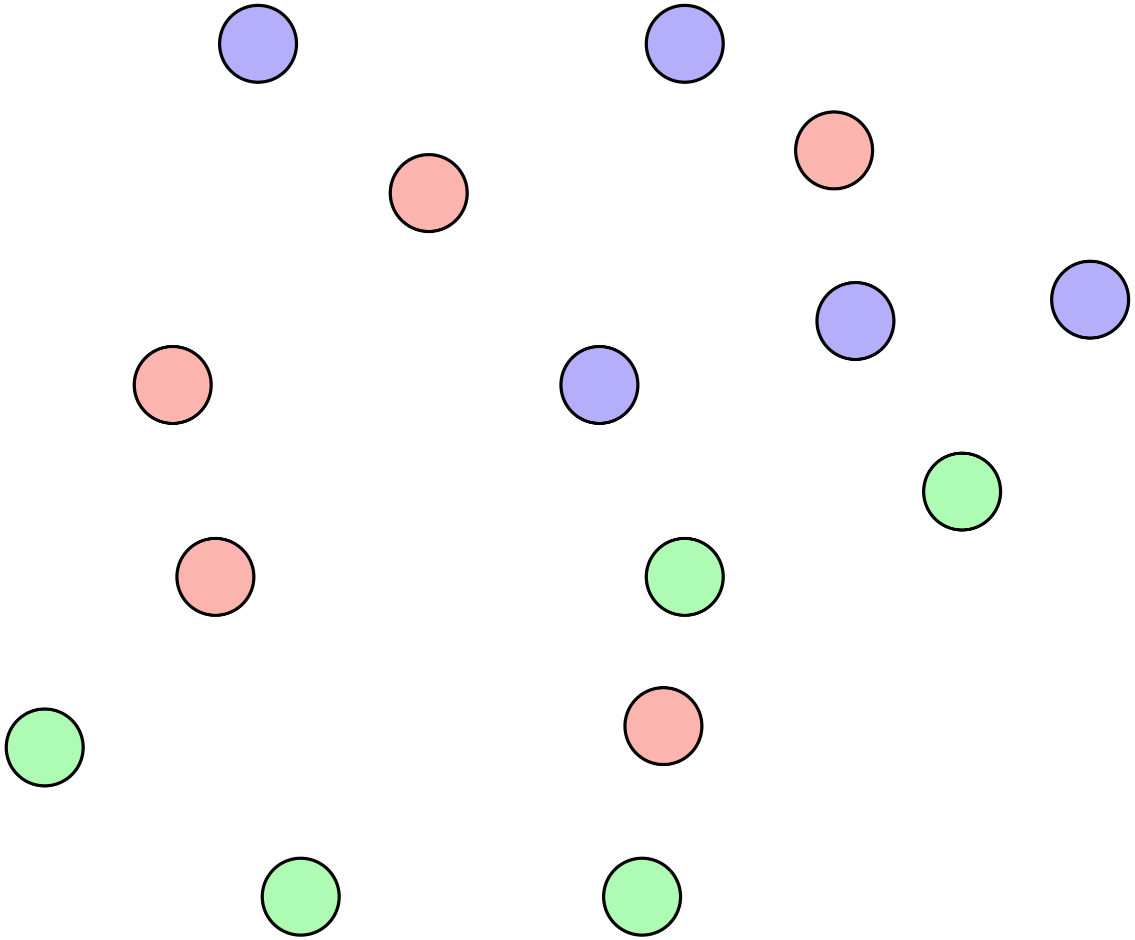
a *better* solution



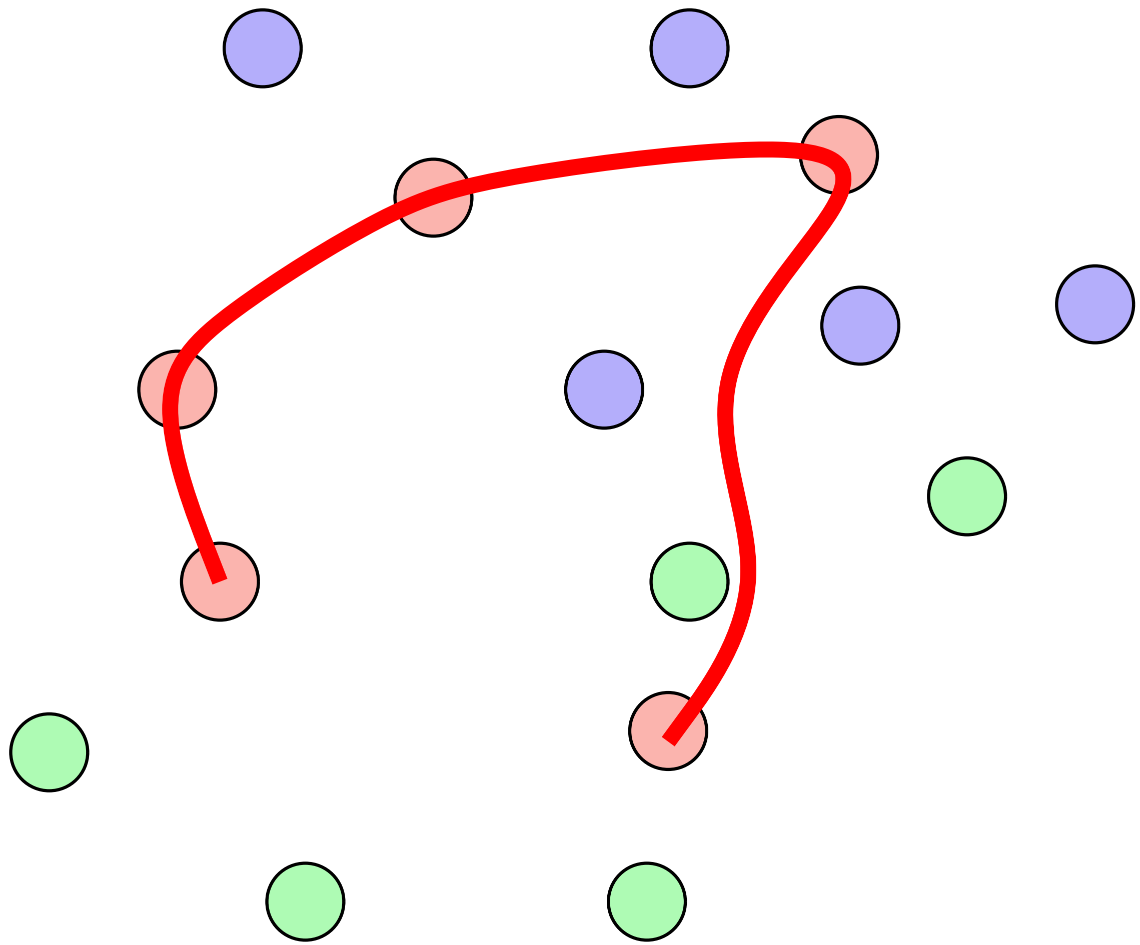
a *better* solution



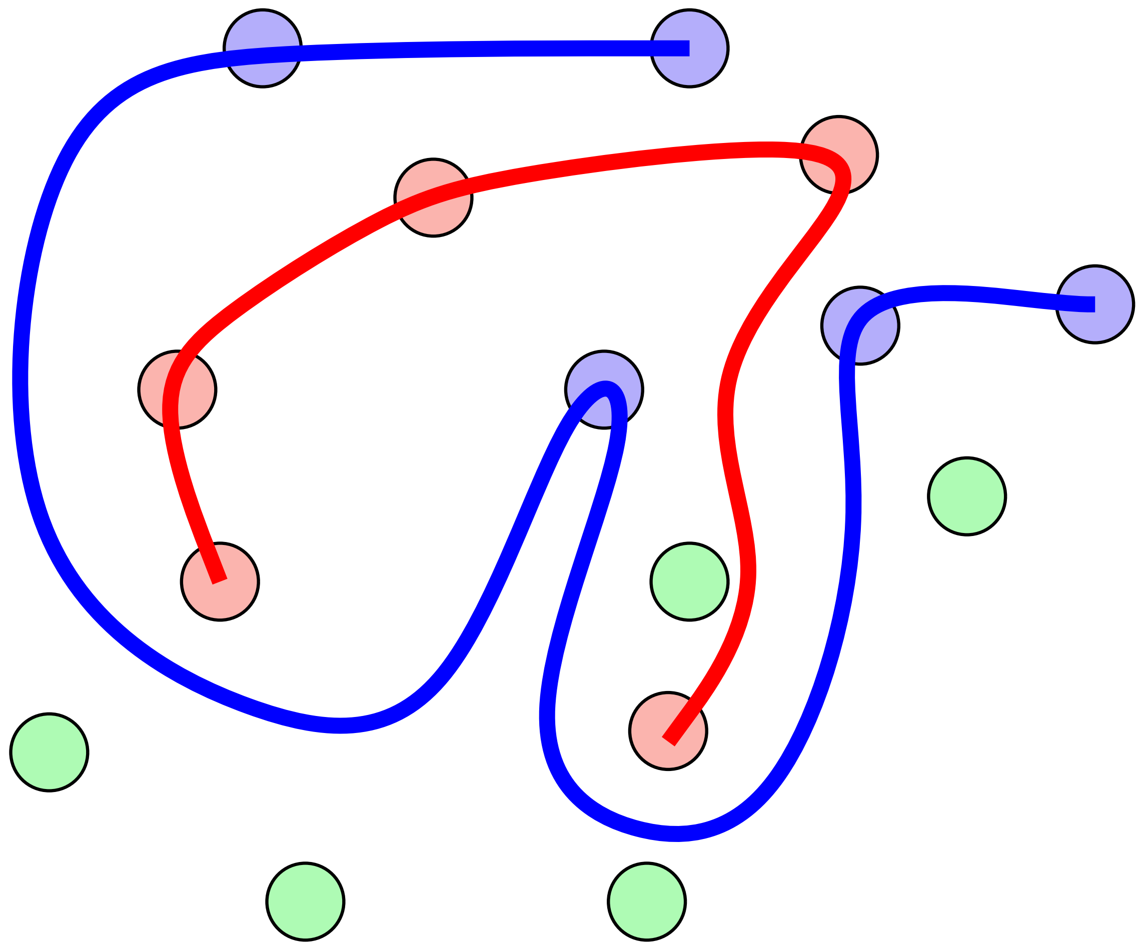
There is always a solution...



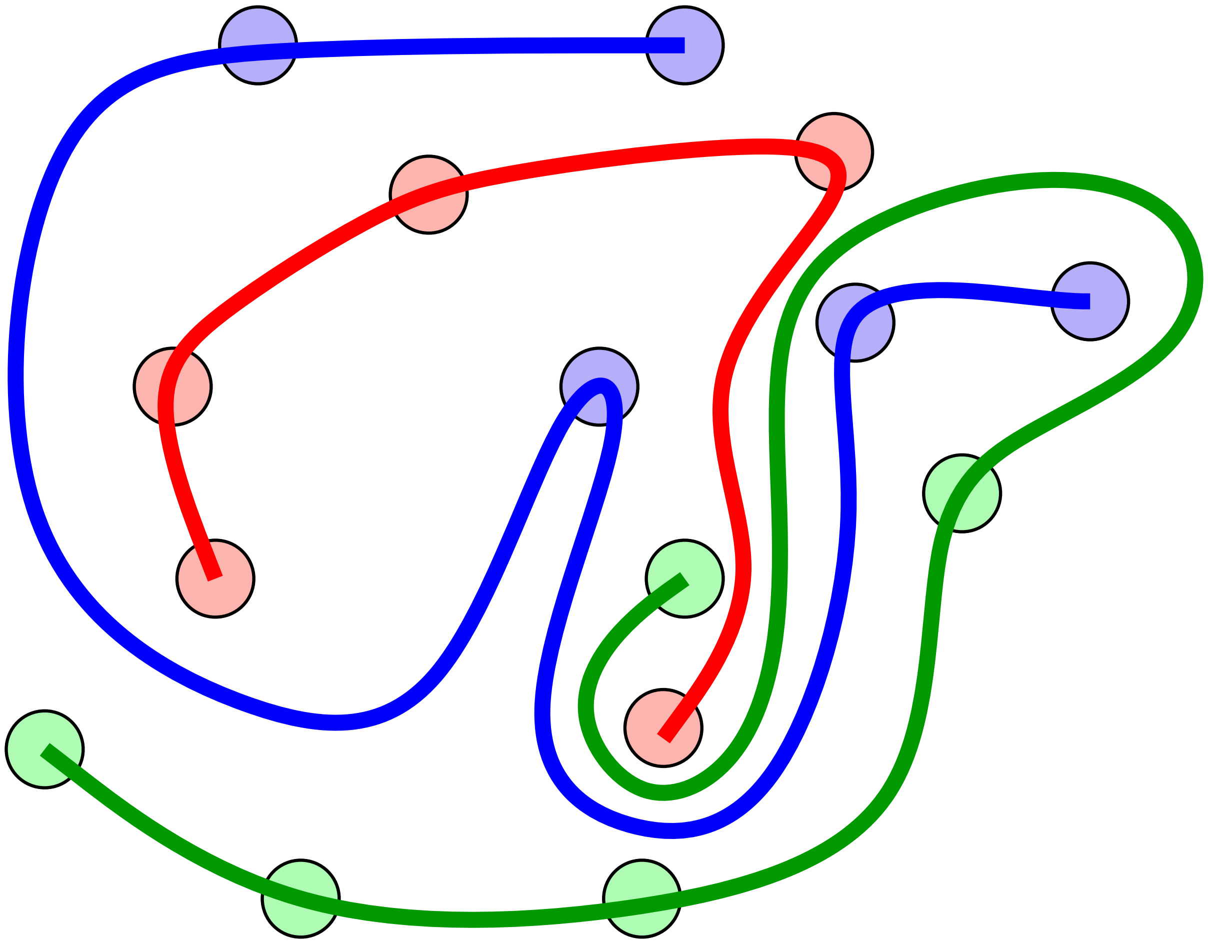
There is always a solution...



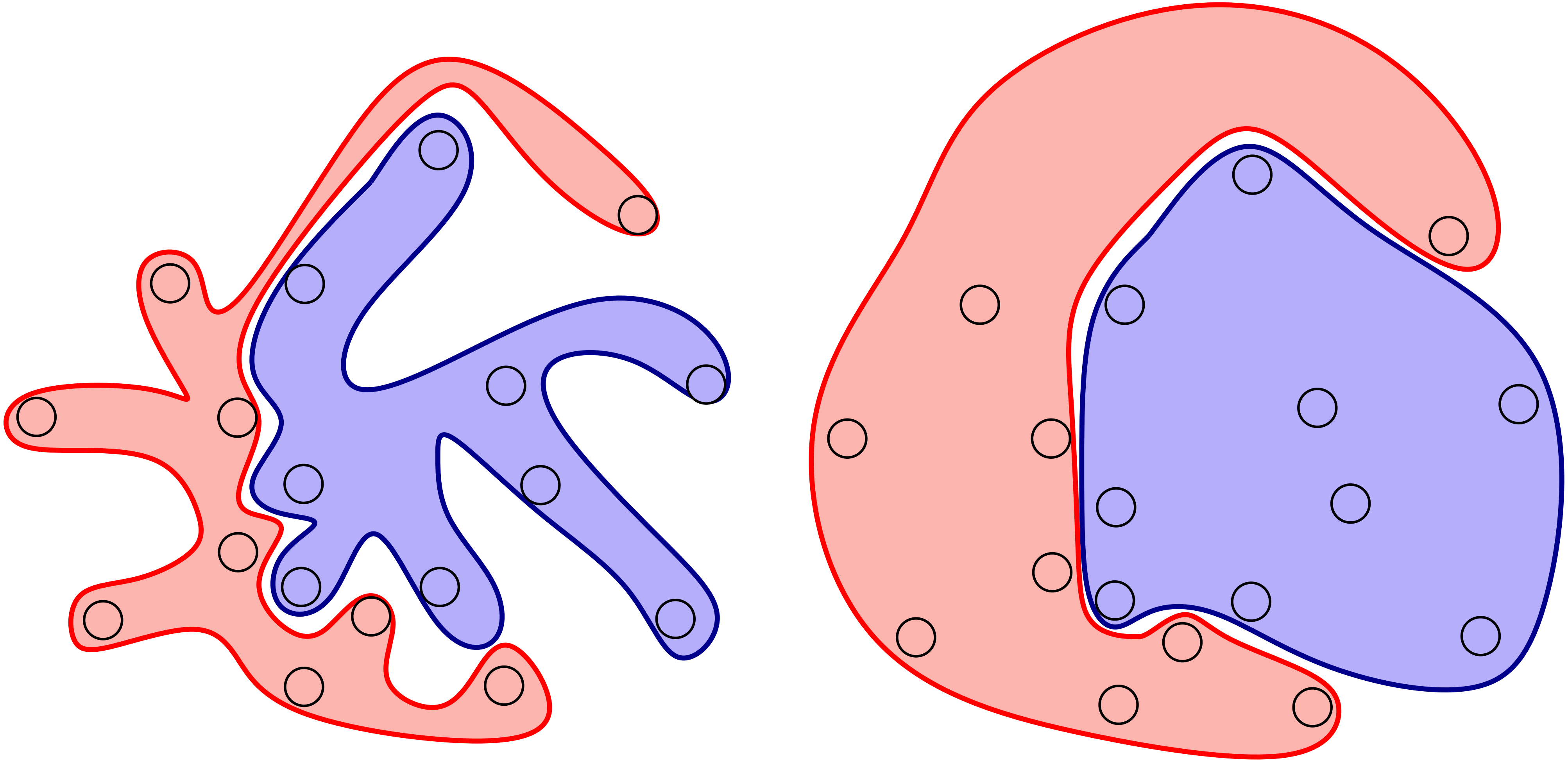
There is always a solution...



