OI-OPPA. European Social Fund
Prague & EU: We invest in your future.

k-th order Voronoi diagrams

Petr Egert

A4M39VG
winter 2012/2013

Outline

- Introduction
- Relation to other VDs
- Direct GVP construction
- Iterative algorithm description
- Questions

Introduction

k-th order Voronoi diagram

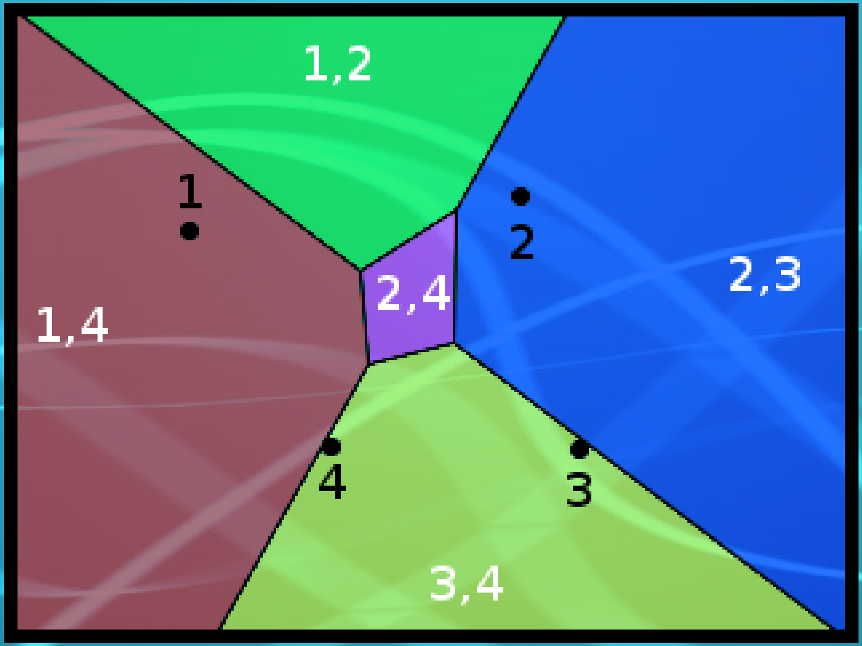
- Also called Higher Order Voronoi Diagram (HOVD)
- Notation $Vor_k(S)$
- Union of GVPs
- Returns *k* nearest neighbours by finding the appropriate GVP
- Extendible to higher dimensions
  - 2D case used here

Generalized Voronoi Polygon

- GVP
- Notation *V(T)*
  - Each site in *T* closer to point *p* than any site not in *T*
  - ie. *V({1,2})* = area, where sites 1 and 2 are closer than any other sites
- Always convex
- Can be empty

