MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS


Lars Arge
Aarhus University


Sarfraz Raza
Aarhus University

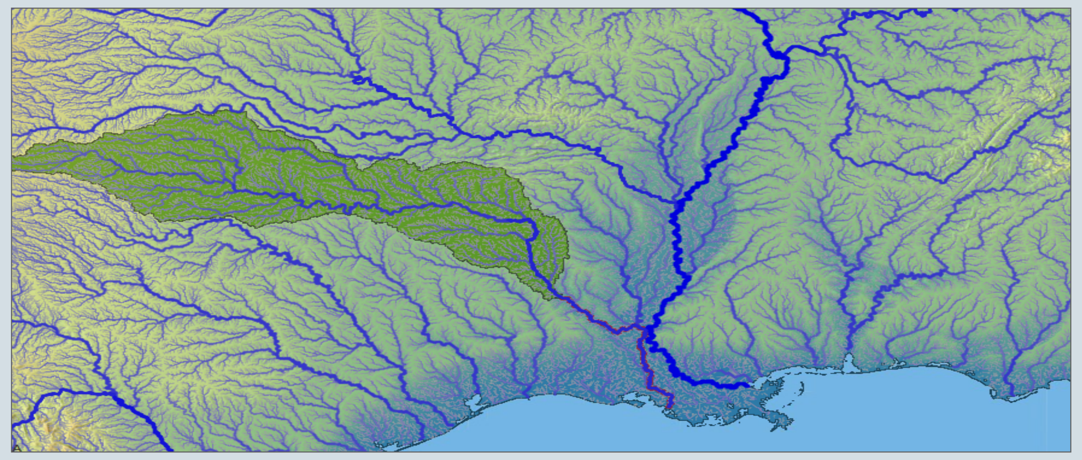

Morten Revsbæk
Aarhus University

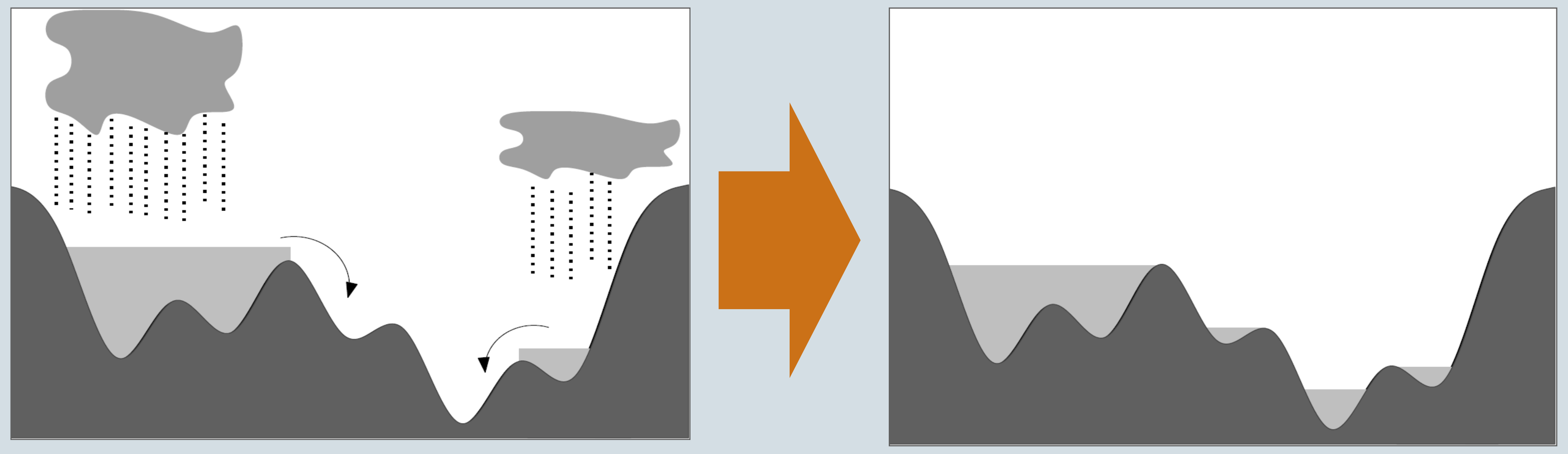
Danmarks Grundforskningsfond
Danish National Research Foundation