There is always a solution...



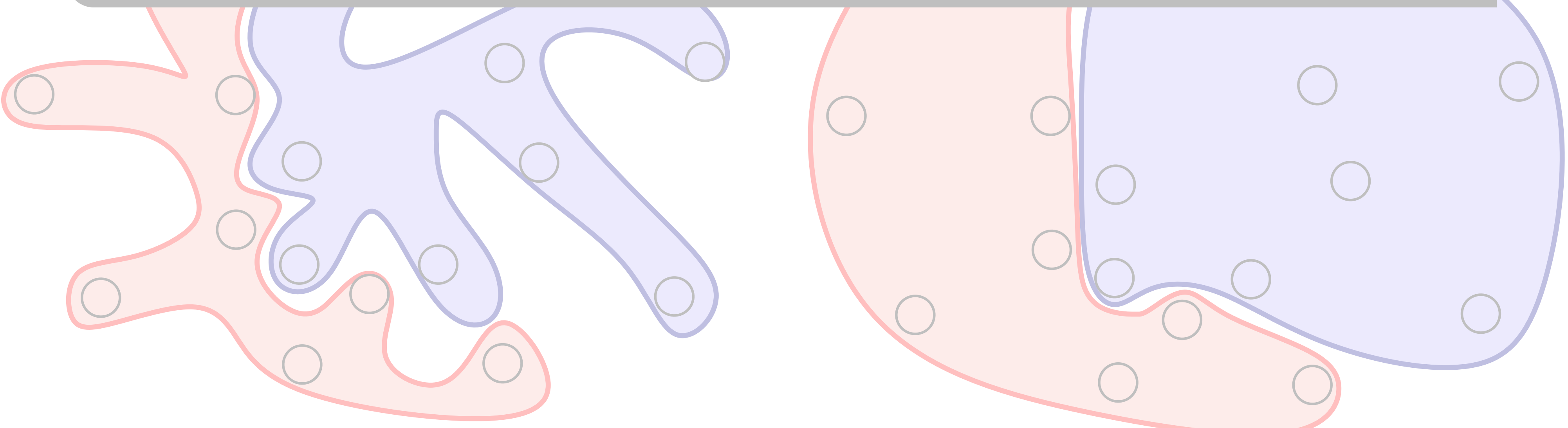
...but not all look the same!



...but not all look the same!

Main Question

How to construct *disjoint contigous* regions, that are as *convex* as possible?



...but not all look the same!

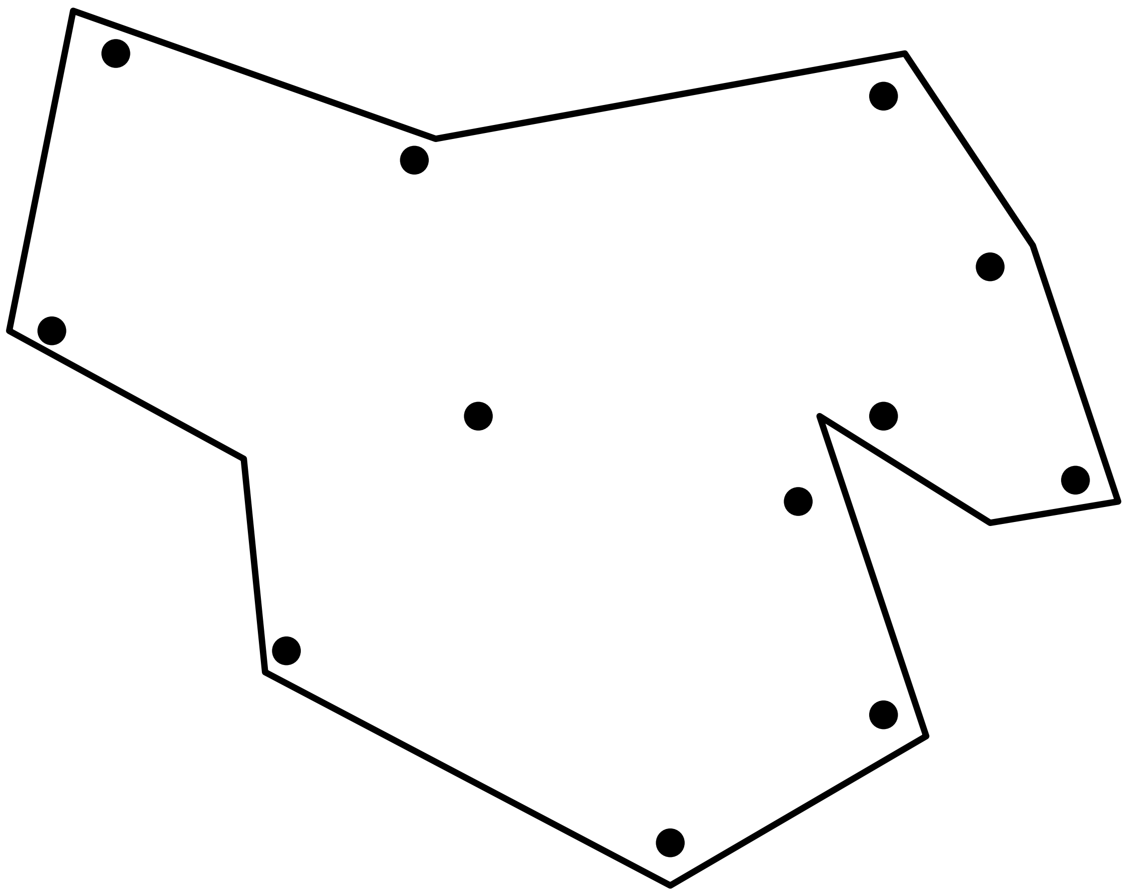
Main Question

How to construct *disjoint contigous* regions, that are as *convex* as possible?

Result

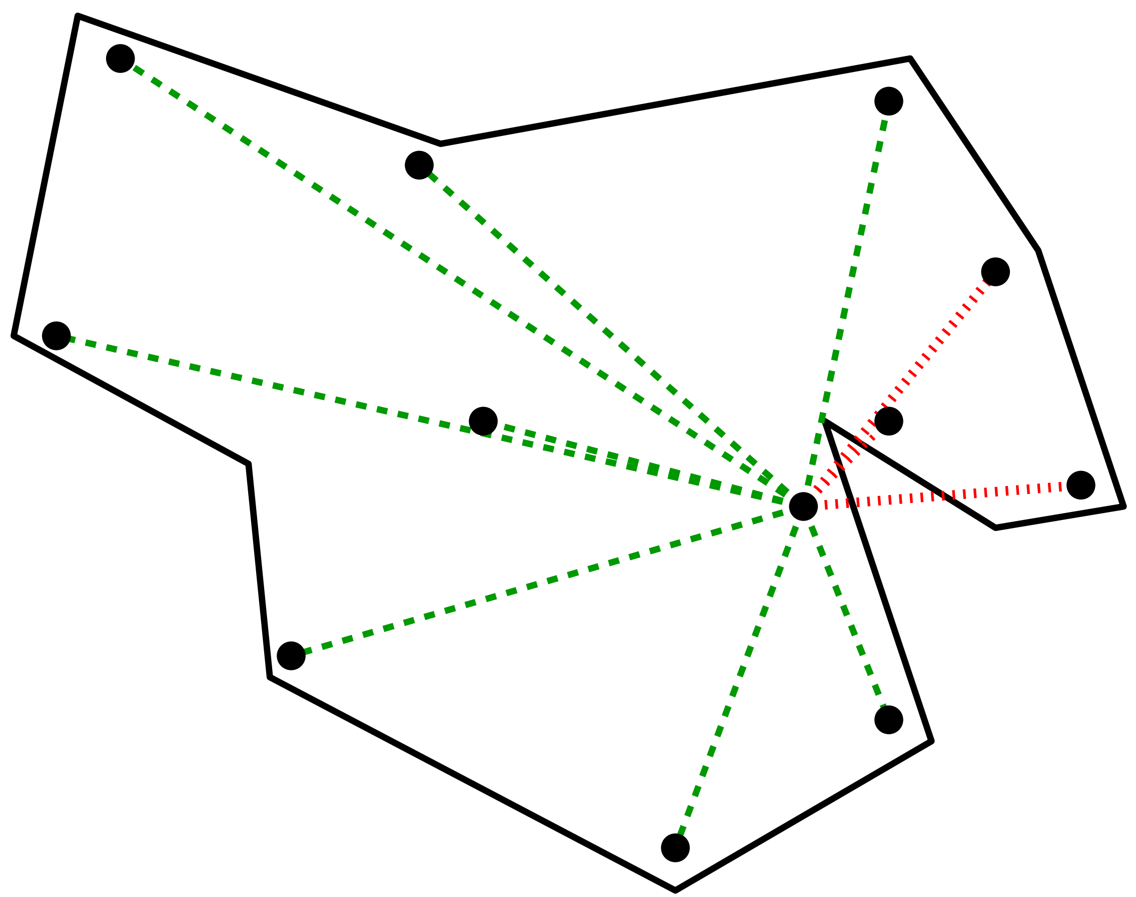
MapSets:
– available at `http://gmap.cs.arizona.edu`
– guarantees non-fragmented non-overlapping regions
– based on a novel geometric problem aiming at optimizing convexity

How to measure *convexity*?



How to measure *convexity*?

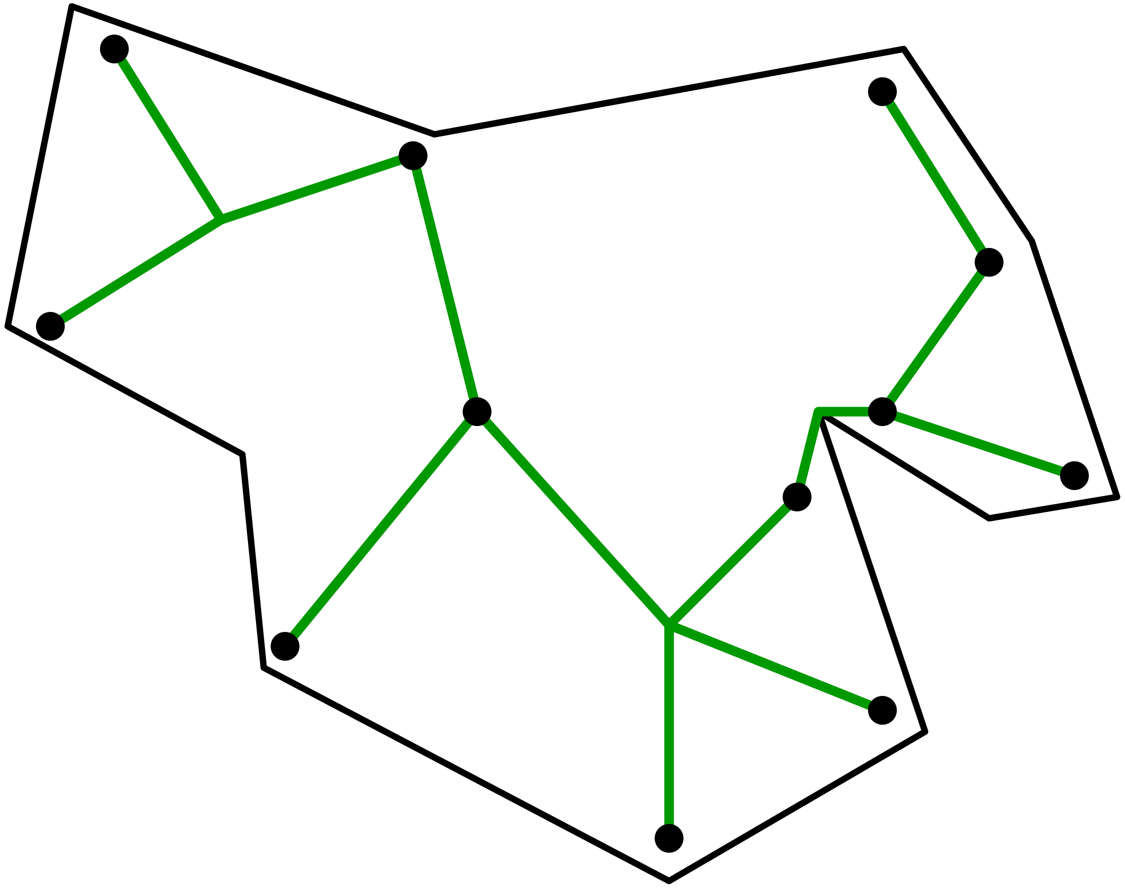
Def.(*visibility-based*): how many points "see" each other



How to measure *convexity*?

Def.(*visibility-based*): how many points “see” each other

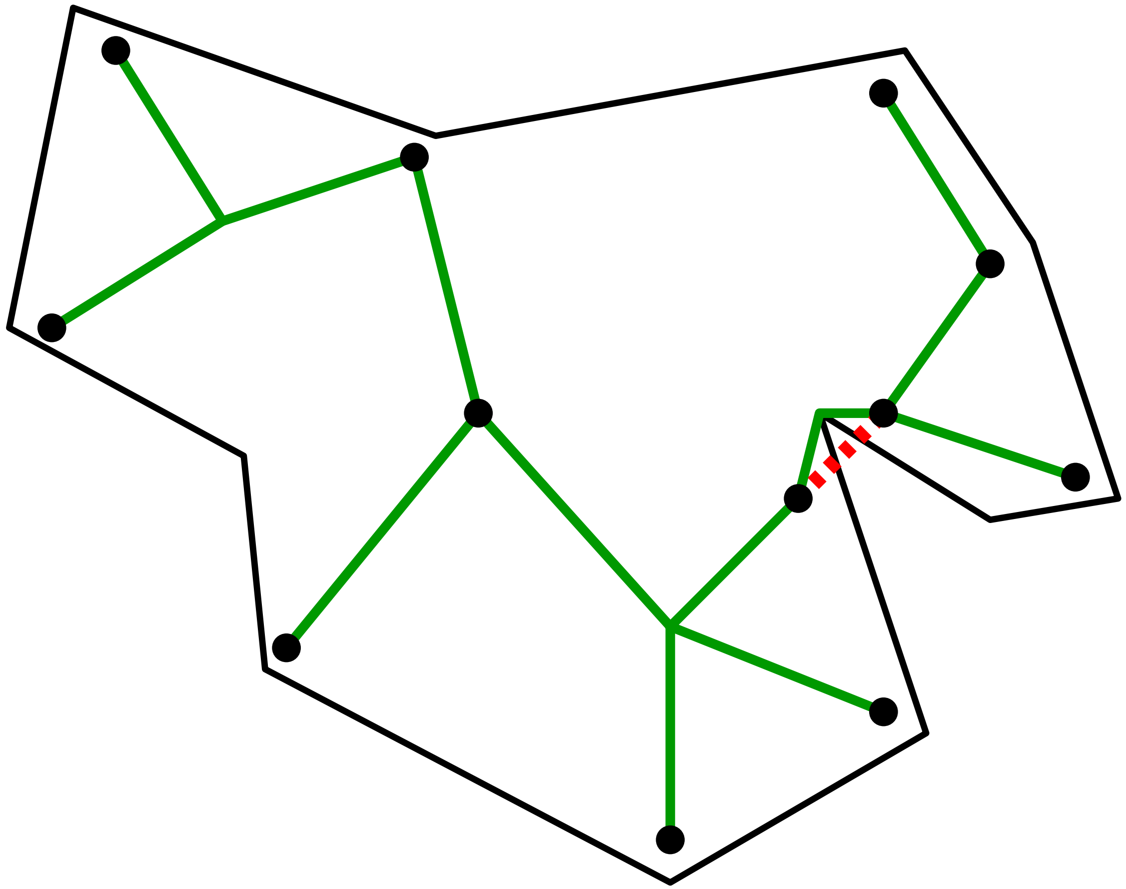
Def.(*ink-based*): length of the shortest spanning tree inside the polygon



How to measure *convexity*?