Example

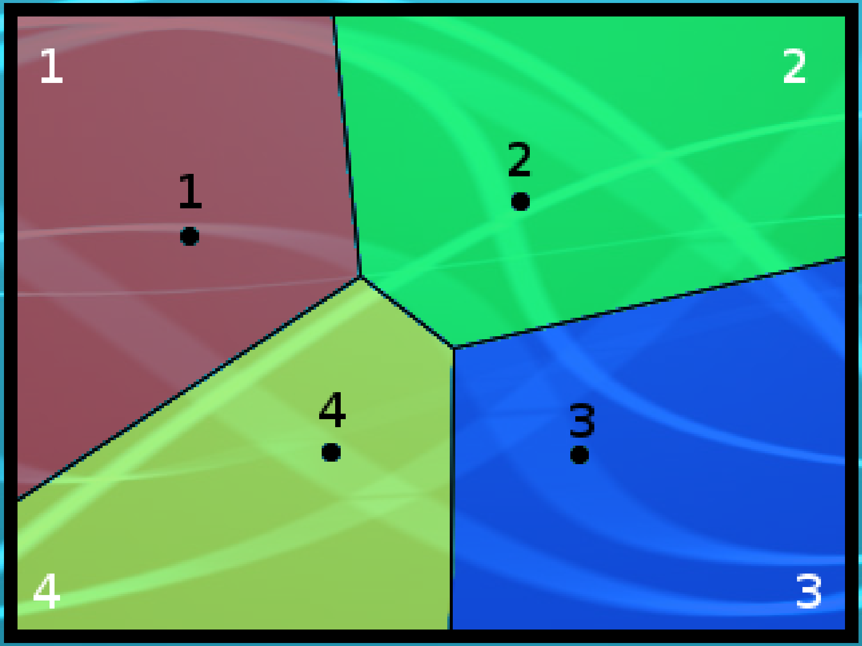
- k = 2



Relation to other VDs

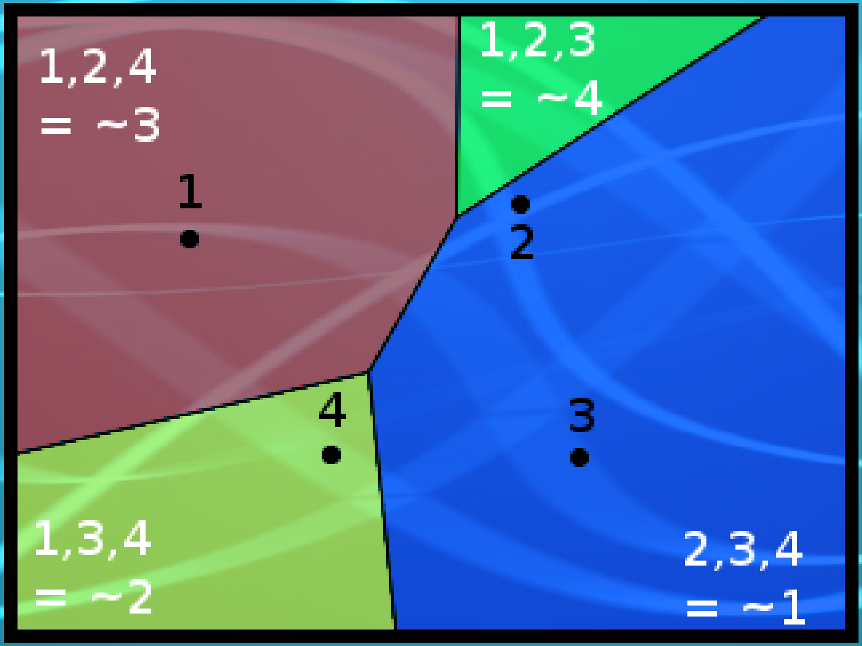
Ordinary VD

- k = 1
  - $Vor_1(S)$ = Ordinary Voronoi diagram



Farthest point VD

- k = N - 1
  - $Vor_{n-1}(S)$ = Farthest point Voronoi diagram

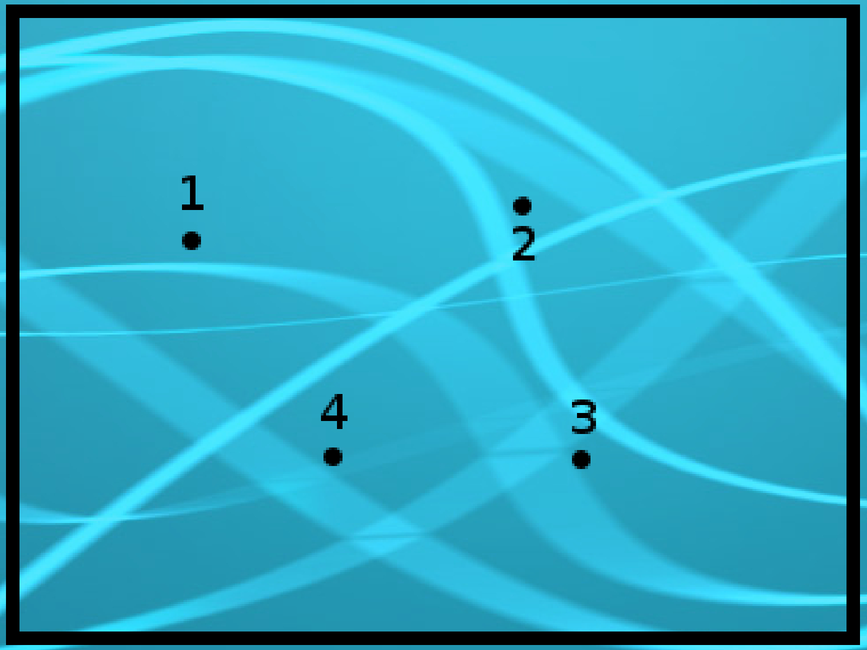


Direct GVP construction

Direct GVP construction

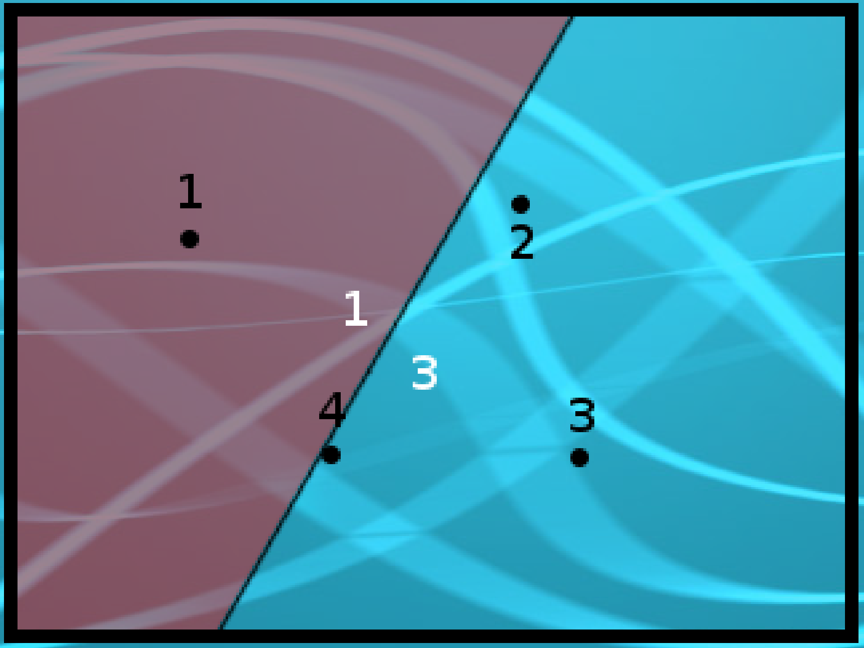
- *V(T)* = intersection of all halfplanes, except for those created by bisections of *T*
  1. Compute bisections of each site in *T* with all other sites, except for those in *T*
  2. Intersect all halfplanes containing the given site
     - The resulting GVP can be empty

Example - $V(\{1,2,4\})$



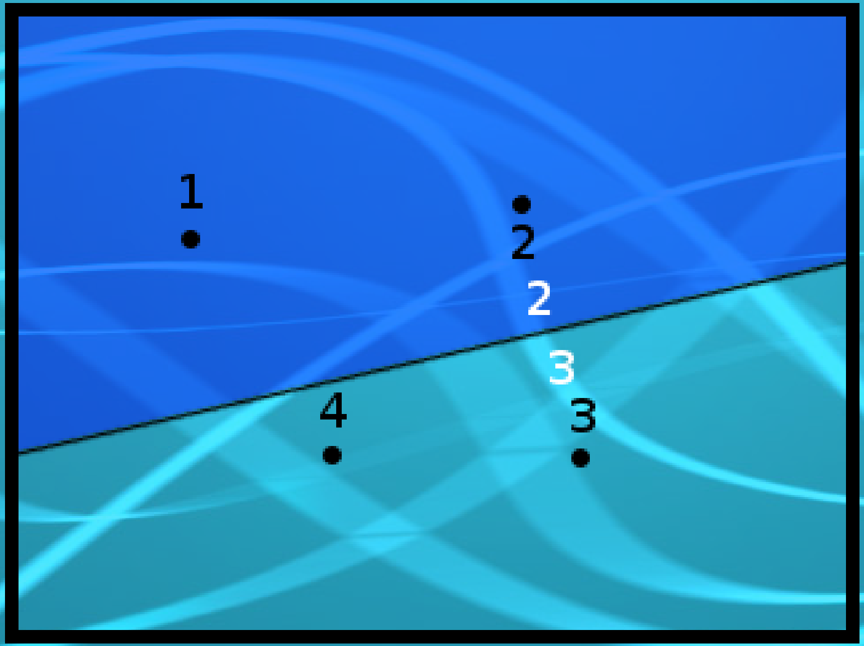
Example - *V({1,2,4})*

- Find bisections between 1 and all others
  - Ignore those within *T*, ie. *H(1,2)* and *H(1,4)*



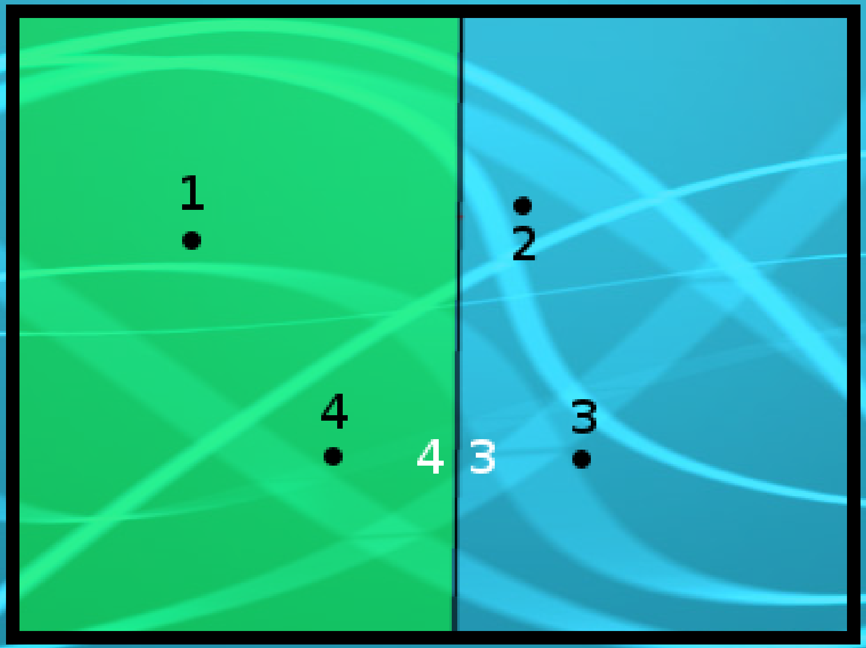
Example - *V({1,2,4})*

- Find bisections between 2 and all others
  - Ignore those within *T*, ie. *H(2,1)* and *H(2,4)*