Water Flow and Terrain Flood Risk with Non-uniform Rain Events

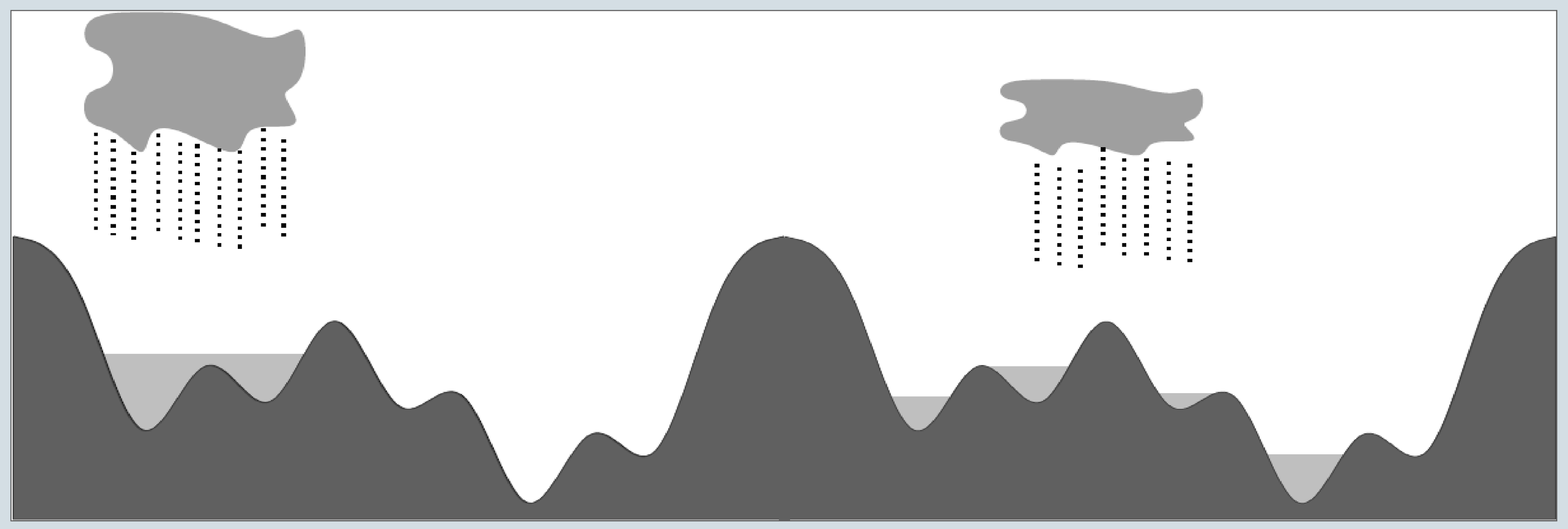
Problem

- How does the water flow and depressions fill during non-uniform rain over a terrain?





- Can we efficiently compute how water flow and depressions fill during a local rain event (without processing the entire terrain)?



- Can we incorporate non-terrain features such as soil characteristic, sewer system and etc. in uniform or non-uniform rain event computations?

Previous Results

- Computing the **spill time** of all the terrain points was first addressed by Lie et al. [1]. The solution was in the RAM–model.
- Arge et al. [2] gave the solution for computing the spill times of all the points over the terrain I/O-efficiently.

Deficiency in both results

- It is raining uniformly all over the terrain.
- Updating the local rain event cannot be incorporated.