Def.(*visibility-based*): how many points "see" each other

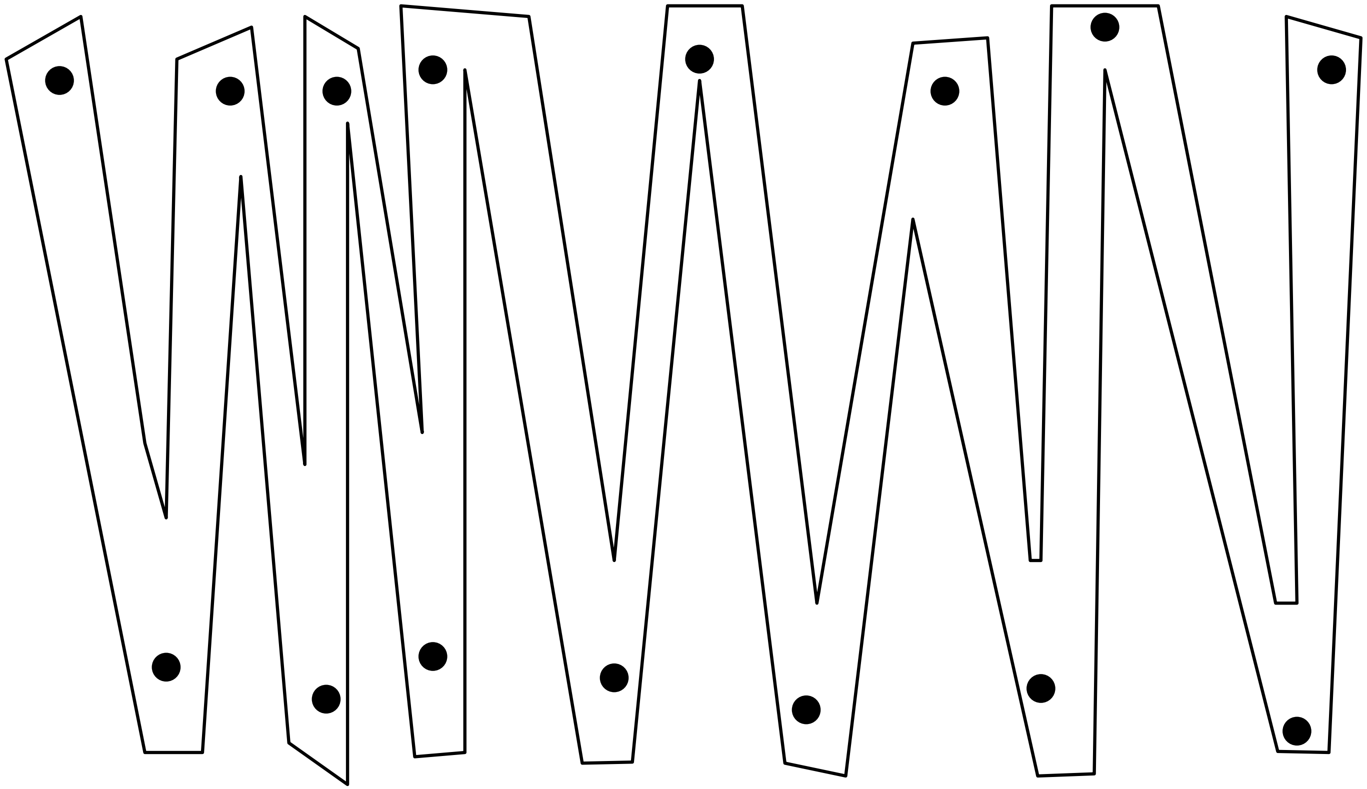
Def.(*ink-based*): length of the shortest spanning tree inside the polygon



How to measure *convexity*?

Def.(*visibility-based*): how many points “see” each other

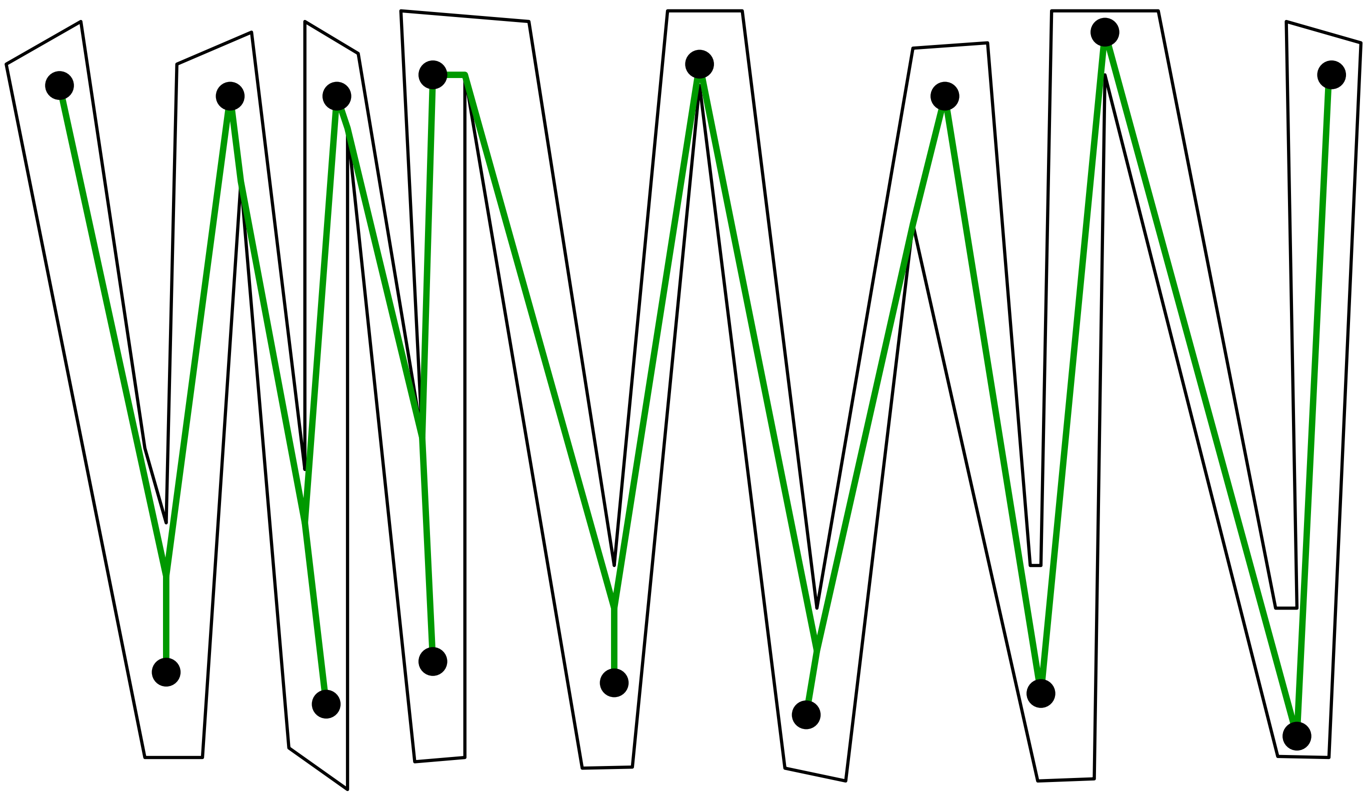
Def.(*ink-based*): length of the shortest spanning tree inside the polygon



How to measure *convexity*?

Def.(*visibility-based*): how many points "see" each other

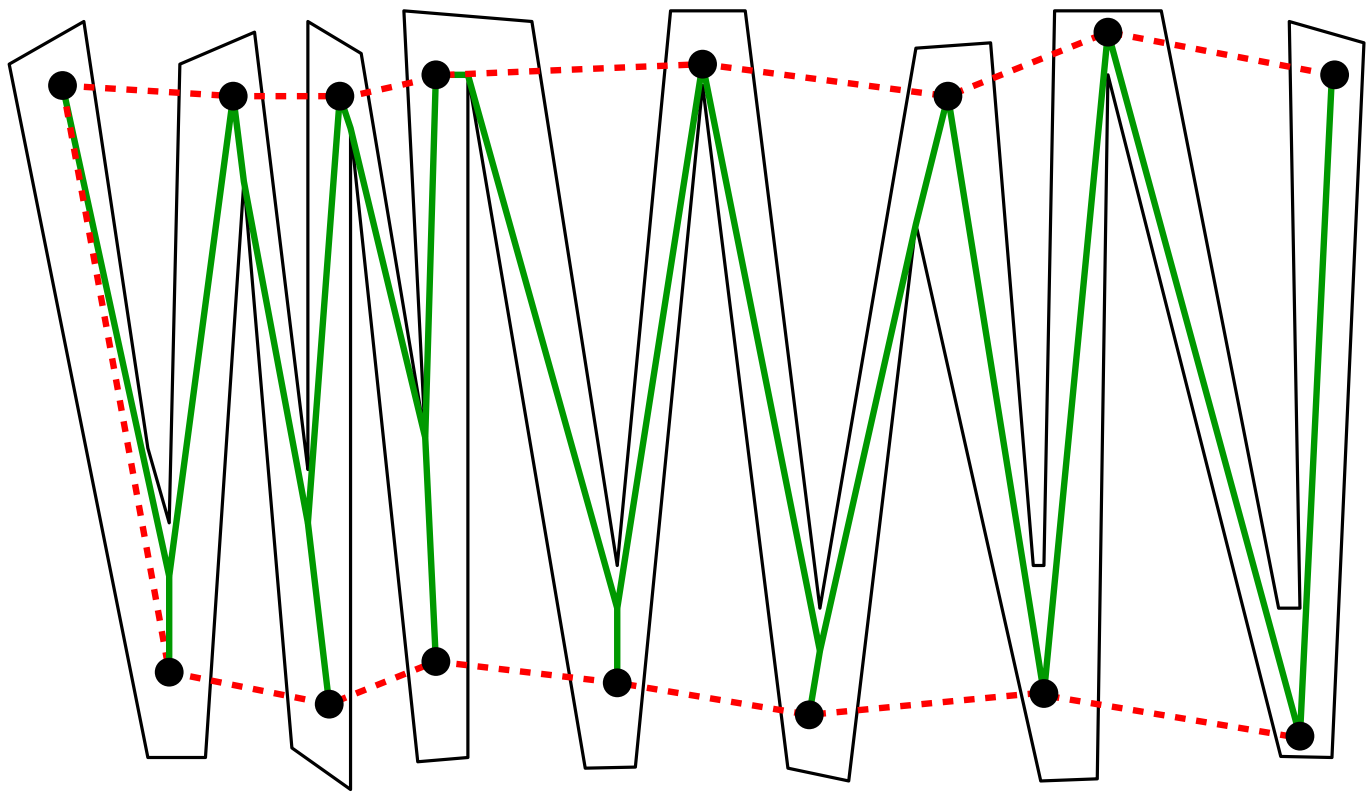
Def.(*ink-based*): length of the shortest spanning tree inside the polygon



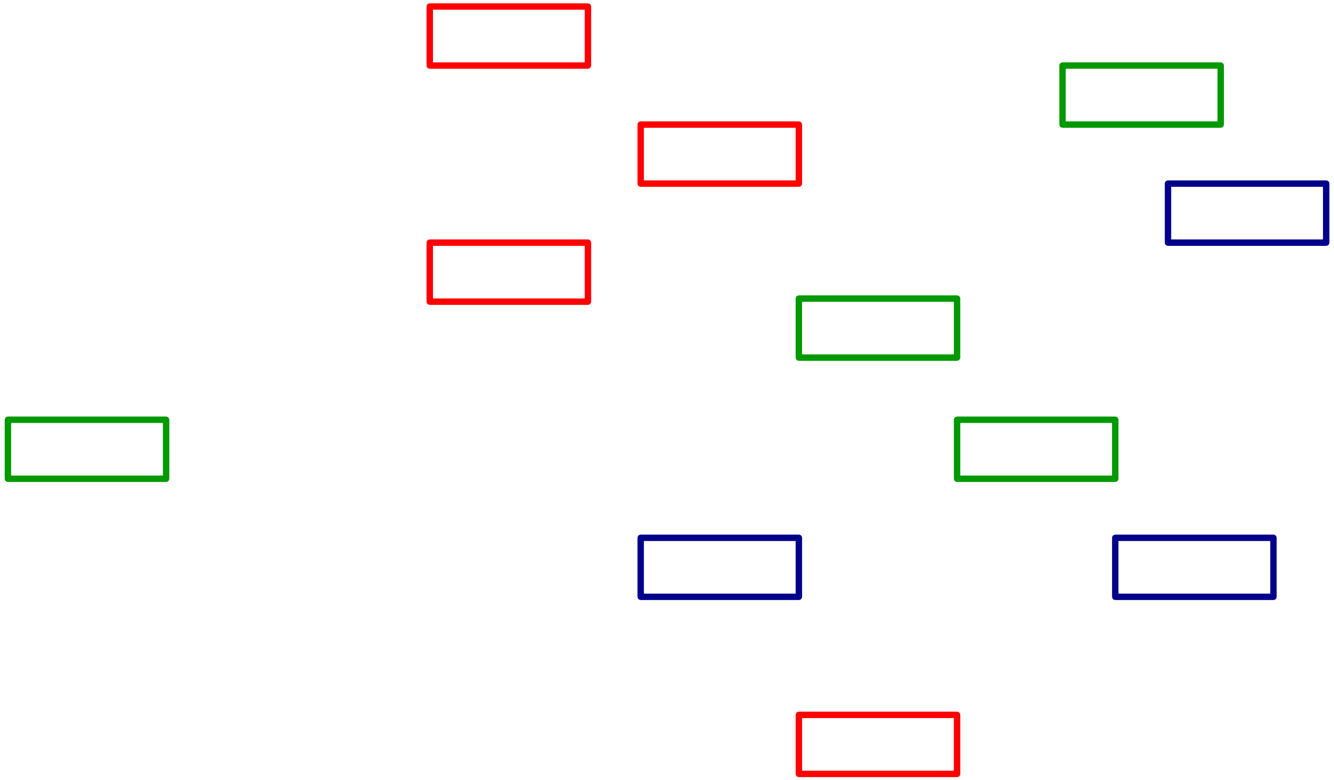
How to measure *convexity*?