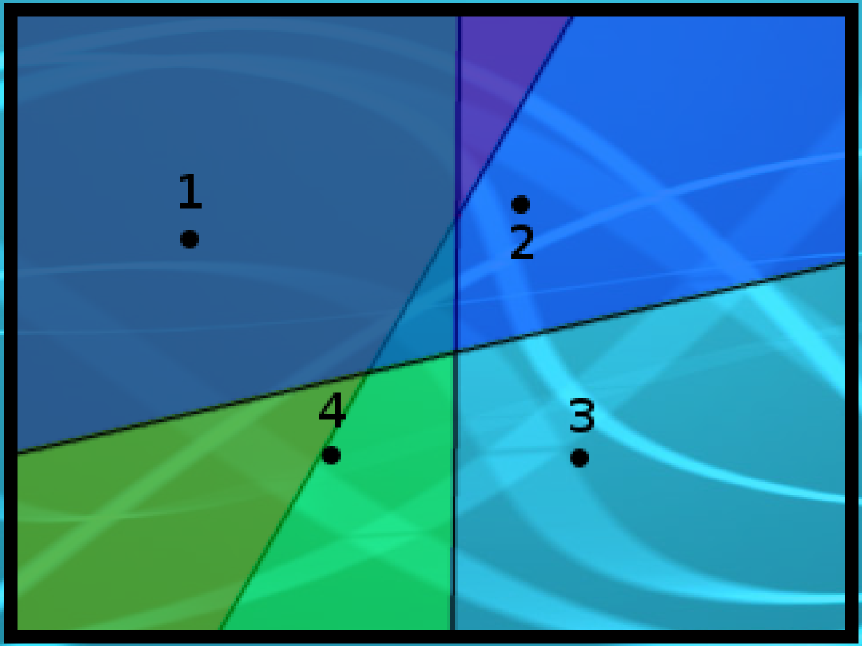
Example - *V({1,2,4})*

- Find bisections between 4 and all others
  - Ignore those within *T*, ie. *H(4,1)* and *H(4,2)*



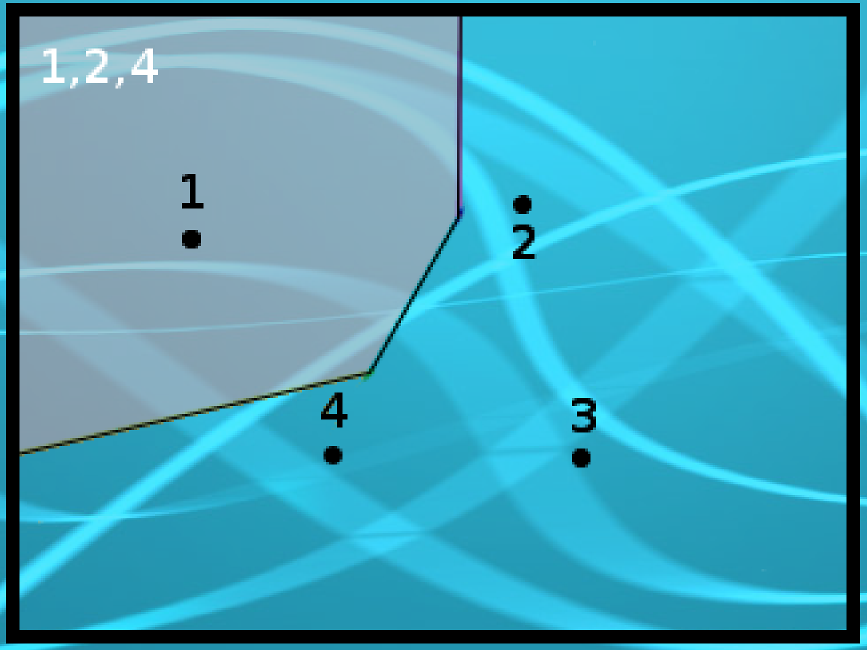
Example - *V({1,2,4})*

- Intersect all the halfplanes



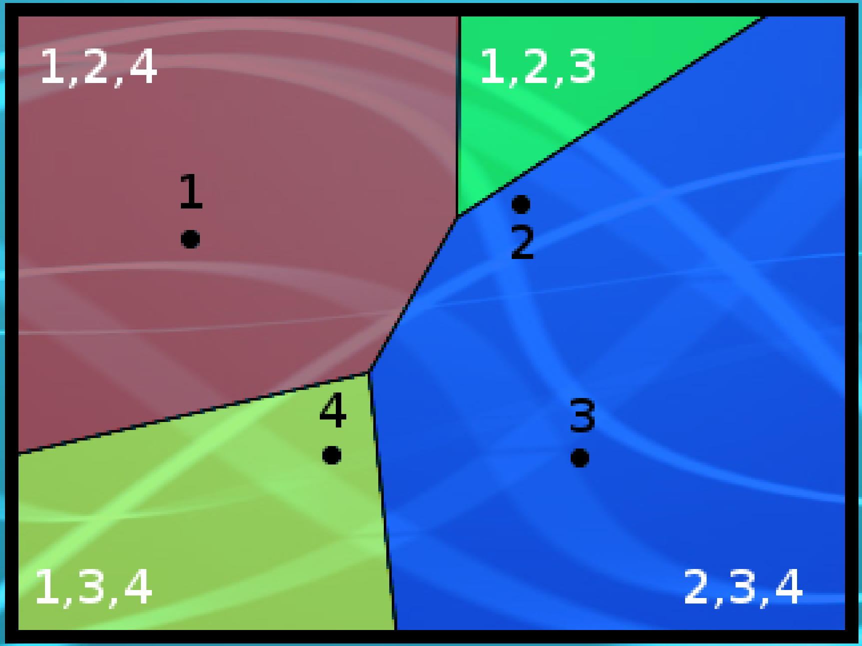
Example - *V({1,2,4})*

- The resulting GVP is found



Example - *V({1,2,4})*

- Repeat for each combination get the whole diagram



Pros and Cons

- Pros
  - Can construct a single GVP
  - Can construct order-k diagram directly
  - Higher order means less processing
- Cons
  - O($\binom{N}{k}$) time complexity
  - Processing power wasted on empty GVPs

Iterative algorithm

Iterative algorithm

- Computes $Vor_k(S)$ from $Vor_{k-1}(S)$
- Idea
  - In $Vor_{k-1}(S)$ we already know *k-1* closest sites
  - To obtain *k* closest sites, it's enough to find the missing one

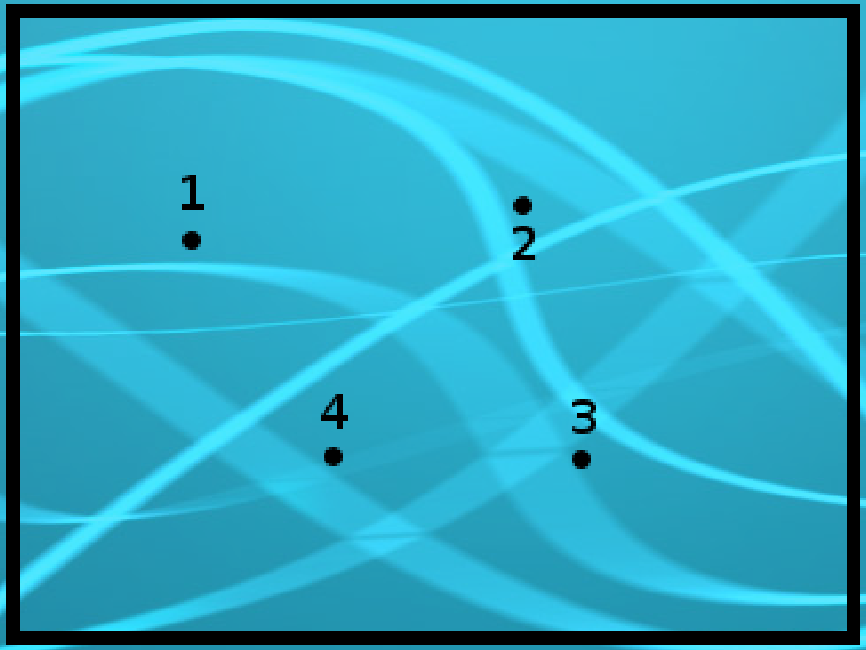
The algorithm

- Start with a known $Vor_{k-1}(S)$
  - ie. ordinary $Vor_1(S)$ in the beginning
- Repartition each GVP of $Vor_{k-1}(S)$ using the next closest site in range
- Collapse neighbouring cells having the same closest sites
- $Vor_k(S)$ is obtained