Assumption and Preliminaries

- Massive Dataset (**I/O model**).
- Two neighboring depressions have exactly one **saddle point**, where they meet, and no other depression meet them on the same height.
- We already have the **merge tree** of the terrain defined below:

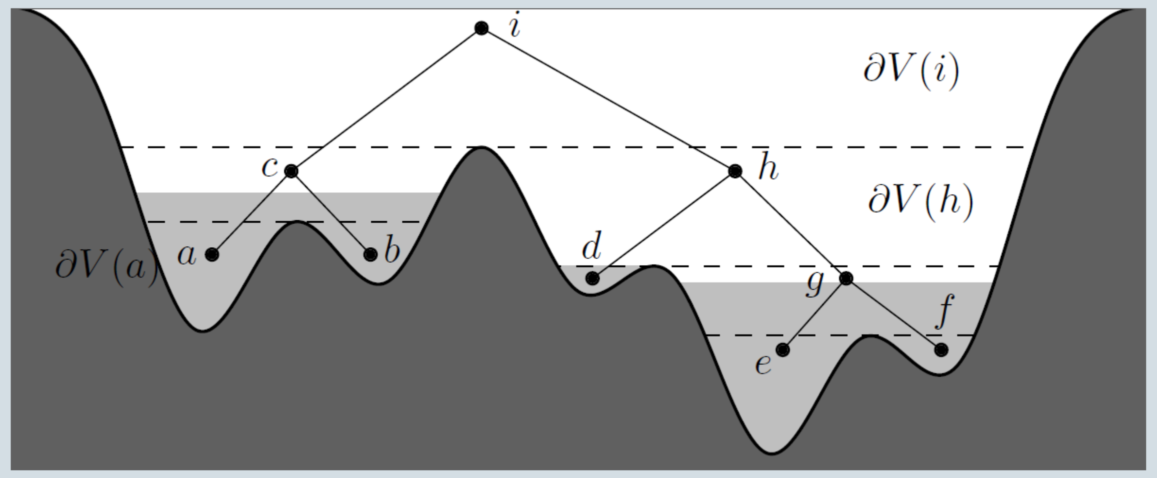
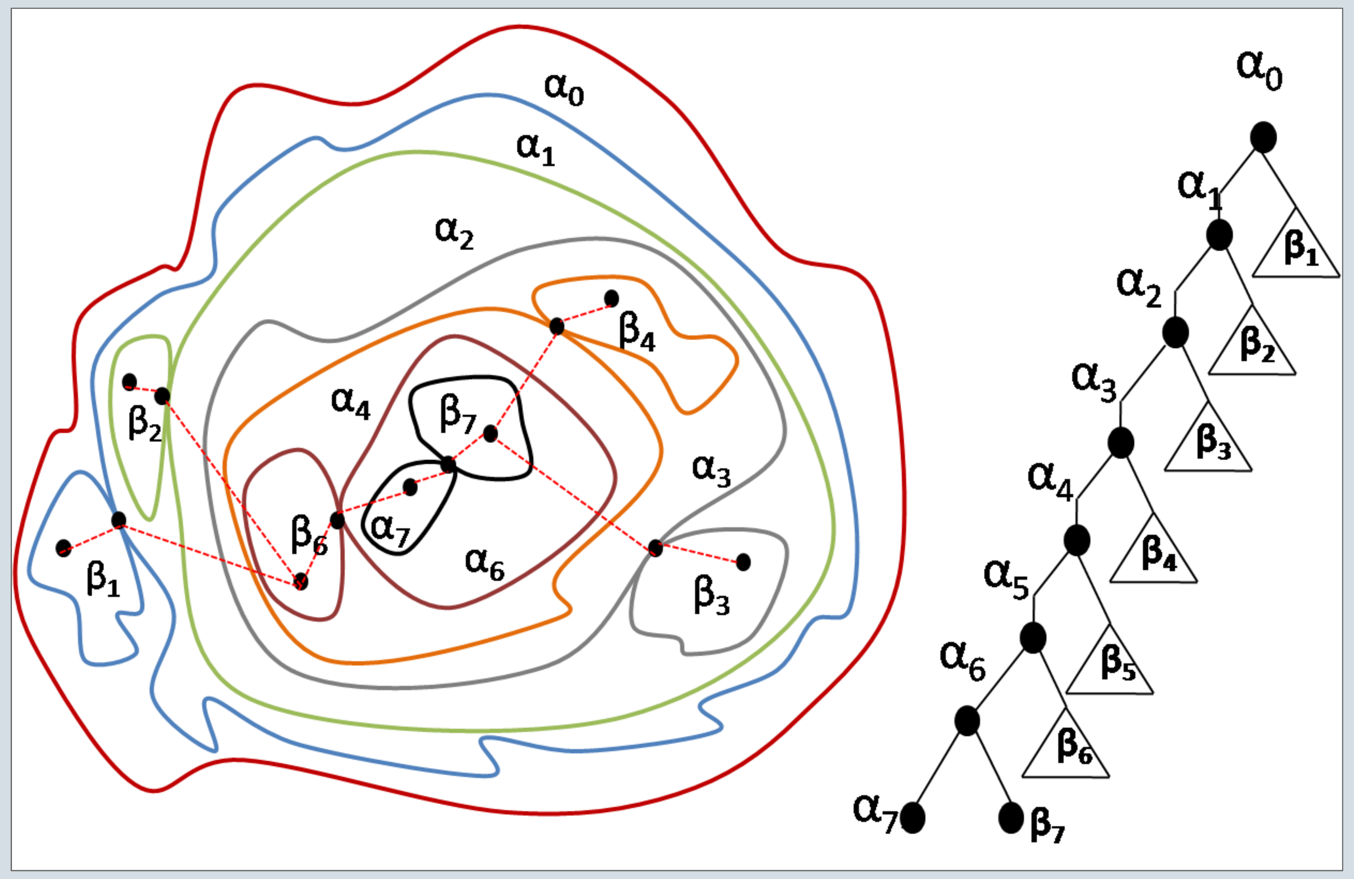