Def.(*visibility-based*): how many points "see" each other

Def.(*ink-based*): length of the shortest spanning tree inside the polygon

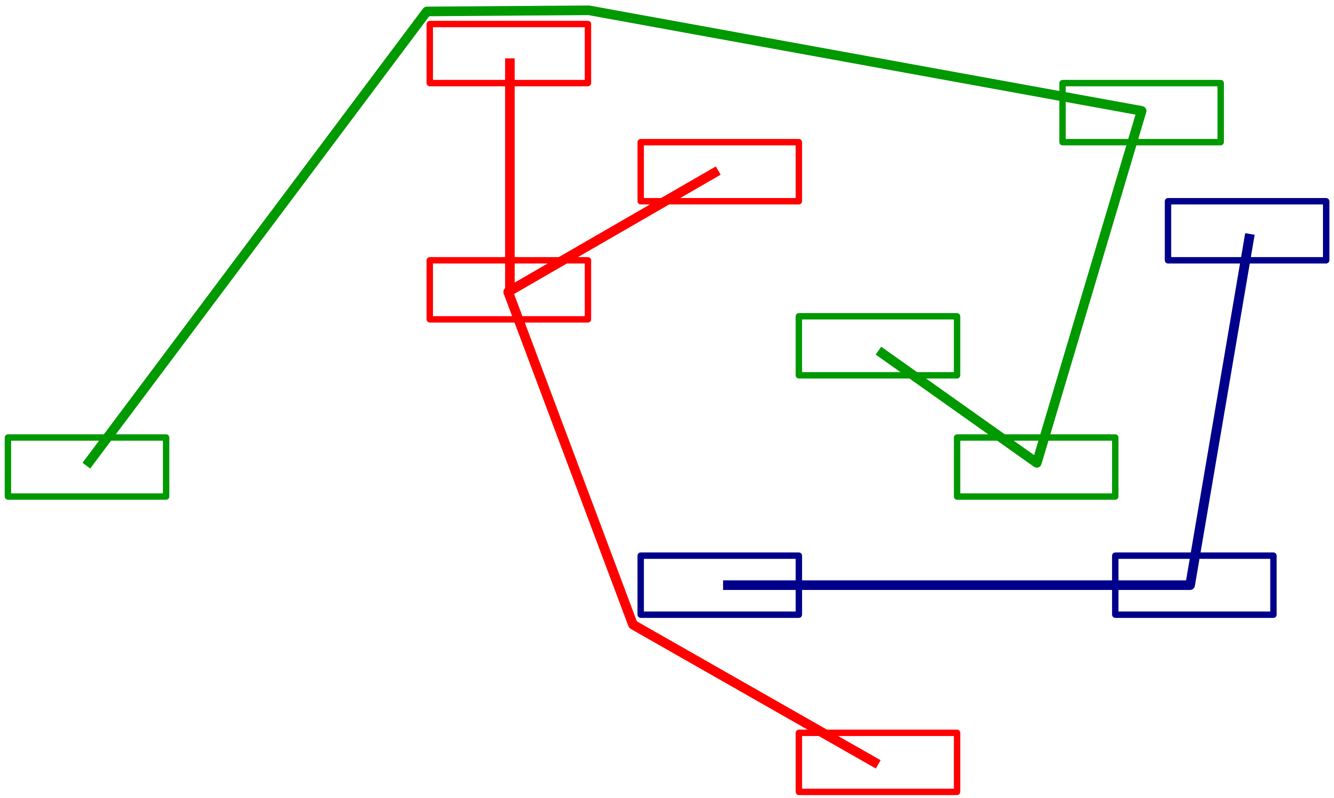


Input

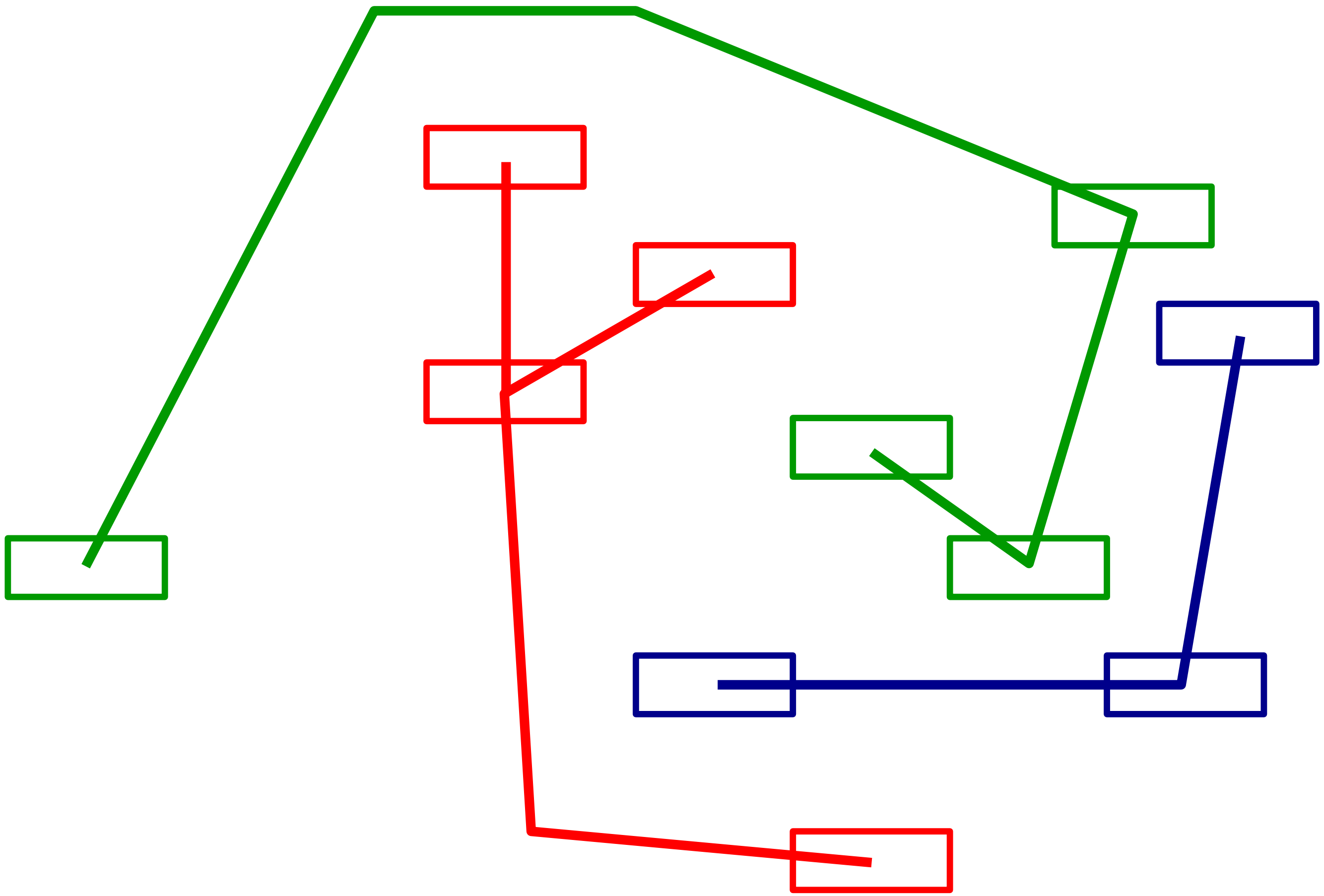


Step 1: Tree Construction

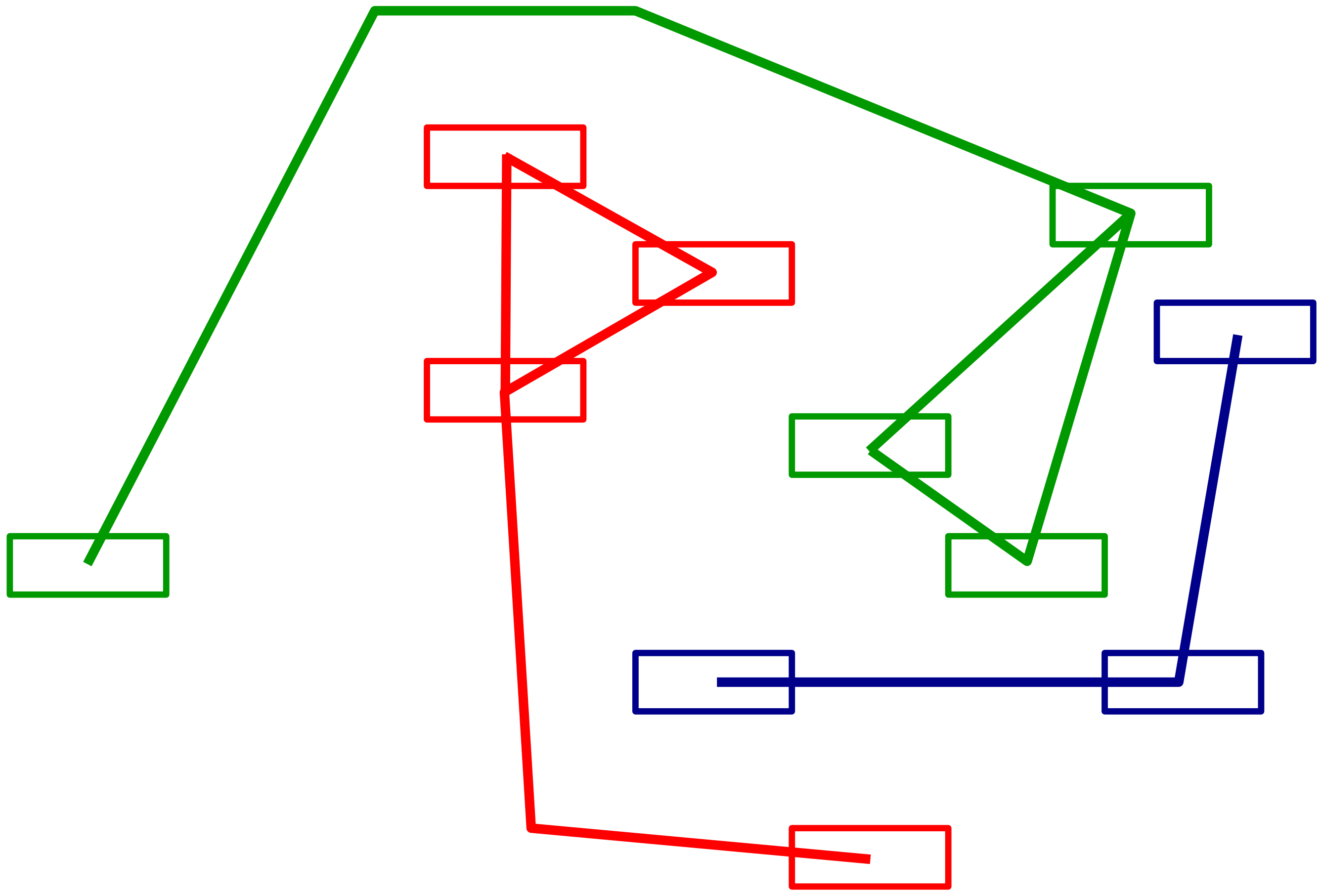
(optimizing ink-based convexity)



Step 2: Force-directed Adjustment



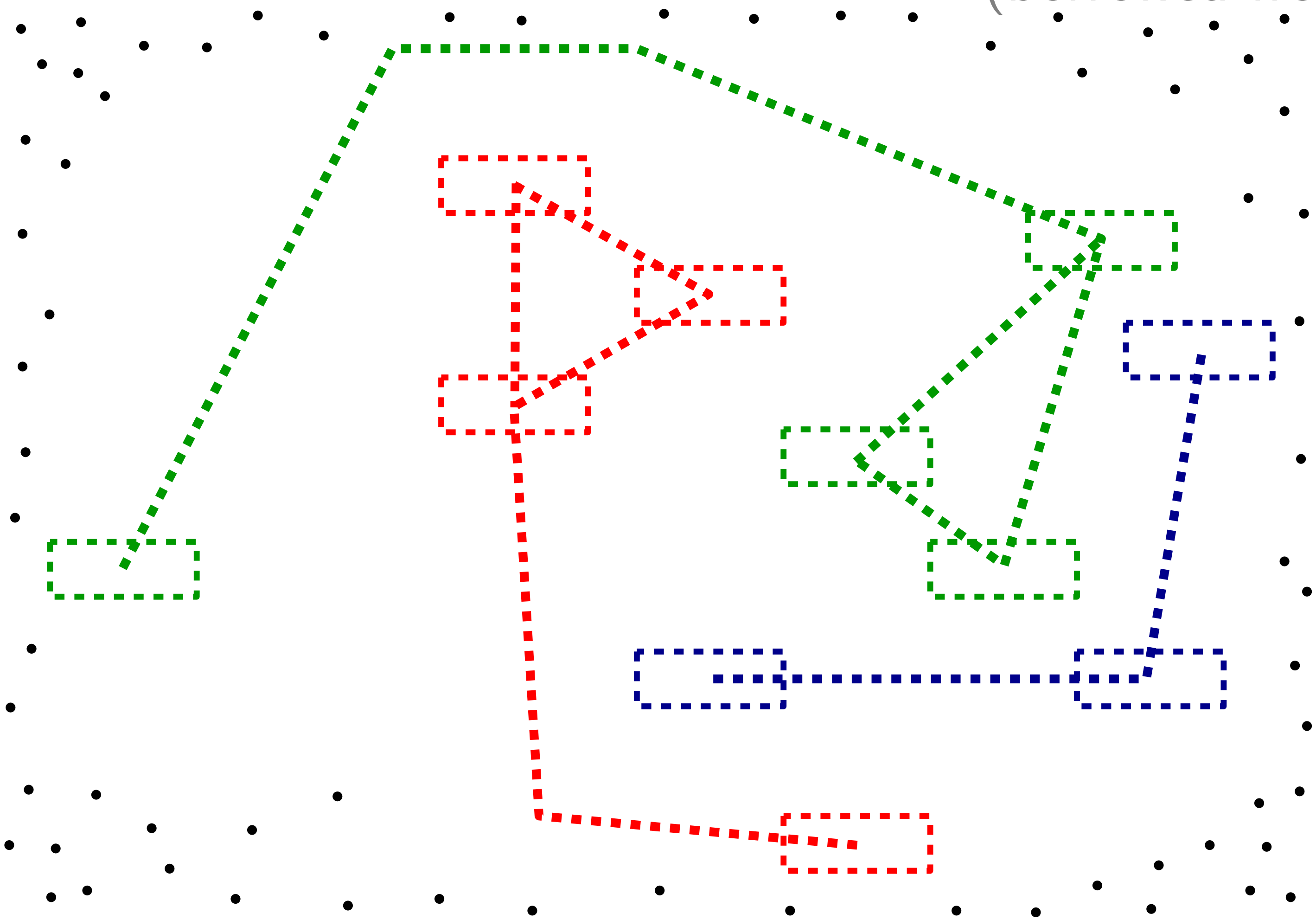
Step 3: Edge Augmentation
(optimizing visibility-based convexity)



MapSets http://gmap.cs.arizona.edu

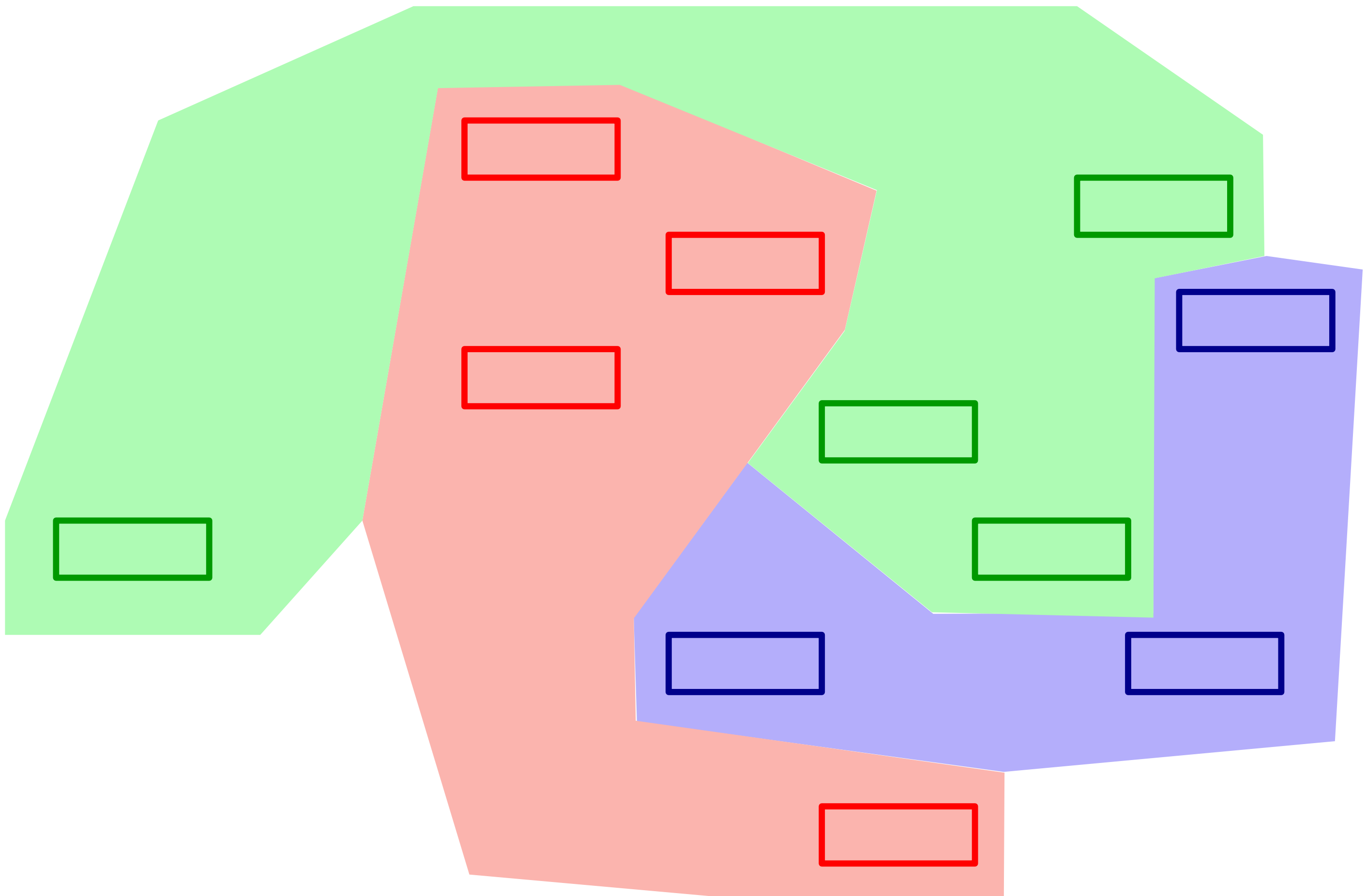
Step 4: Adding Dummy Points

(borrowed from GMap)



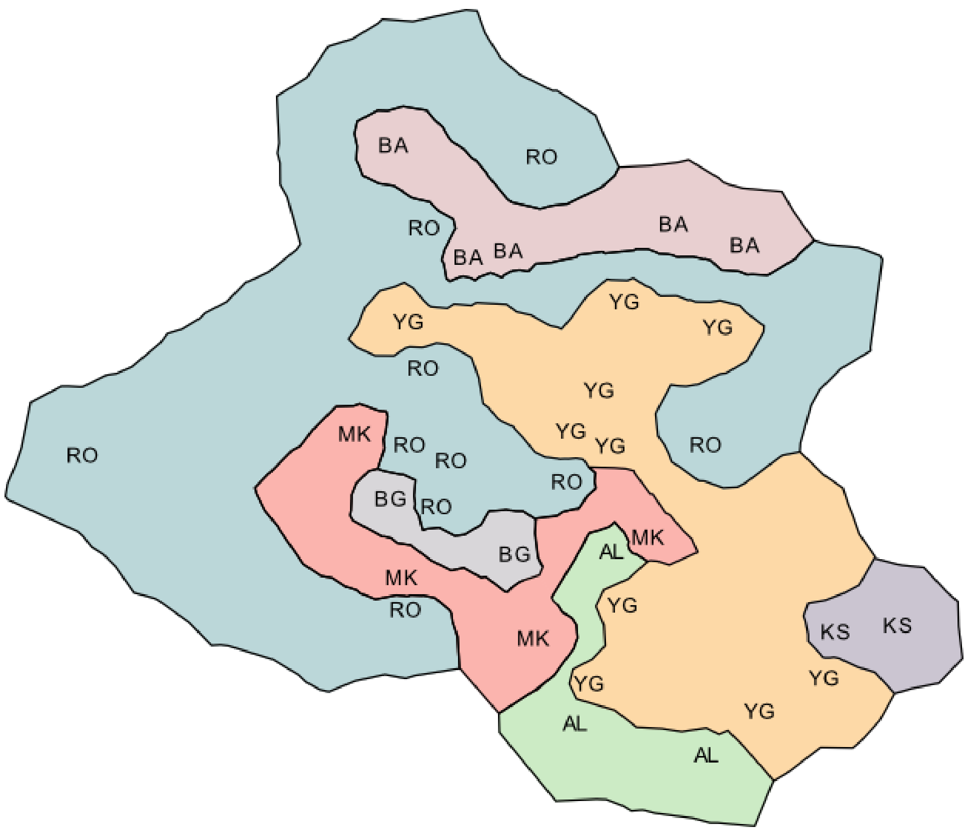
Step 5: Computing Regions

(borrowed from GMap)

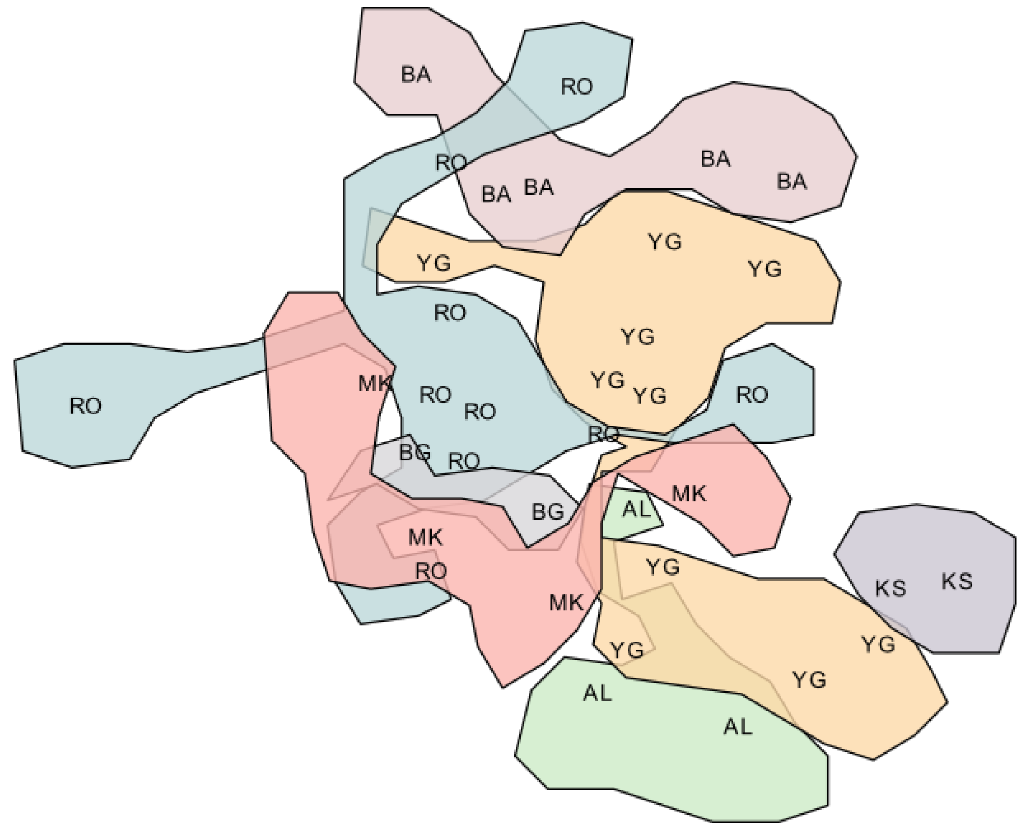


Examples

MapSets



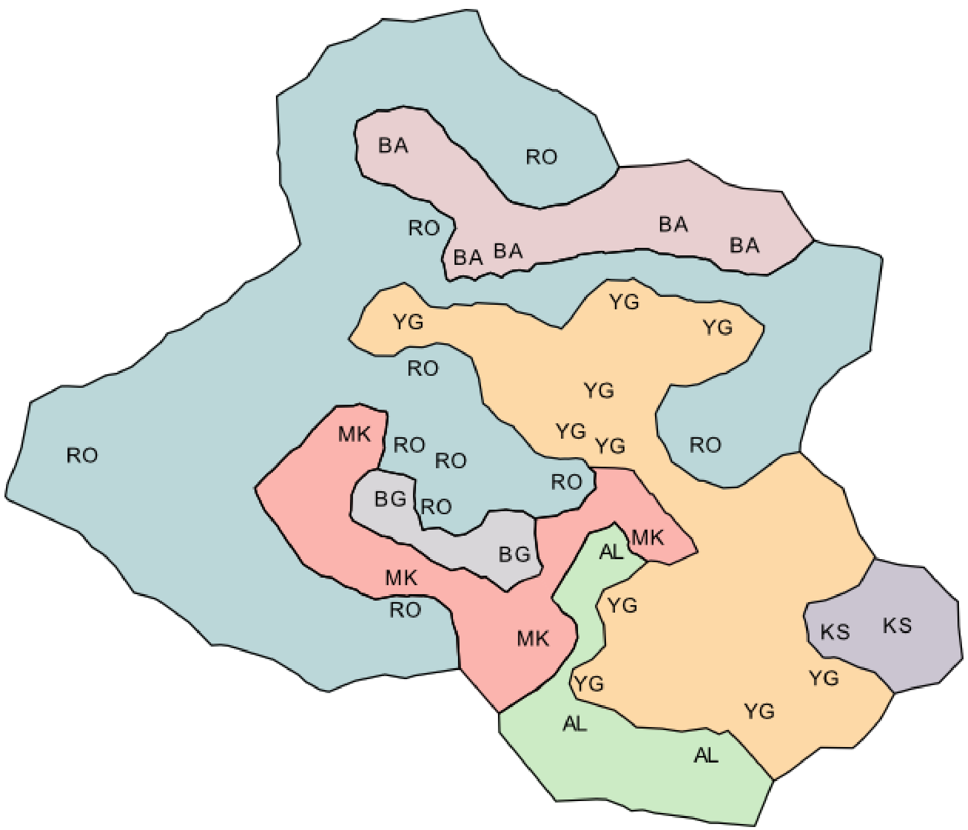
BubbleSets



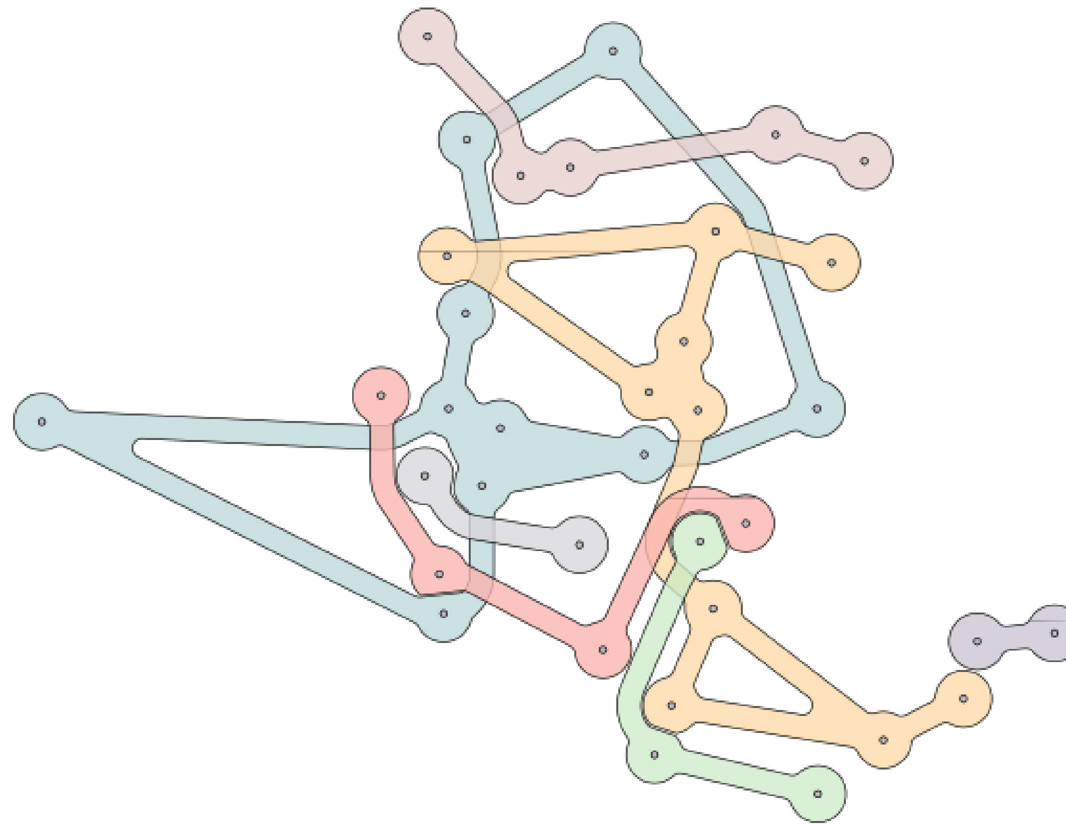
Dataset: genetic similarities between individuals in Europe
50 vertices, 7 clusters

Examples

MapSets



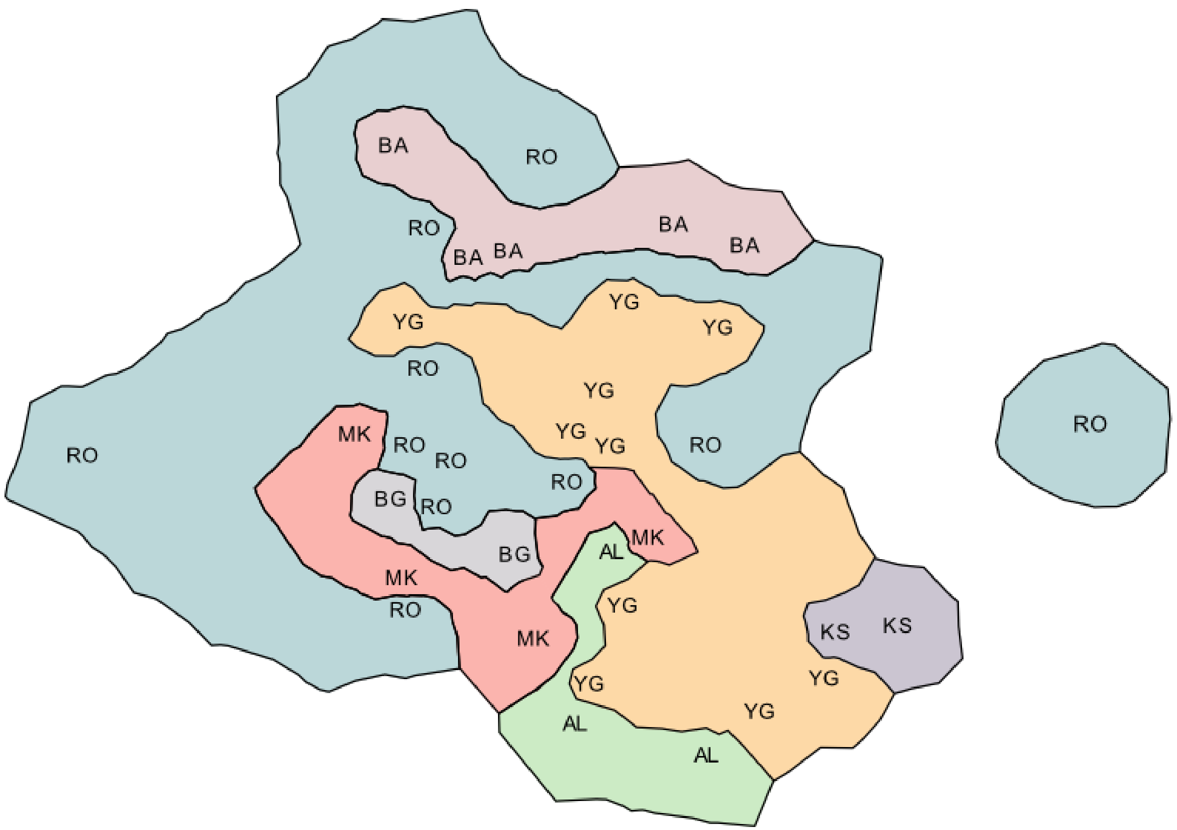
KelpFusion



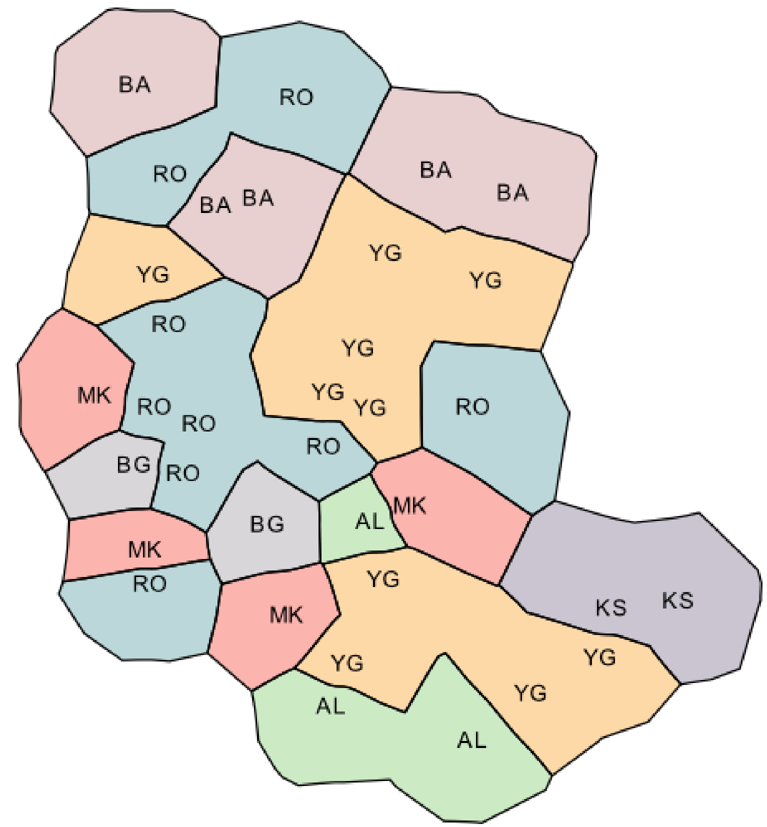
Dataset: genetic similarities between individuals in Europe
50 vertices, 7 clusters

Examples

MapSets



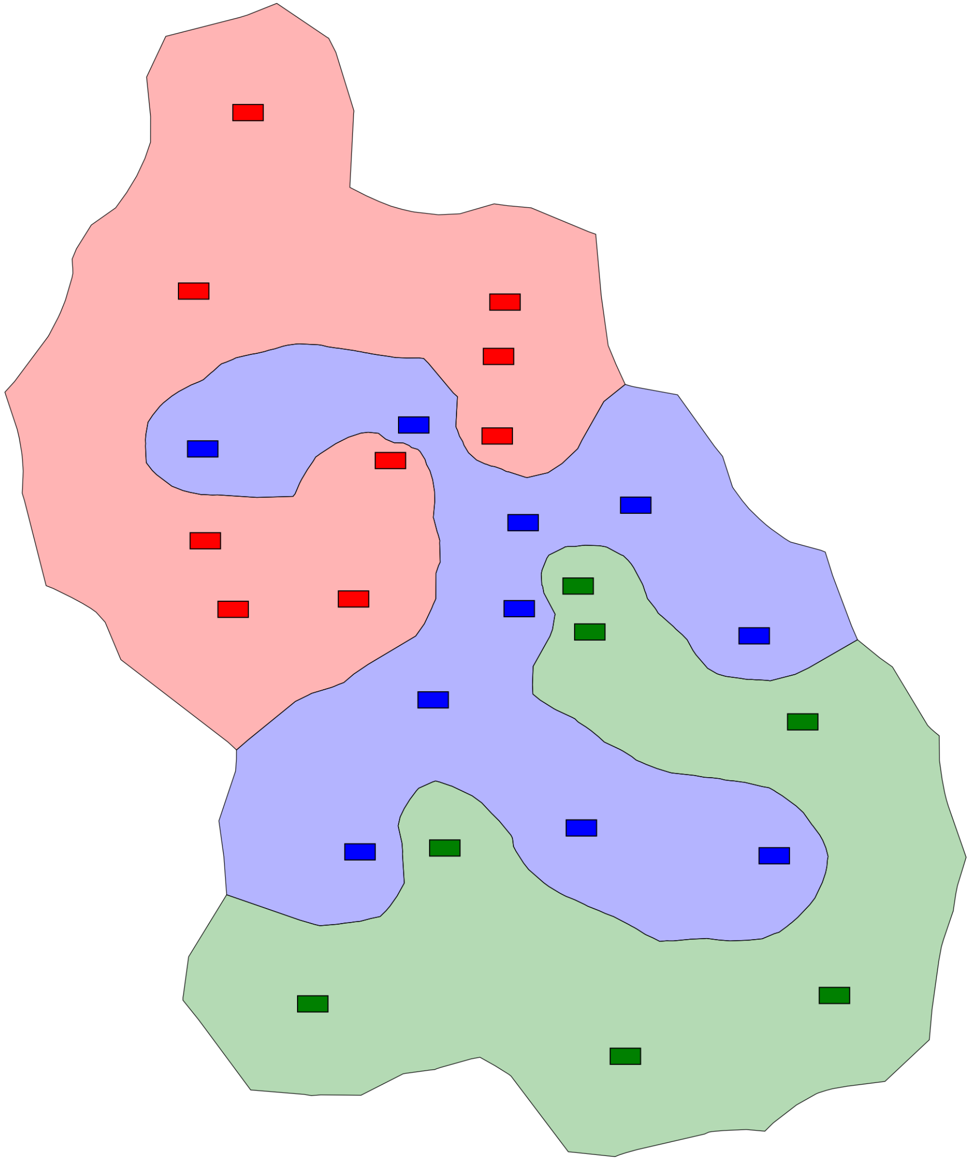
GMap



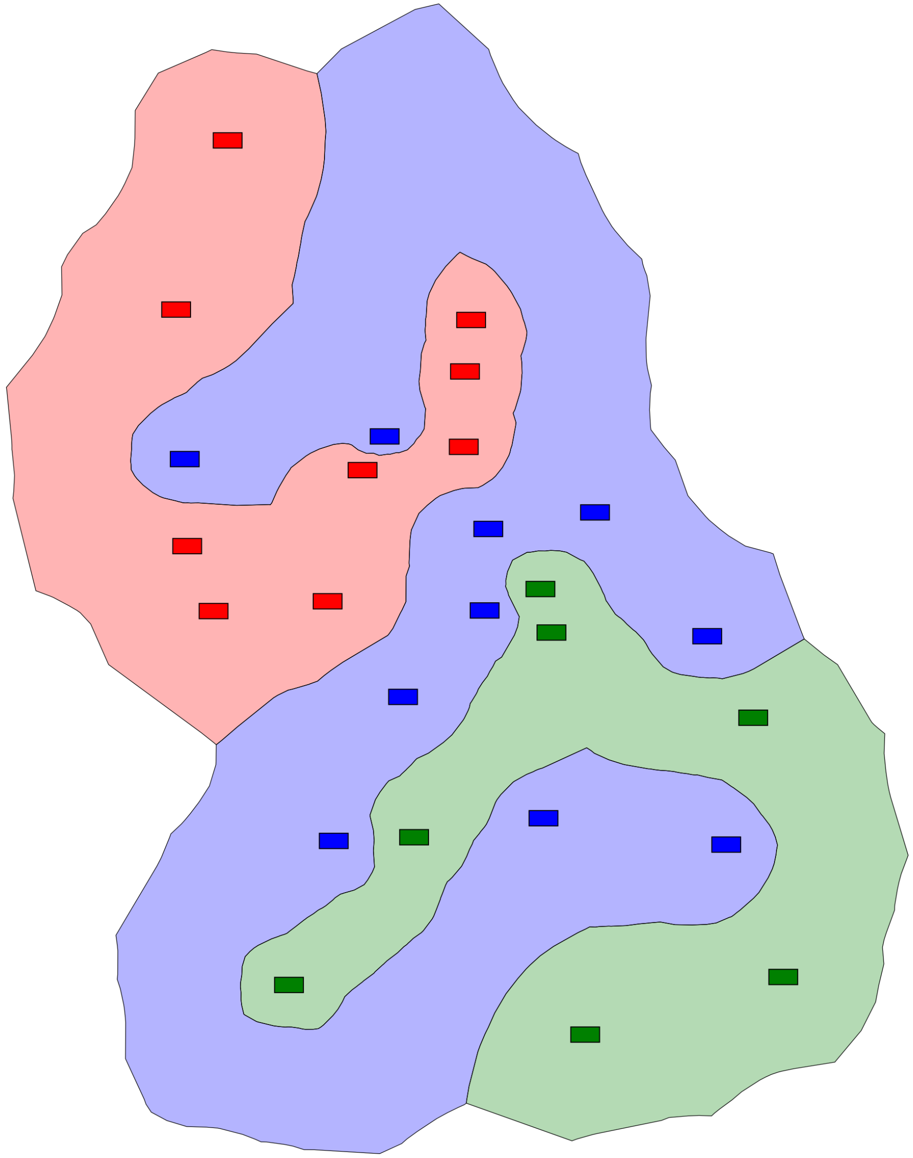
Dataset: genetic similarities between individuals in Europe
50 vertices, 7 clusters

Examples

MapSets

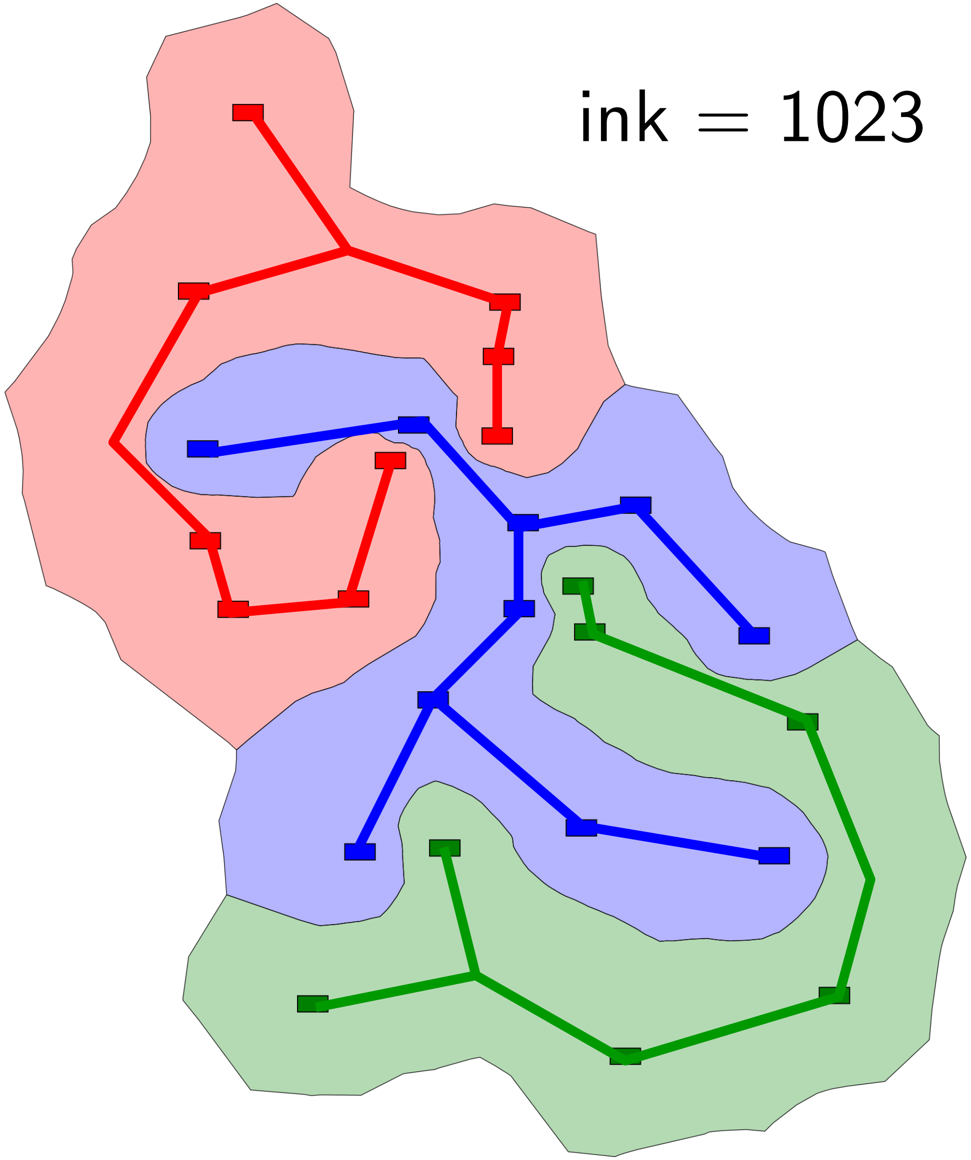


w/o optimizing ink

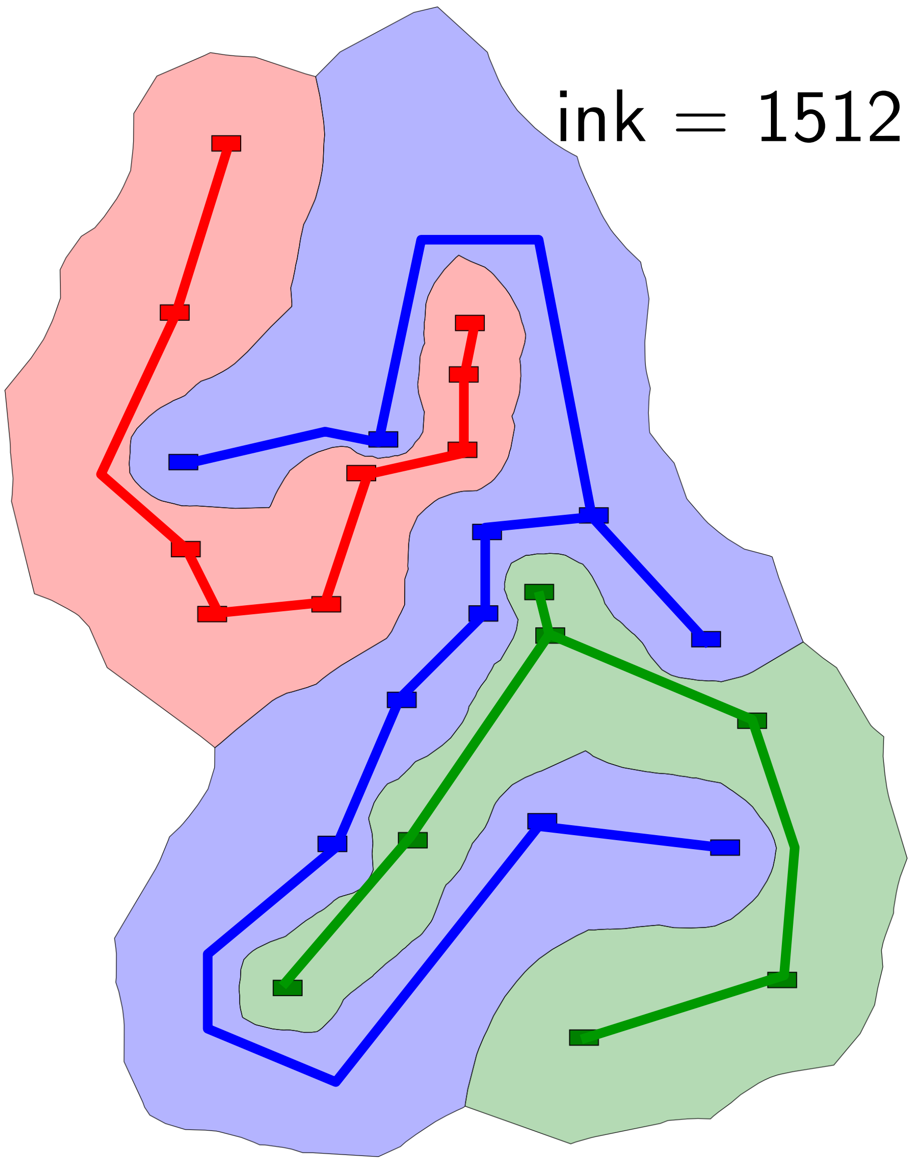


Examples

MapSets

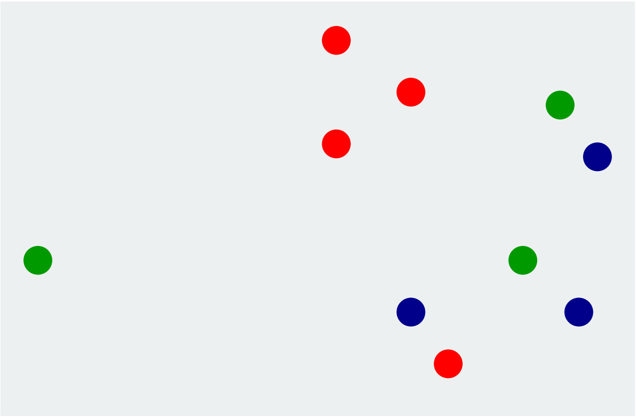


w/o optimizing ink



Colored (Euclidean) Spanning Trees

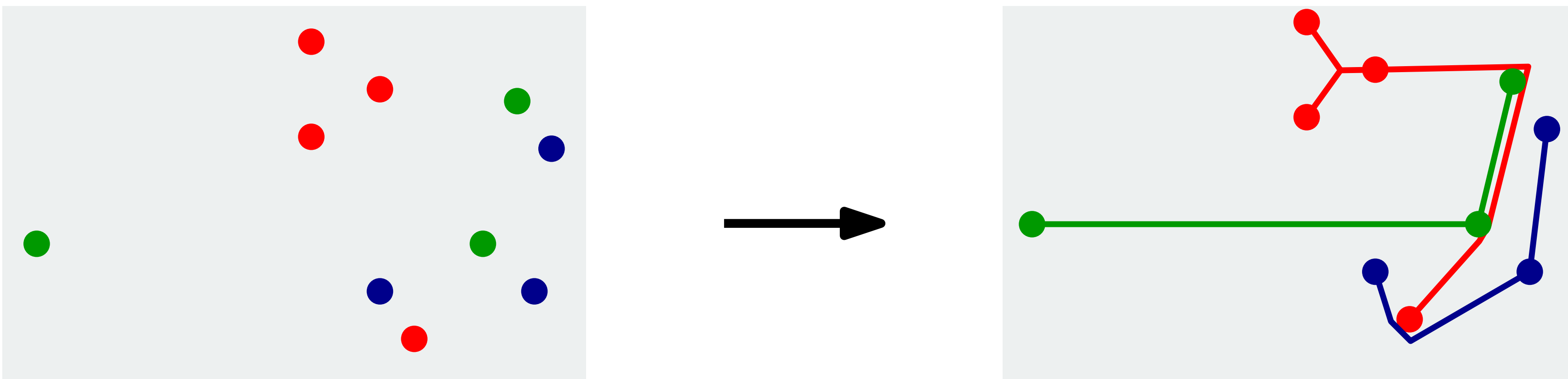
Input



k-colored point set in R^2

Colored (Euclidean) Spanning Trees

Input | **Output**

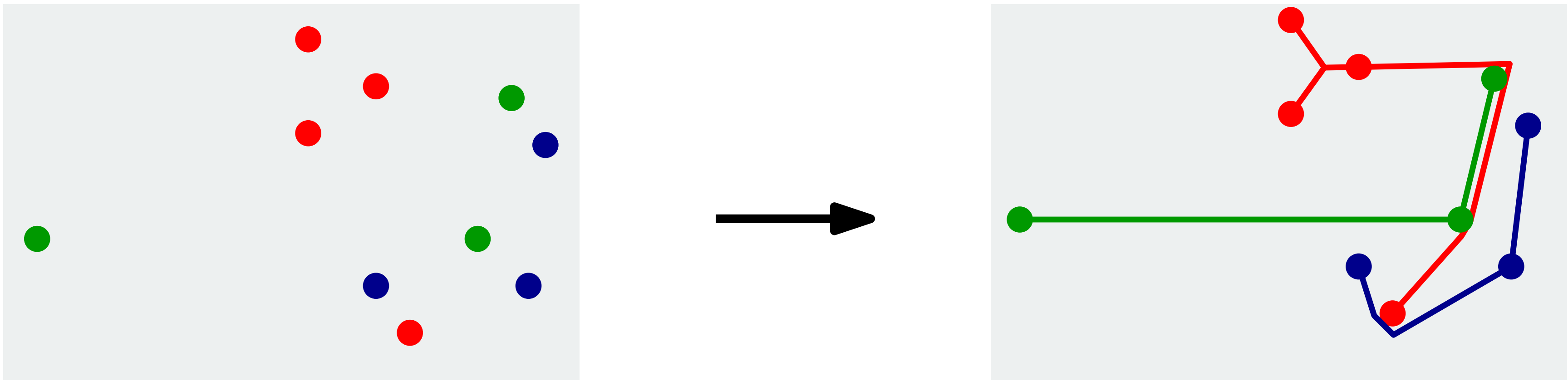


k-colored point set in R^2

k non-crossing Steiner trees

*C*olored (*E*uclidean) *S*panning *T*rees

Input | **Output**



k-colored point set in R^2 | k non-crossing Steiner trees

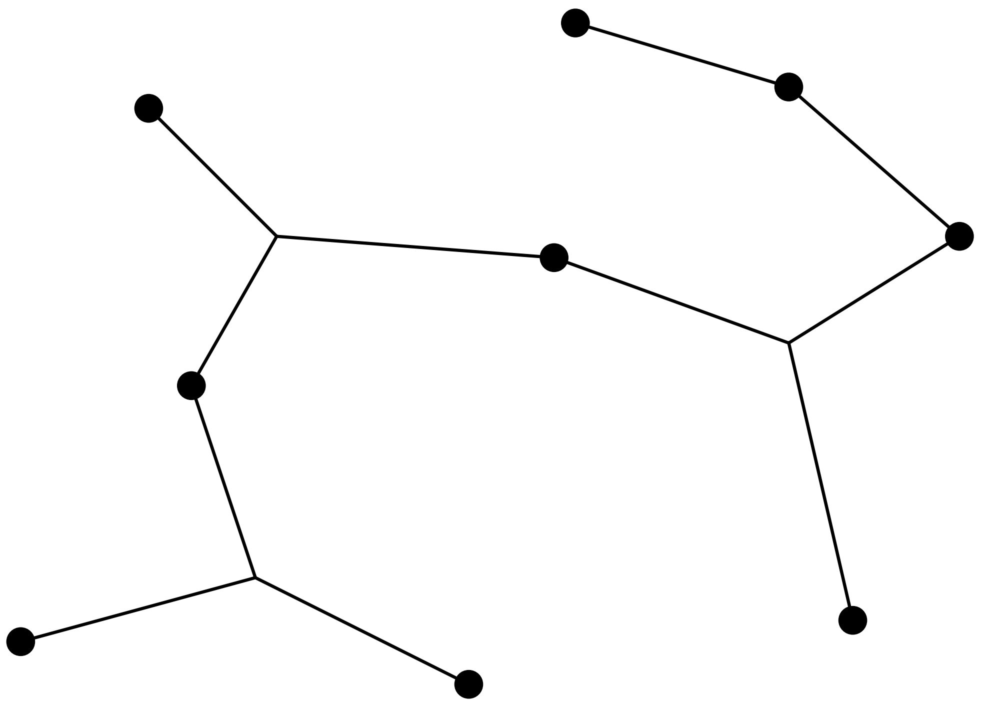
CST: Minimize total length!

*C*olored (*E*uclidean) *S*panning *T*rees

Observation 1 CST is NP-hard

*C*olored (*E*uclidean) *S*panning *T*rees

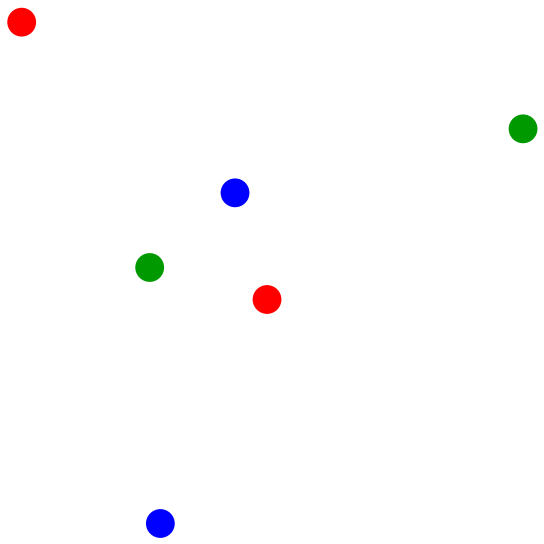
Observation 1 CST is NP-hard, even if $k = 1$



*C*olored (*E*uclidean) *S*panning *T*rees

Observation 1 CST is NP-hard, even if $k = 1$

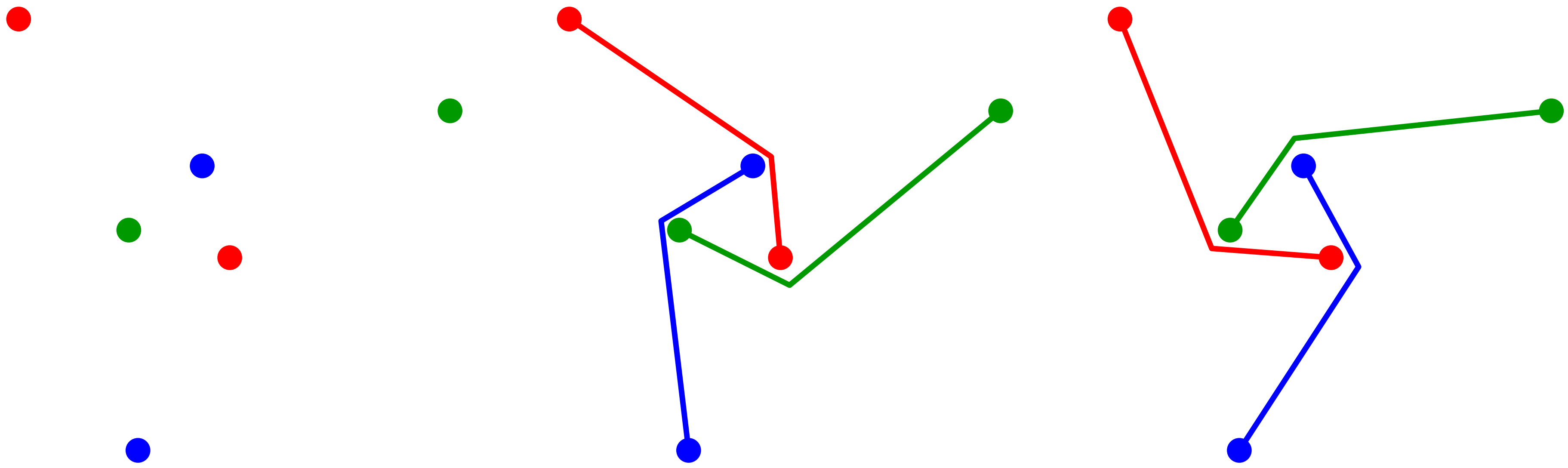
Observation 2 CST is NP-hard, even if
– Steiner points are not allowed
– every cluster consists of *two* points [Bastert Fekete, TR'96]



*C*olored (*E*uclidean) *S*panning *T*rees

Observation 1 CST is NP-hard, even if $k = 1$

Observation 2 CST is NP-hard, even if
– Steiner points are not allowed
– every cluster consists of *two* points [Bastert Fekete, TR'96]



*C*olored (*E*uclidean) *S*panning *T*rees

Observation 1 CST is NP-hard, even if $k = 1$

Observation 2 CST is NP-hard, even if
– Steiner points are not allowed
– every cluster consists of *two* points [Bastert Fekete, TR'96]

Observation 3 CST (with $k = n/2$) is equivalent to
MIN. LENGTH EMBEDDING OF *Matchings* AT FIXED VERTEX LOCATIONS
[Chan Hoffmann Kiazyk Lubiw, GD'13]

Colored (Euclidean) Spanning Trees

Observation 1 CST is NP-hard, even if $k = 1$

Observation 2 CST is NP-hard, even if
– Steiner points are not allowed
– every cluster consists of *two* points [Bastert Fekete, TR'96]

Observation 3 CST (with $k = n/2$) is equivalent to
MIN. LENGTH EMBEDDING OF *Matchings* AT FIXED VERTEX LOCATIONS
[Chan Hoffmann Kiazyk Lubiw, GD'13]

Theorem (Chan et al.) CST (with $k = n/2$) admits an $O(\sqrt{k} \log k)$-approximation

*C*olored (*E*uclidean) *S*panning *T*rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Colored (Euclidean) Spanning Trees

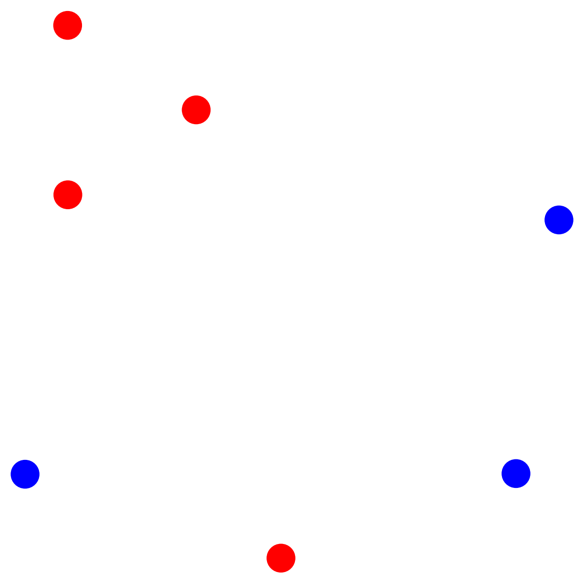
Theorem CST (with k colors) admits a $(k\rho)$-approximation

$1.15 < \rho < 1.22$
Steiner ratio, that is,
$inf\{\frac{|\text{Steiner Tree}|}{|\text{Spanning Tree}|}\}$

*C*olored (*E*uclidean) *S*panning *T*rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):



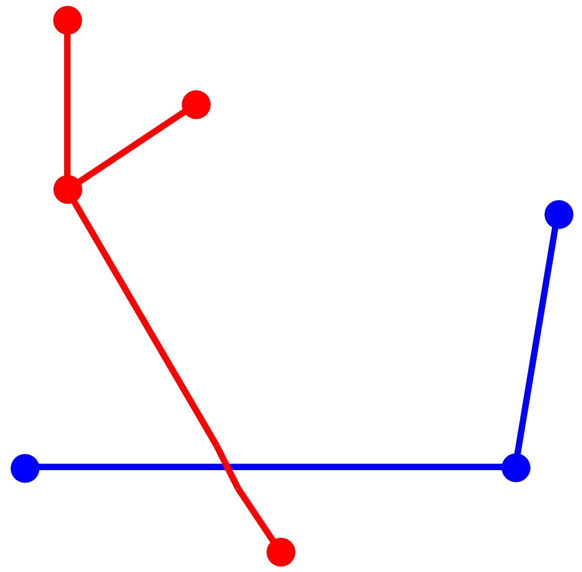
Analysis:

*C*olored (*E*uclidean) *S*panning *T*rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):
– construct red and blue minimum spanning trees

Analysis:



*C*olored (*E*uclidean) *S*panning *T*rees

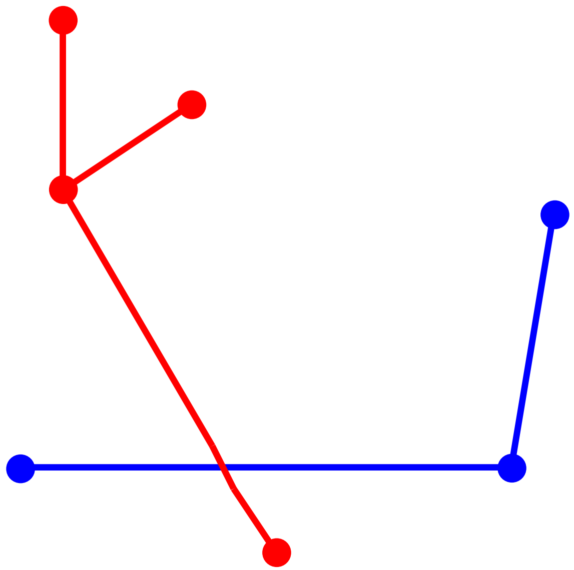
Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one

Analysis:



Colored (Euclidean) Spanning Trees

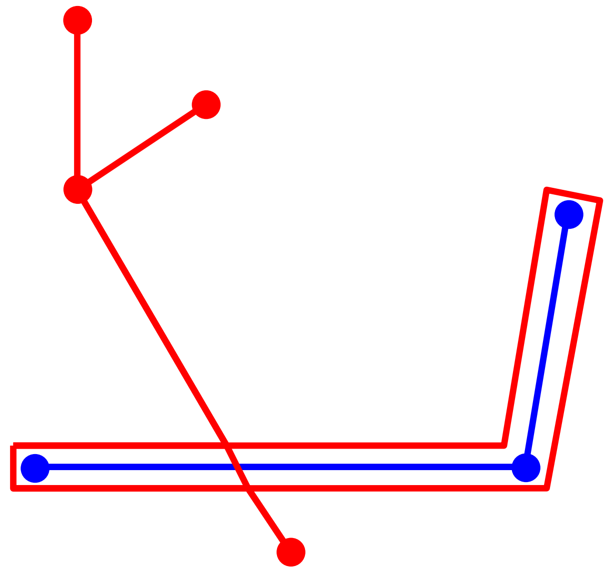
Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

Analysis:



*C*olored (*E*uclidean) *S*panning *T*rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

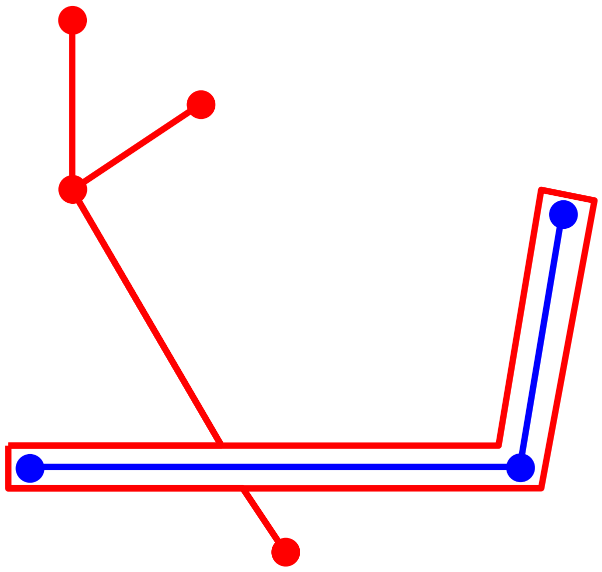
Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a “shell” around it of another color

– remove crossings

Analysis:



$\mathcal{C}$olored ($\mathcal{E}$uclidean) $\mathcal{S}$panning $\mathcal{T}$rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

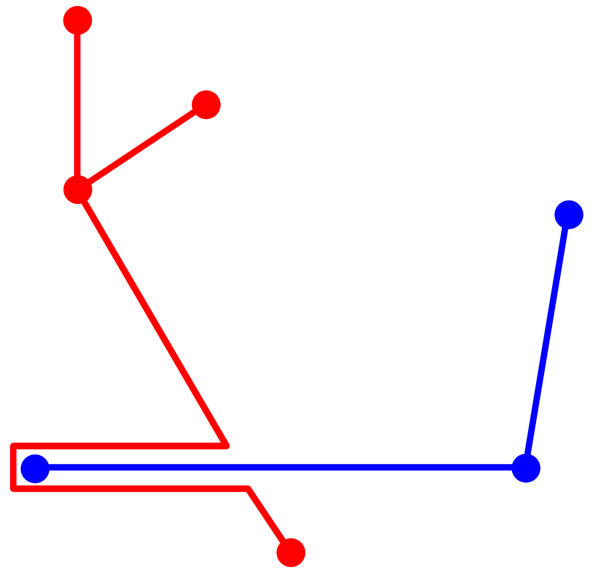
Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings and cycles

Analysis:



*C*olored (*E*uclidean) *S*panning *T*rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

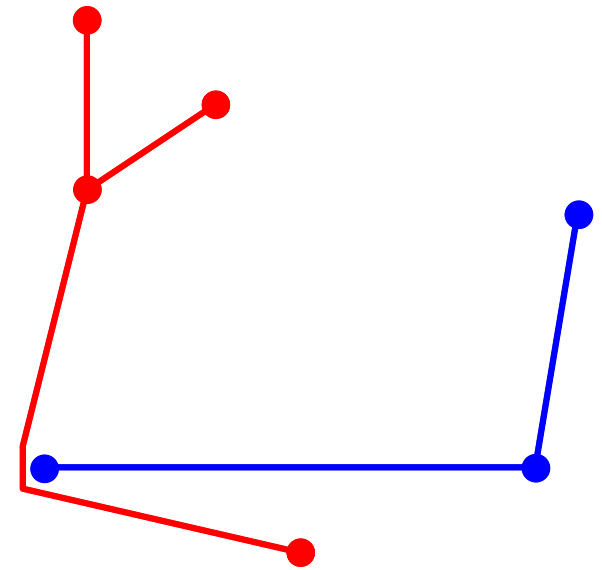
Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a “shell” around it of another color

– remove crossings and cycles, shortcut

Analysis:



Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

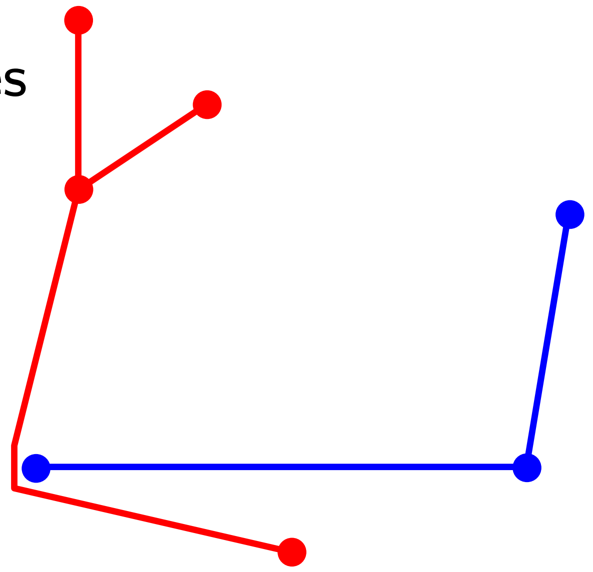
– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings and cycles , shortcut

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees



Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings and cycles , shortcut

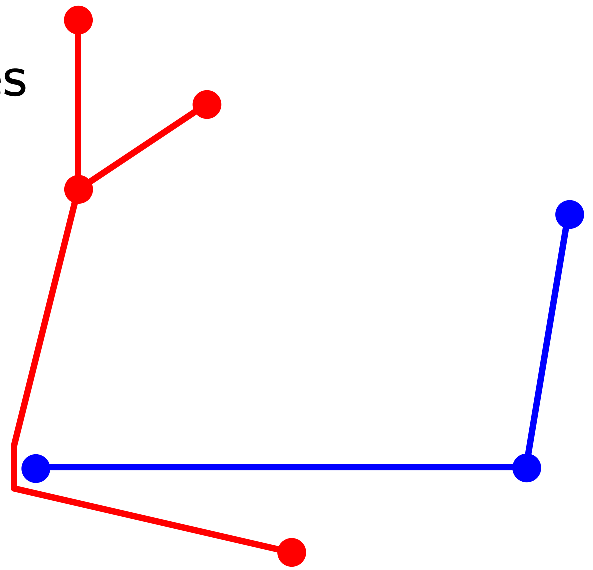
Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\mathrm{OPT}_B \geq |\text{Steiner Tree}_B|$

$\mathrm{OPT}_R \geq |\text{Steiner Tree}_R|$



Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a “shell” around it of another color

– remove crossings and cycles , shortcut

Analysis:

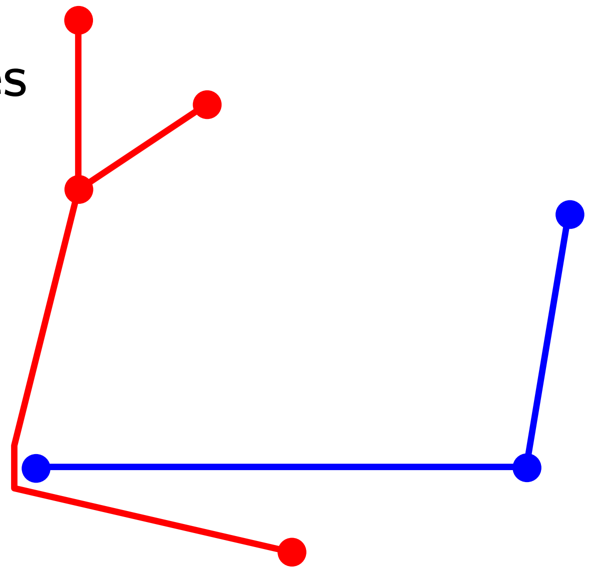
– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees



Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a “shell” around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

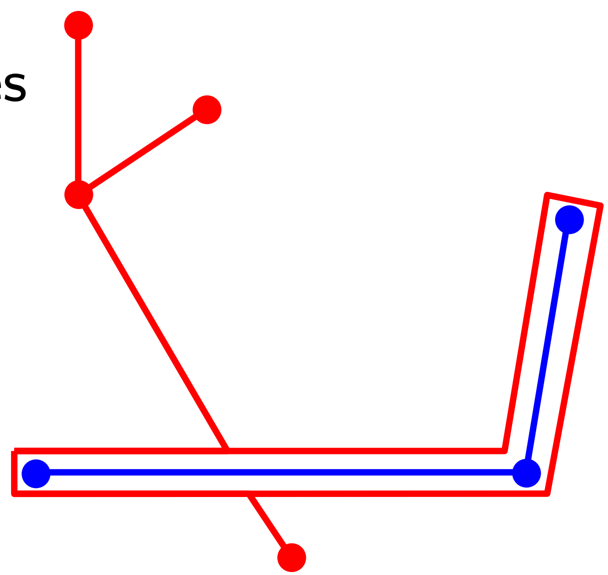
$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\text{ALG}_B = |\text{MST}_B|$

$\text{ALG}_R = |\text{MST}_R| + 2|\text{MST}_B|$



Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\mathrm{OPT}_B \geq |\text{Steiner Tree}_B|$

$\mathrm{OPT}_R \geq |\text{Steiner Tree}_R|$

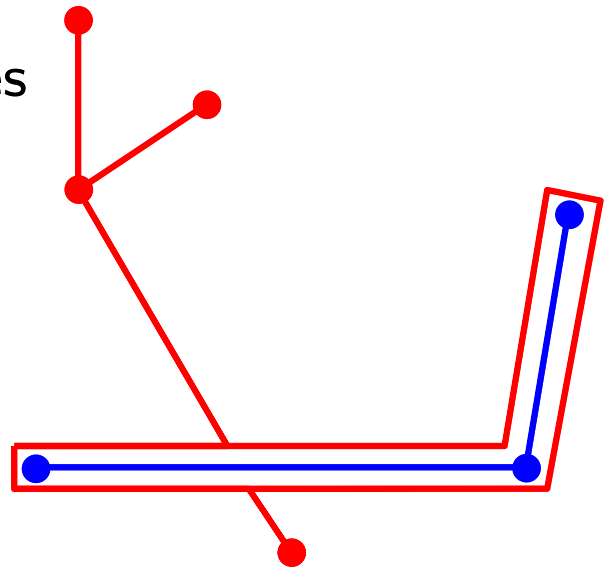
– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\mathrm{ALG}_B = |\,\mathrm{MST}_B\,|$

$\mathrm{ALG}_R = |\,\mathrm{MST}_R\,| + 2|\,\mathrm{MST}_B\,|$

$$\frac{\mathrm{ALG}}{\mathrm{OPT}} \leq \frac{|\,\mathrm{MST}_R\,| + 3|\,\mathrm{MST}_B\,|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|}$$



Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

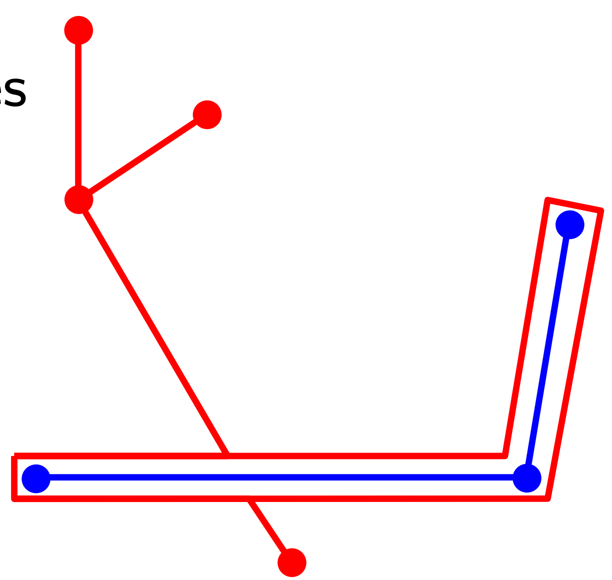
$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\text{ALG}_B = |\,\text{MST}_B\,|$

$\text{ALG}_R = |\,\text{MST}_R\,| + 2|\,\text{MST}_B\,|$



$$\frac{\text{ALG}}{\text{OPT}} \leq \frac{|\,\text{MST}_R\,| + 3|\,\text{MST}_B\,|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\,\text{MST}_R\,| + 3|\,\text{MST}_B\,|}{|\,\text{MST}_R\,| + |\,\text{MST}_B\,|}$$

Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

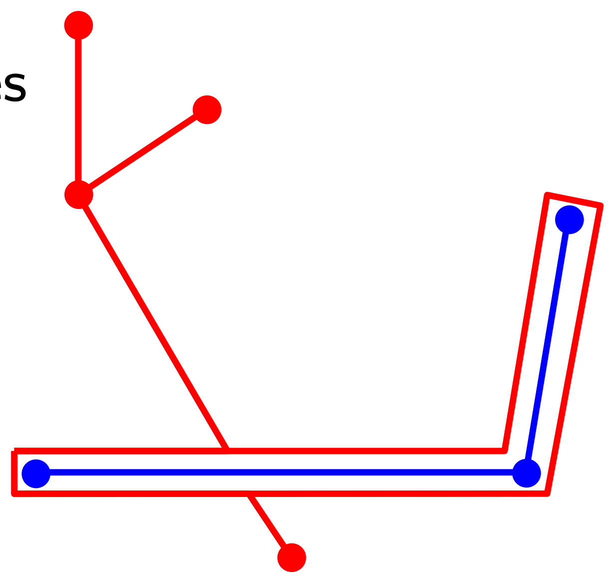
$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\text{ALG}_B = |\,\text{MST}_B\,|$

$\text{ALG}_R = |\,\text{MST}_R\,| + 2|\,\text{MST}_B\,|$



$$\frac{\text{ALG}}{\text{OPT}} \leq \frac{|\,\text{MST}_R\,| + 3|\,\text{MST}_B\,|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\,\text{MST}_R\,| + 3|\,\text{MST}_B\,|}{|\,\text{MST}_R\,| + |\,\text{MST}_B\,|} \leq \rho \cdot 2$$

Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a “shell” around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

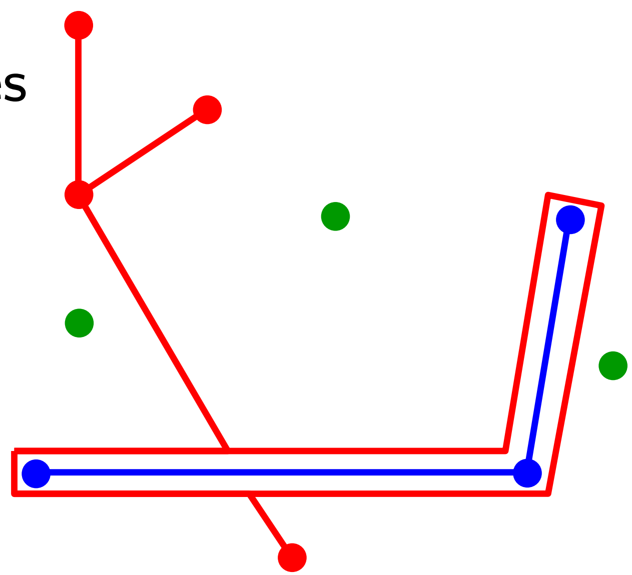
$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\text{ALG}_B = |\text{MST}_B|$

$\text{ALG}_R = |\text{MST}_R| + 2|\text{MST}_B|$



$$\frac{\text{ALG}}{\text{OPT}} \leq \frac{|\text{MST}_R| + 3|\text{MST}_B|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\text{MST}_R| + 3|\text{MST}_B|}{|\text{MST}_R| + |\text{MST}_B|} \leq \rho \cdot 2$$

Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\mathrm{OPT}_B \geq |\text{Steiner Tree}_B|$

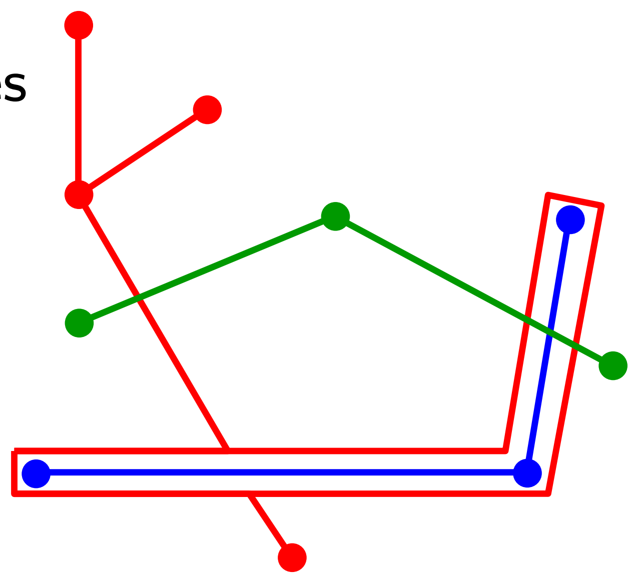
$\mathrm{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\mathrm{ALG}_B = |\,\mathrm{MST}_B\,|$

$\mathrm{ALG}_R = |\,\mathrm{MST}_R\,| + 2|\,\mathrm{MST}_B\,|$



$$\frac{\mathrm{ALG}}{\mathrm{OPT}} \leq \frac{|\,\mathrm{MST}_R\,| + 3|\,\mathrm{MST}_B\,|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\,\mathrm{MST}_R\,| + 3|\,\mathrm{MST}_B\,|}{|\,\mathrm{MST}_R\,| + |\,\mathrm{MST}_B\,|} \leq \rho \cdot 2$$

Colored (Euclidean) Spanning Trees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$):

– construct red and blue minimum spanning trees

– take the shorter one , add a "shell" around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

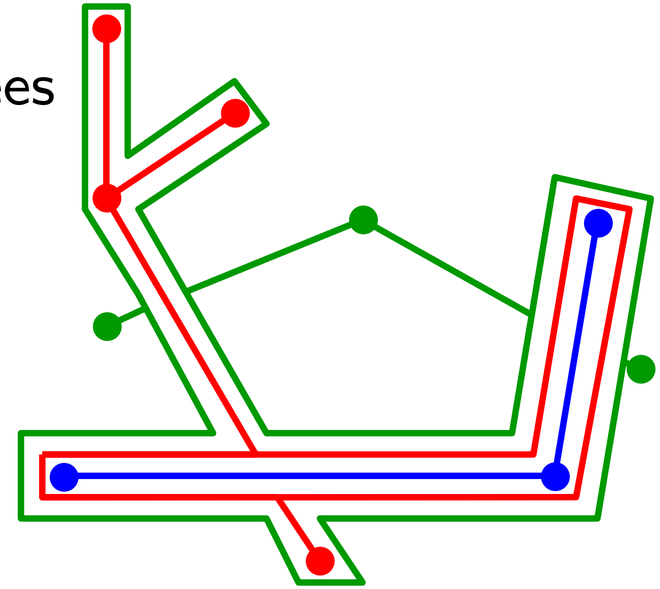
$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\text{ALG}_B = |\text{MST}_B|$

$\text{ALG}_R = |\text{MST}_R| + 2|\text{MST}_B|$



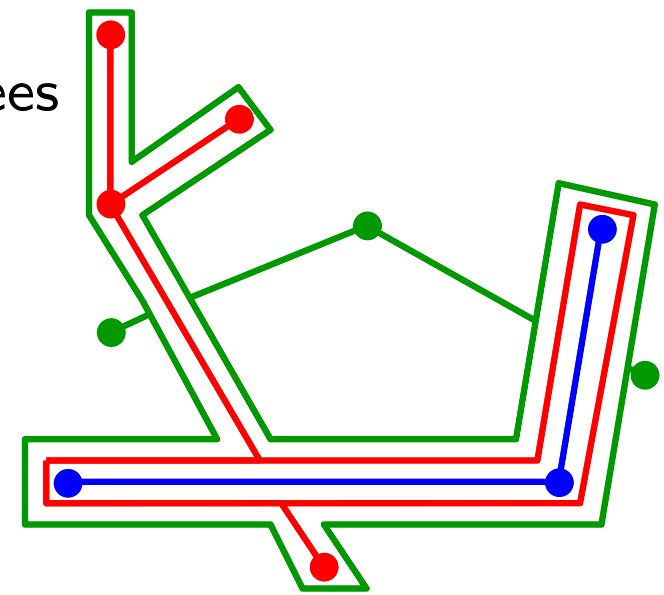
$$\frac{\text{ALG}}{\text{OPT}} \leq \frac{|\text{MST}_R| + 3|\text{MST}_B|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\text{MST}_R| + 3|\text{MST}_B|}{|\text{MST}_R| + |\text{MST}_B|} \leq \rho \cdot 2$$

Colored (*E*uclidean) *S*panning *T*rees

Theorem CST (with k colors) admits a $(k\rho)$-approximation

Proof Algorithm ($k = 2$): $(1 + \varepsilon)$-approx. Steiner
– construct red and blue ~~minimum spanning~~ trees
– take the shorter one , add a “shell” around it of another color
– remove crossings

Analysis:
– let OPT_B, OPT_R be optimal non-crossing trees
Since the trees connect points



$\mathrm{OPT}_B \geq |\text{Steiner Tree}_B|$
$\mathrm{OPT}_R \geq |\text{Steiner Tree}_R|$
– let ALG_B, ALG_R be the resulting trees
Before removing cycles/shortcutting
$\mathrm{ALG}_B = |\,\mathrm{MST}_B\,|$
$\mathrm{ALG}_R = |\,\mathrm{MST}_R\,| + 2|\,\mathrm{MST}_B\,|$

$$\frac{\mathrm{ALG}}{\mathrm{OPT}} \leq \frac{|\,\mathrm{MST}_R\,| + 3|\,\mathrm{MST}_B\,|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\,\mathrm{MST}_R\,| + 3|\,\mathrm{MST}_B\,|}{|\,\mathrm{MST}_R\,| + |\,\mathrm{MST}_B\,|} \leq \rho \cdot 2$$

Colored (Euclidean) Spanning Trees

$(k+\varepsilon)$

Theorem CST (with k colors) admits a ~~$(k\rho)$~~-approximation

Proof Algorithm ($k = 2$): $(1+\varepsilon)$-approx. Steiner

– construct red and blue ~~minimum spanning~~ trees

– take the shorter one , add a "shell" around it of another color

– remove crossings

Analysis:

– let OPT_B, OPT_R be optimal non-crossing trees

Since the trees connect points

$\text{OPT}_B \geq |\text{Steiner Tree}_B|$

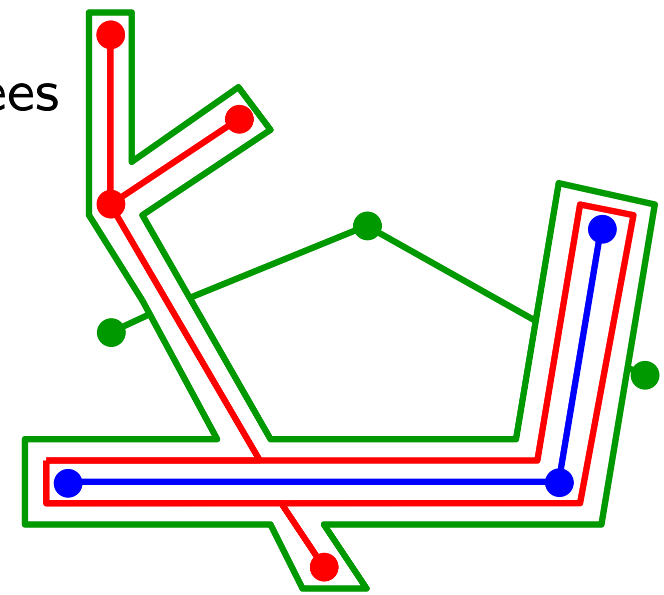
$\text{OPT}_R \geq |\text{Steiner Tree}_R|$

– let ALG_B, ALG_R be the resulting trees

Before removing cycles/shortcutting

$\text{ALG}_B = |\text{MST}_B|$

$\text{ALG}_R = |\text{MST}_R| + 2|\text{MST}_B|$



$$\frac{\text{ALG}}{\text{OPT}} \leq \frac{|\text{MST}_R| + 3|\text{MST}_B|}{|\text{Steiner Tree}_R| + |\text{Steiner Tree}_B|} \leq \rho \frac{|\text{MST}_R| + 3|\text{MST}_B|}{|\text{MST}_R| + |\text{MST}_B|} \leq \rho \cdot 2$$

Conclusions

- new visualization method MapSets

  demo is at `http://gmap.cs.arizona.edu`

  source code is at `GitHub` (C++, javascript)

Conclusions

- new visualization method MapSets
  demo is at `http://gmap.cs.arizona.edu`
  source code is at `GitHub` (C++, javascript)
- new geometric problem CST
  $(k + \varepsilon)$-approximation

Conclusions

- new visualization method MapSets
  demo is at `http://gmap.cs.arizona.edu`
  source code is at `GitHub` (C++, javascript)
- new geometric problem CST
  $(k + \varepsilon)$-approximation

What is next?

- quantitative/qualitative evaluation of the different methods
- visualizing *graphs* rather than sets
- improve approximation factors for CST

Conclusions

- new visualization method MapSets
  demo is at `http://gmap.cs.arizona.edu`
  source code is at `GitHub` (C++, javascript)
- new geometric problem CST
  $(k + \varepsilon)$-approximation

What is next?

- quantitative/qualitative evaluation of the different methods
- visualizing *graphs* rather than sets
- improve approximation factors for CST

Thank you!