GVP repartitioning

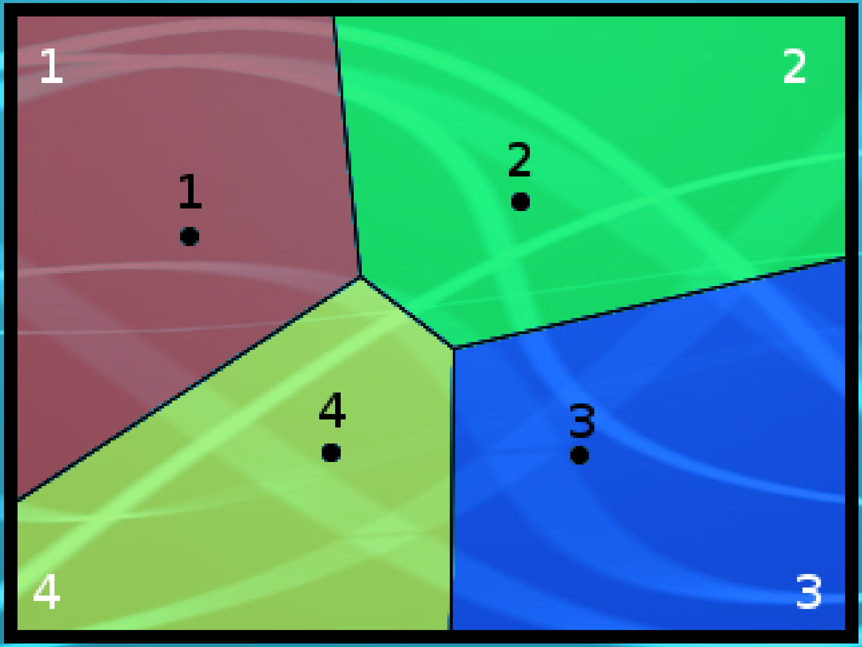
- Idea
  - Intersect *V(T)* with $Vor_1(S-T)$
- Explanation
  - Ordinary VD created from *(S-T)* contains, for any location, the closest site not already in *T*
  - Each given point *p* located inside *V(T)* is known to be closest to *T*
    - This holds even if *V(T)* is subdivided
  - Subdividing *V(T)* by $Vor_1(S-T)$ produces regions closest to both *T* and the next closest site

Example



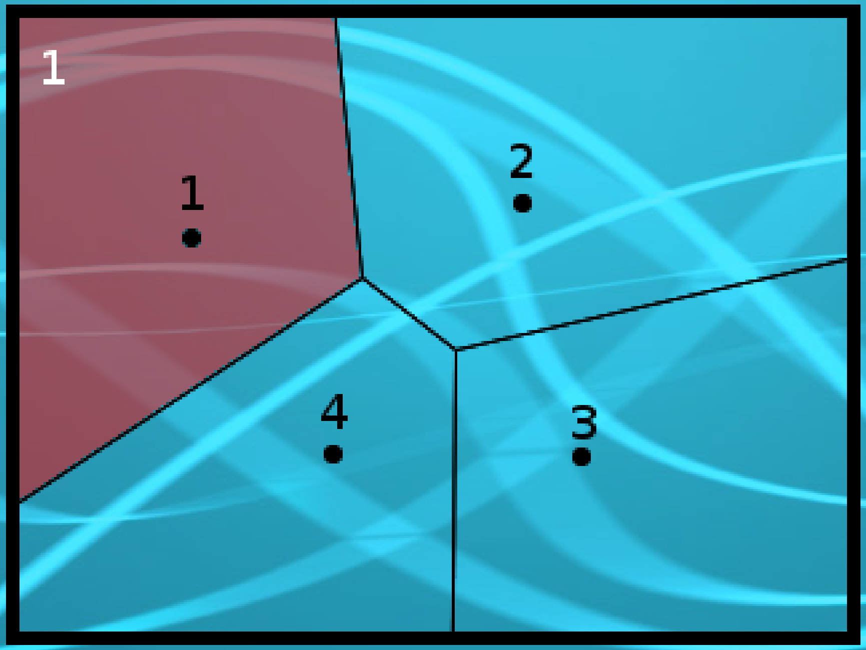
Example

- Start with $Vor_1(S)$



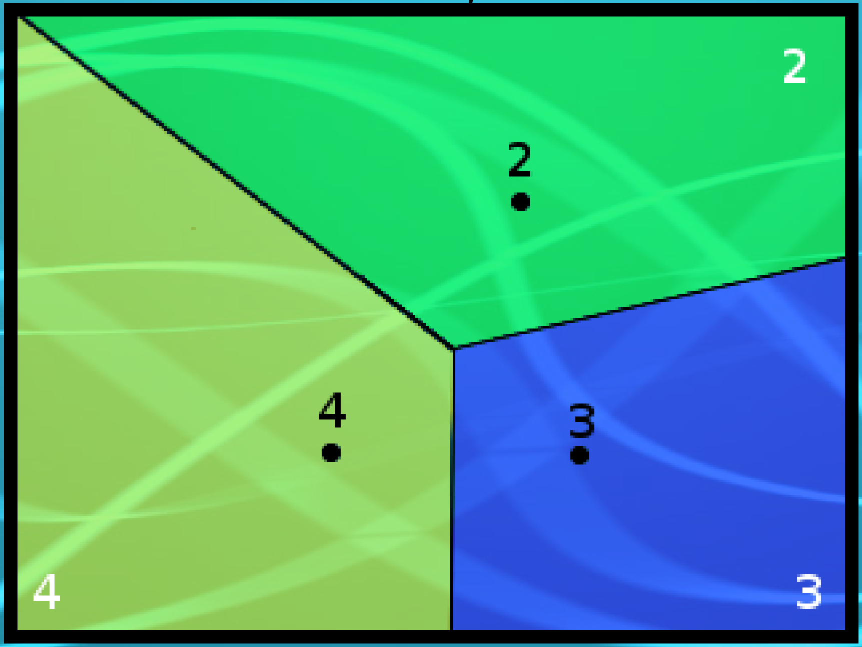
Example

- Repartition each GVP
  - Starting with *V({1})*



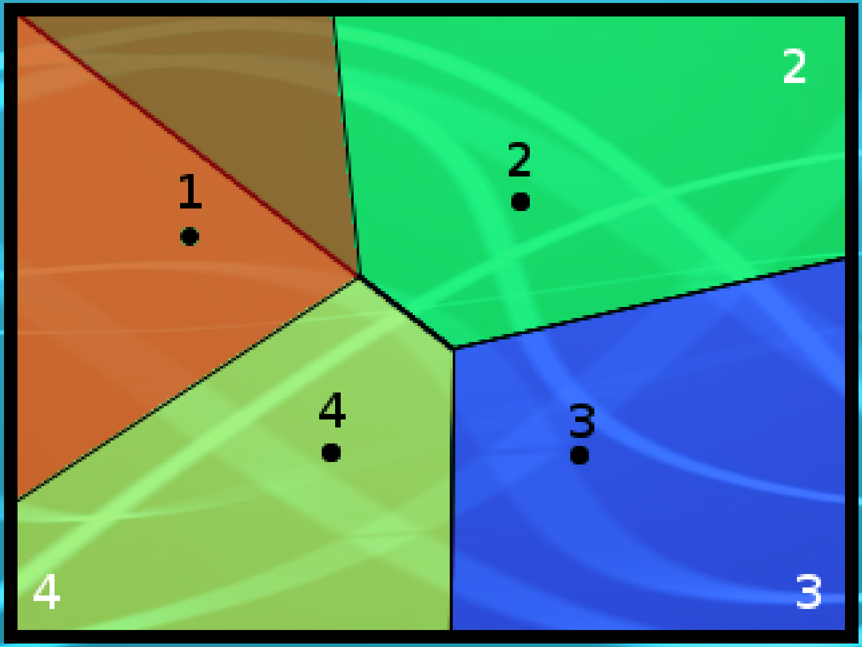
Example

- Compute $Vor_1(S\text{-}T)$
  - $T = \{1\}$, computing $Vor_1(\{2,3,4\})$



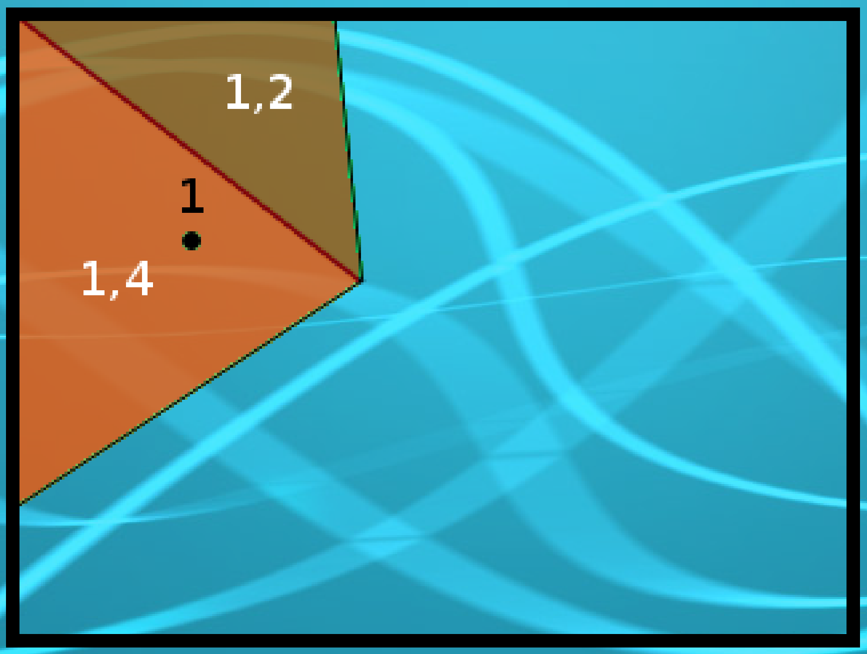
Example

- Intersect *V(T)* with $Vor_1(S\text{-}T)$
  - $T = \{1\}$



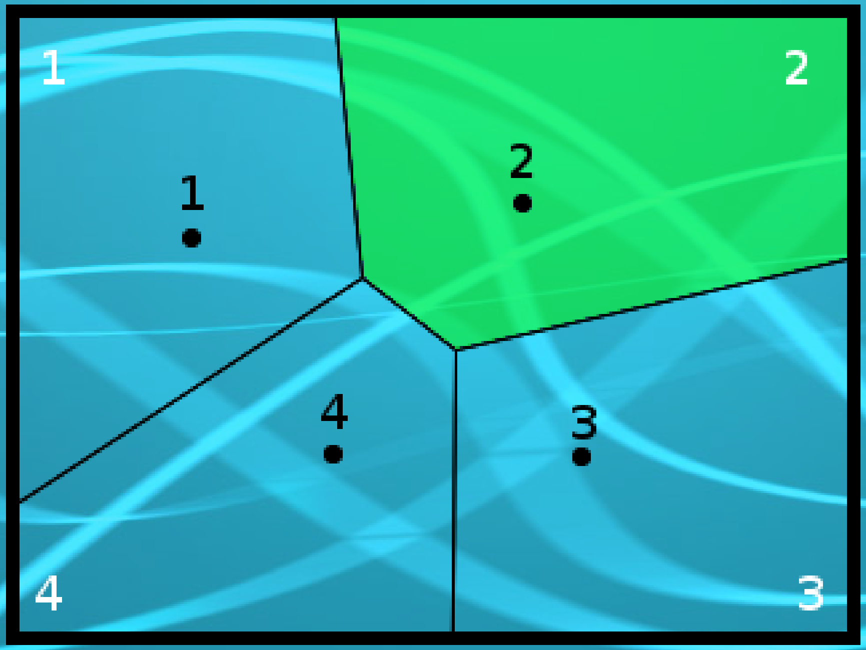
Example

- New subdivision for *V(T)* is obtained



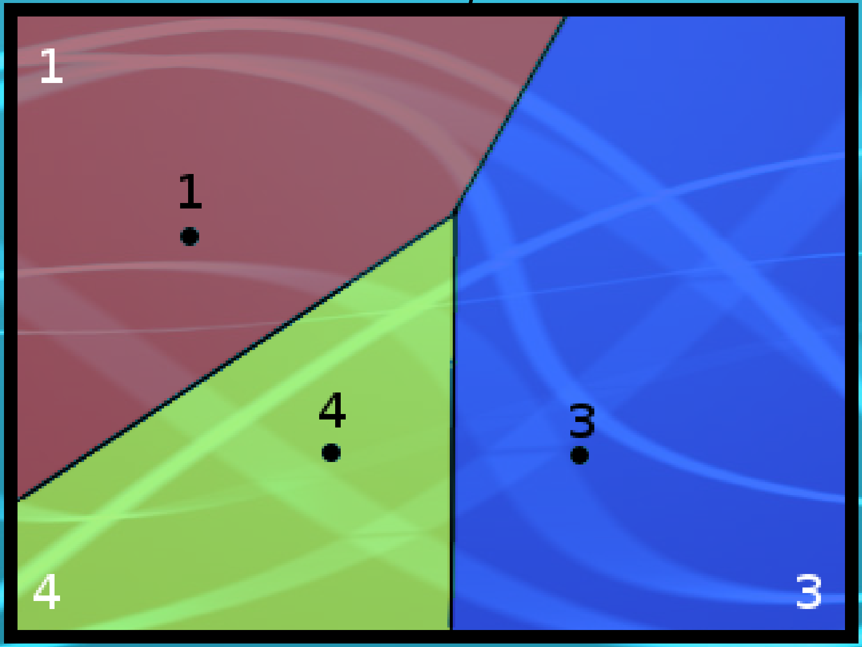
Example

- Continue with *V(T)*
  - *T={2}*



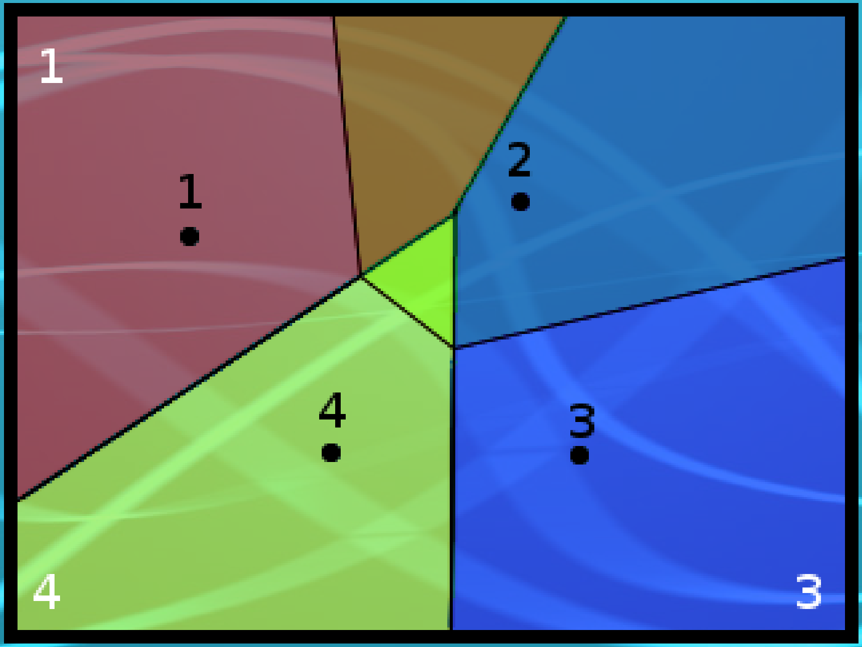
Example

- Compute $Vor_1(S\text{-}T)$
  - T = {2}, computing $Vor_1(\{1,3,4\})$



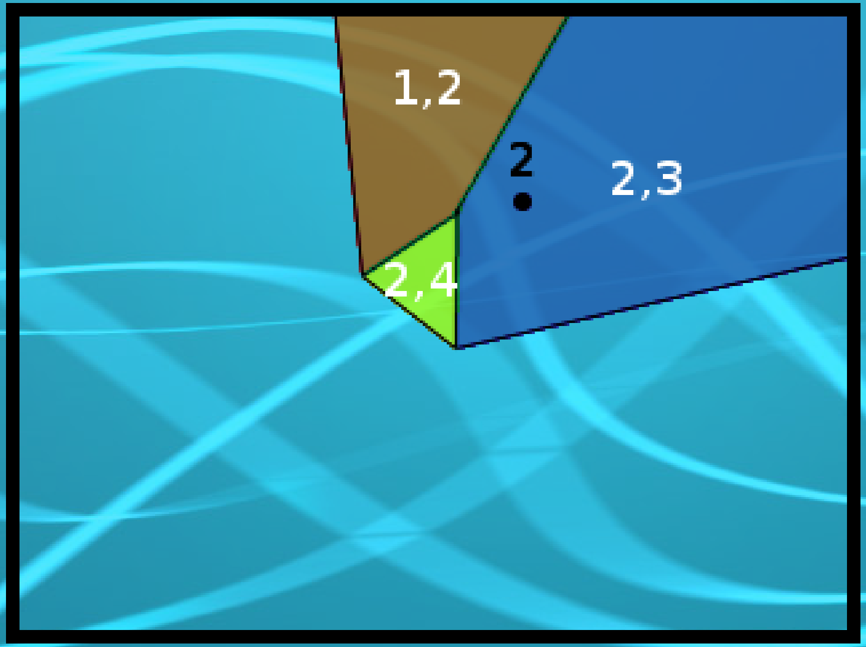
Example

- Intersect *V(T)* with $Vor_1(S\text{-}T)$
  - – $T = \{2\}$



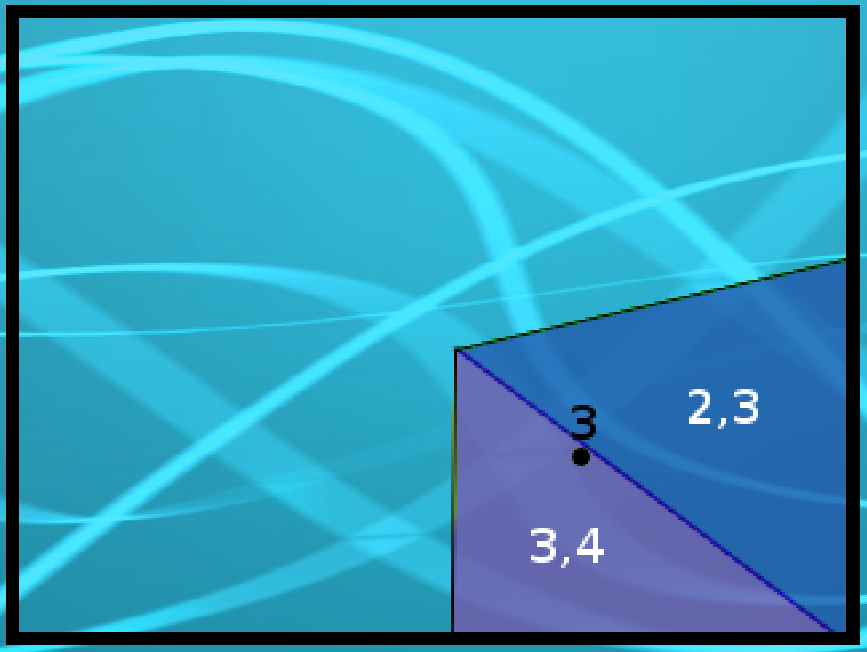
Example

- New subdivision for *V(T)* is obtained
  - *T = {2}*



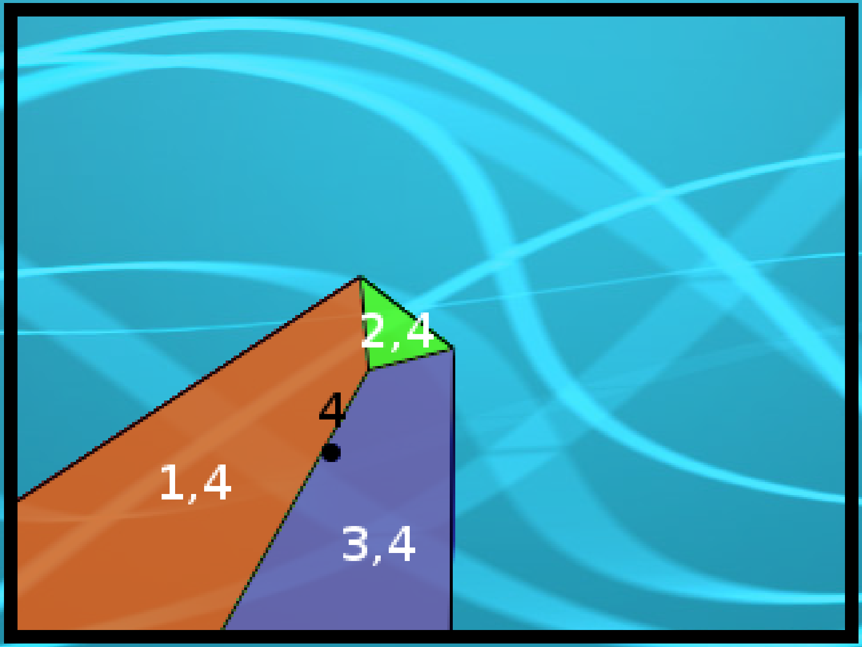
Example

- New subdivision for *V(T)* is obtained
  - *T = {3}*



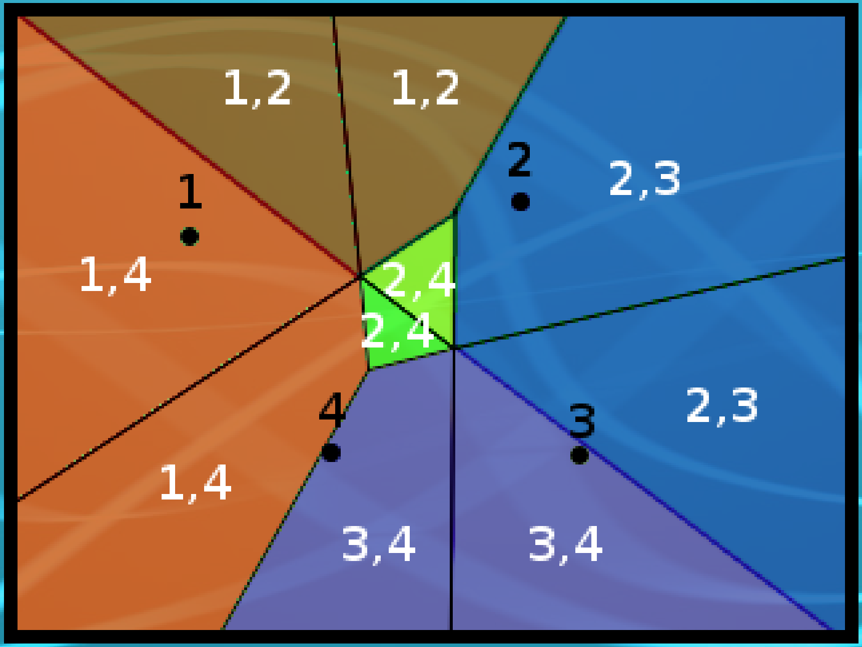
Example

- New subdivision for *V(T)* is obtained
  - *T = {4}*



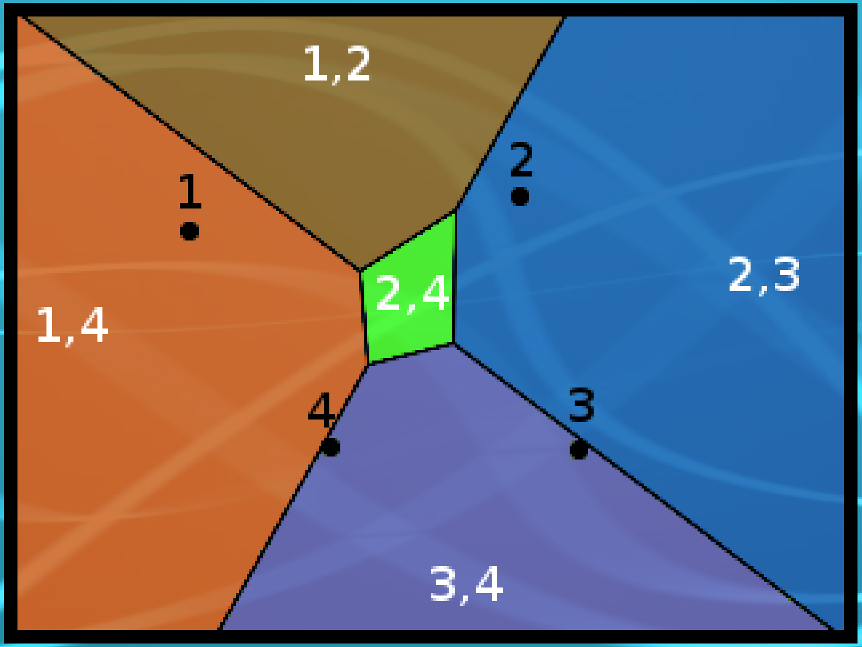
Example

- Collapse neighbouring cells with same *T*



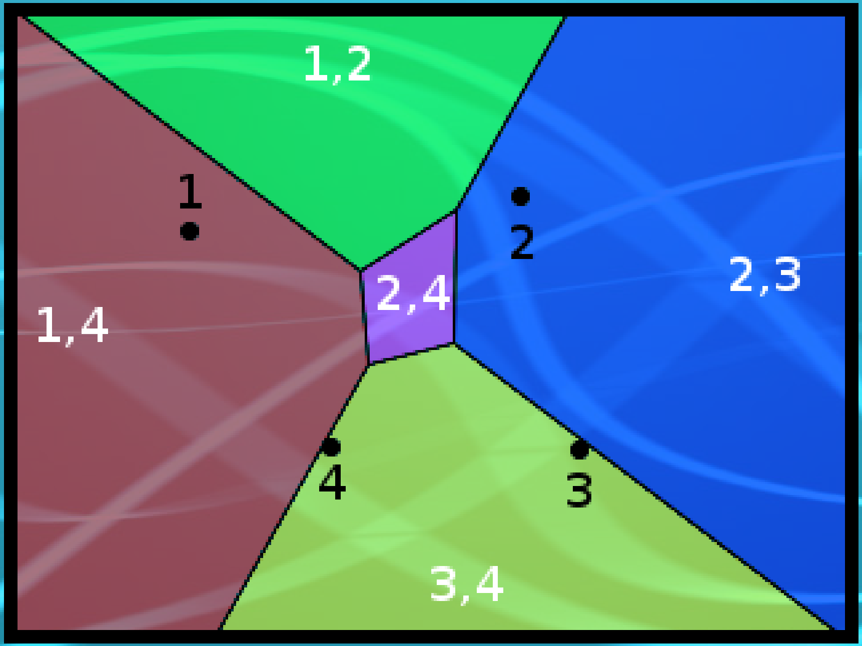
Example

- Collapse neighbouring cells with same *T*



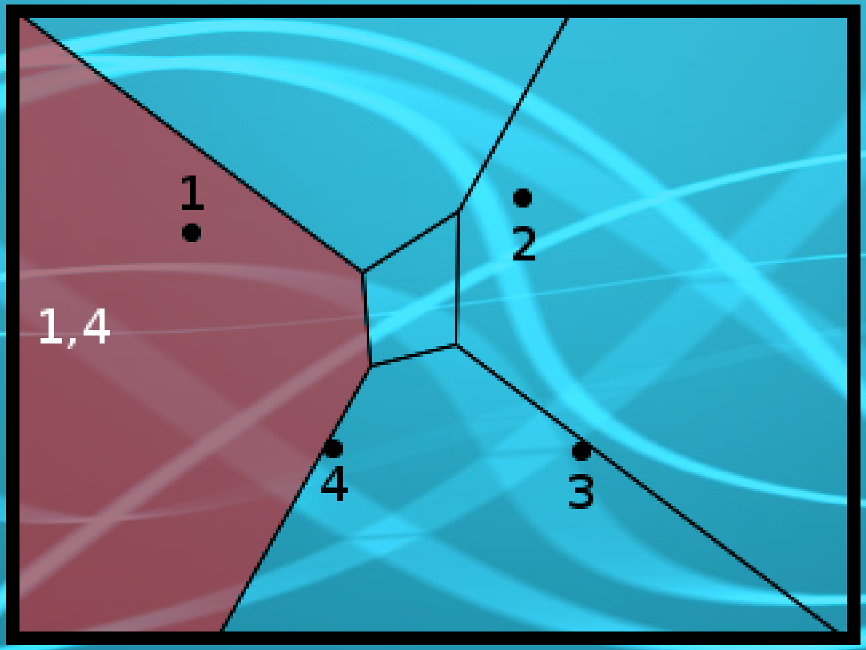
Example

- $Vor_2(S)$ is obtained