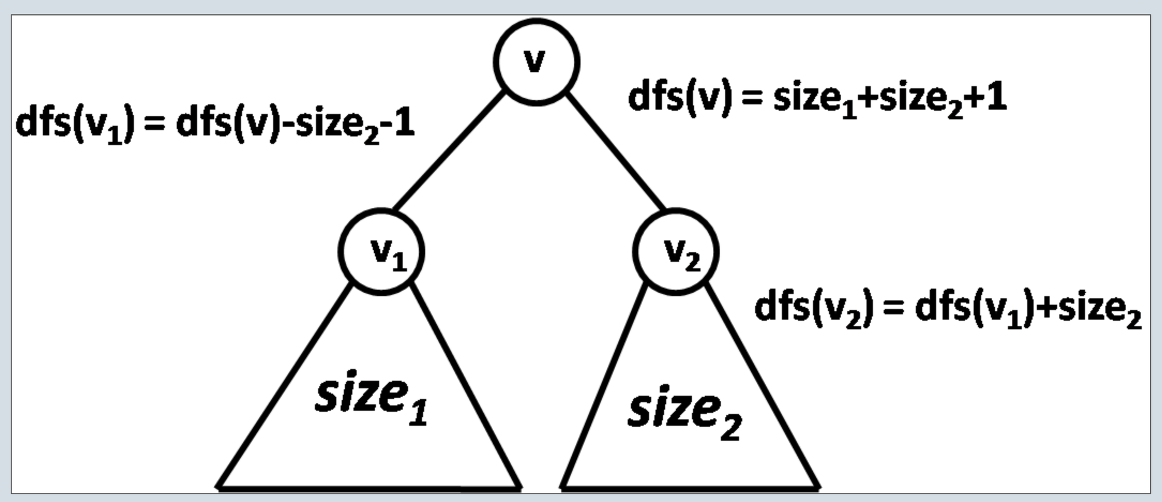
Illustration of one-dimensional merge tree



Terrain seen from above along with its merge tree

Preprocessing The Merge Tree

- Compute the dfs-ordering of the merge tree. We need the ordering to avoid random access.