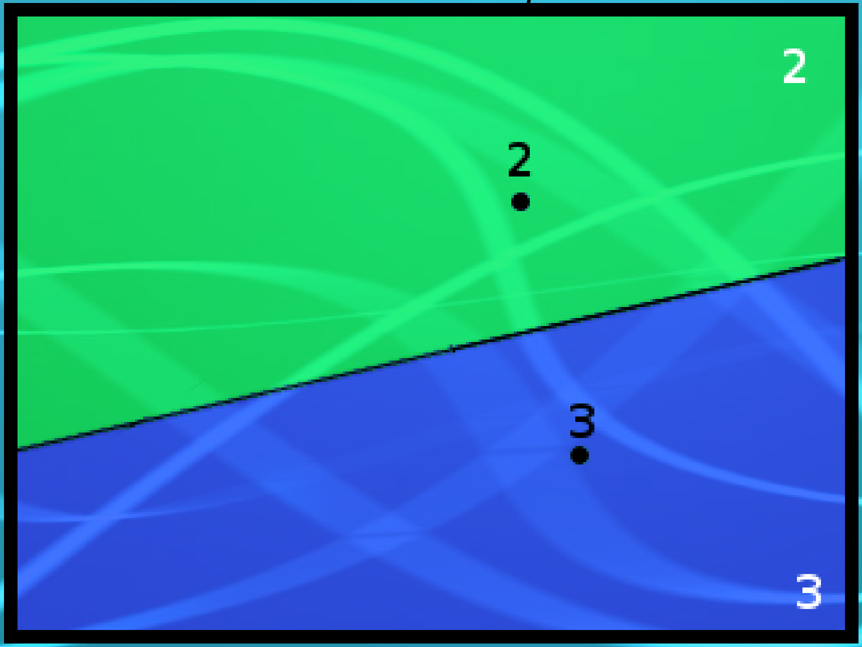
Example

- Repartition each GVP
  - Starting with *V({1,4})*



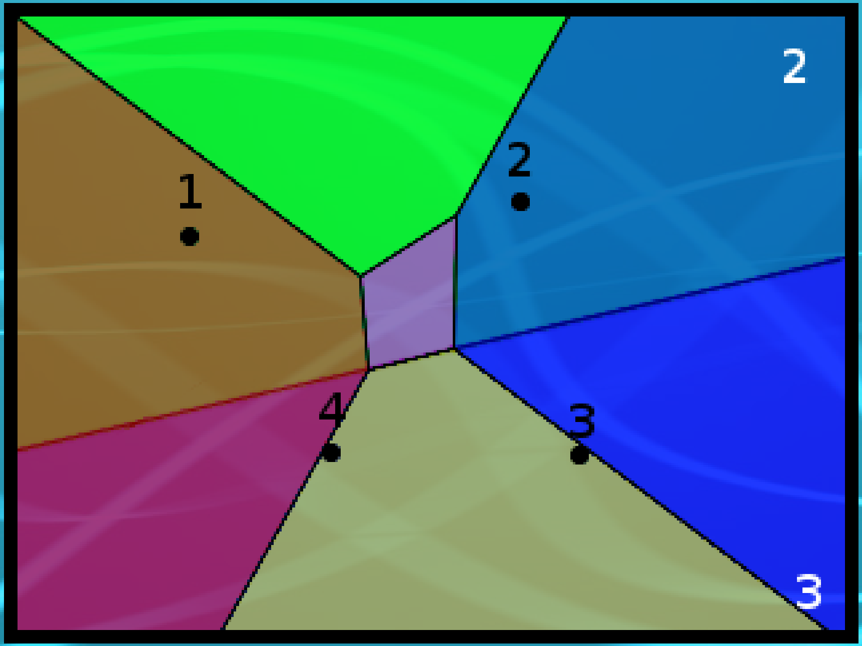
Example

- Compute $Vor_1(S\text{-}T)$
  - $T = \{1,4\}$, computing $Vor_1(\{2,3\})$



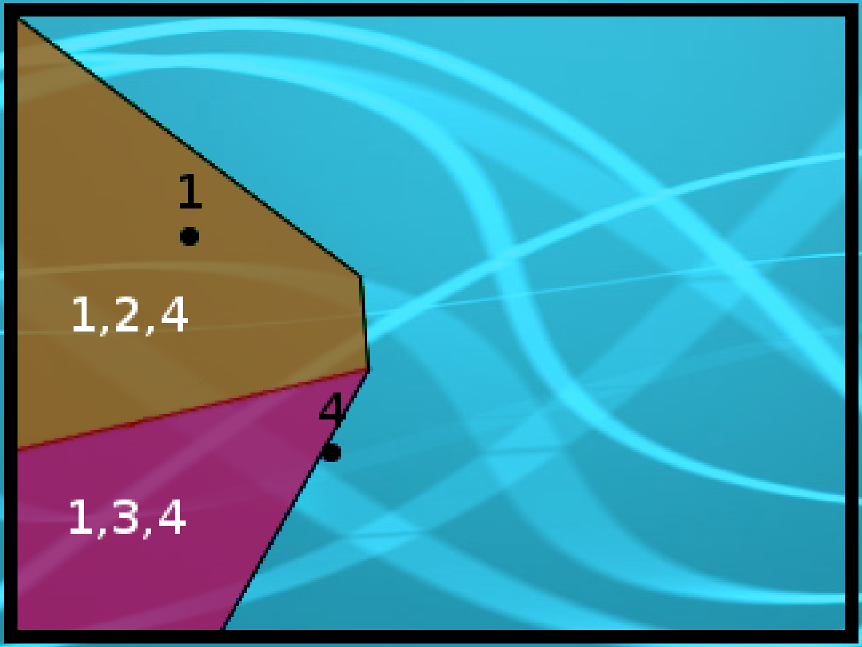
Example

- Intersect $V(T)$ with $Vor_1(S\text{-}T)$
  - $T = \{1,4\}$



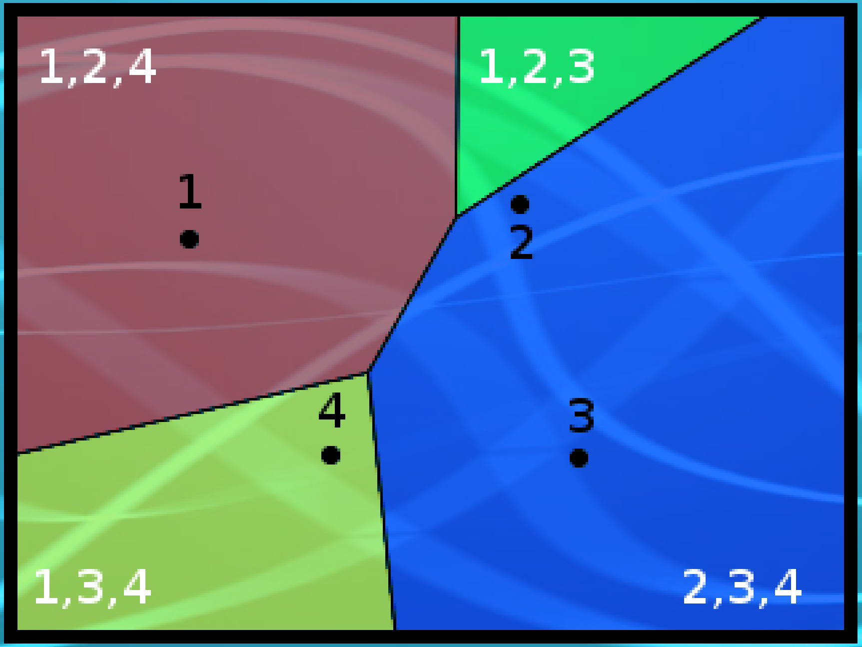
Example

- New subdivision for *V(T)* is obtained



Example

- Repeat previous steps to obtain $Vor_3(S)$



Complexity

- Space
  - $O(k(n-k))$
- Computing $Vor_k(S)$ from $Vor_{k-1}(S)$
  - $O(k(n-k))$
    - Each of $k(n-k)$ GVPs in $Vor_{k-1}(S)$ needs to be reevaluated
- Computing $Vor_k(S)$ from scratch
  - $O(n \log n + k(k(n-k)))$
    - $n \log n$ to build the first VD, then k iterations taking $k(n-k)$ time each

Thanks for your time!

Questions?

References and image sources

- Preperata F.P.- M.I.Shamos: Computational Geometry An Introduction. Berlin, Springer-Verlag,1985.
- http://www.iitg.ac.in/rinkulu/cg/slides/vor-higherorder.pdf
  - R. Inkulu: Computational geometry lecture slides
- http://cw.felk.cvut.cz/lib/exe/fetch.php/misc/projects/oppa_oi_english/courses/ae4m39vg/lectures/07-voronoi-ii.pdf
  - P. Felkel: Computational geometry lecture slides
- http://www.pollak.org/en/otherstuff/voronoi/
  - Demonstration applet by Andreas Pollak

OI-OPPA. European Social Fund
Prague & EU: We invest in your future.