Algorithm

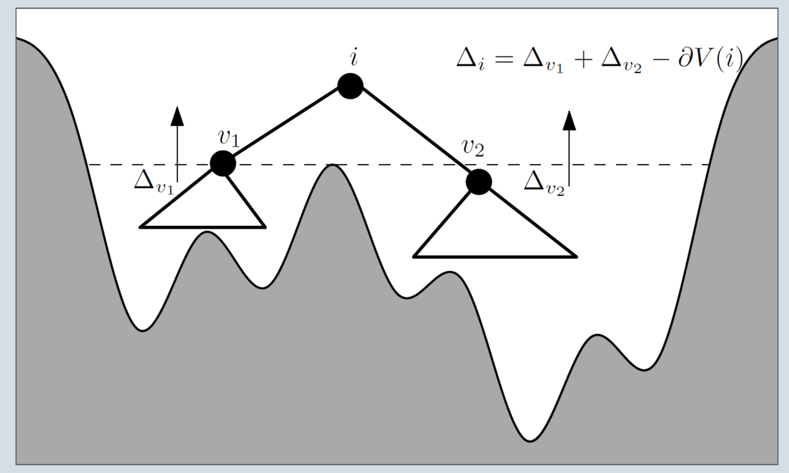
Sweep-up phase (rising water)

– Input: The dfs-layout of merge tree where each leaf is annotated with how much water has accumulated in the sink.

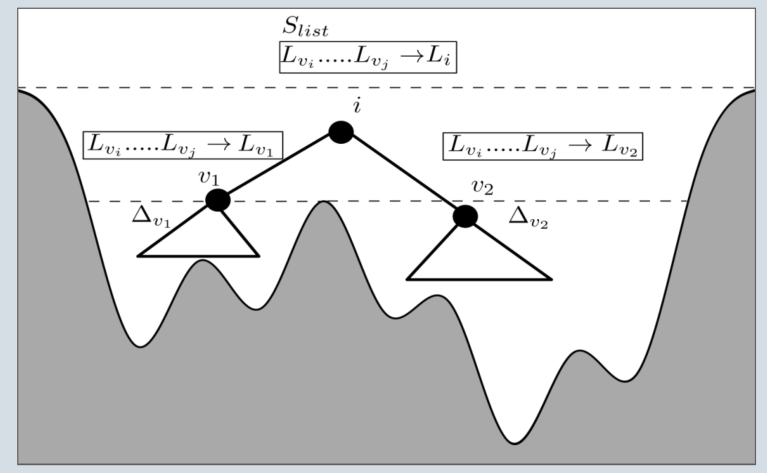
– Output: Compute for each node how much excess water(Δ) is standing for its two children's sub-trees (two depressions).

Algorithm

- Scan dfs-layout and compute for each vertex it's excess water(Δ), as shown in figure, and output it.



Rising water phase



Distribution of excess water phase

Sweep-down phase (distribution of excess water)

– Input: Output of sweep-up phase and the dfs-layout.

– Output: Water over each depression.

Algorithm

- Scan dfs-layout and sweep-up phase output while maintaining S_{list} (a stack of lists, where a list contains all excess water to be distributed for the leaves associated with a depression)
  – If water coming from the watershed hierarchy of a depression is greater then the remaining water volume, we signal its child depression as full.
  – Otherwise, we partition the list of excess water corresponding to the saddle point and check if one of the child depression is filled, if so we send the remaining excess water to the other depression corresponding to the saddle point and push the lists for the child depressions and update its own water.

References

[1] Y. Liu and J. Snoeyink. *Flooding Triangulated Terrain*. International Symposium on Spatial Data Handling, 2005.

[2] L. Arge, M. Revsbæk, N. Zeh. *I/O-efficient Computation of Water Flow Across a Terrain*. Symposium on Computational Geometry, 2010.

[3] L. Arge, S. Raza, M. Revsbæk. *I/O-efficient Water Flow and Terrain Flood Risk with Non-uniform Rain Events*. Manuscript under preparation, 2014.

MADALGO – Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation