Reviewer Response

Richard Barnes

We thank the two referees for their constructive comments on our work. In our response, we have highlighted the reviewer's comments in blue, our response to these comments is highlighted in green, and the changes made to the manuscript are in black.

1 Reviewer #1

> I think this draft only presents the method part of this study without adequate support from real-world hydrological applications. Although in the Application section, the authors list several potential terrain analyzing processes that this new data structure can be beneficial to, there is no concrete evidence to demonstrate the improvement brought by this new data structure. The only result presented with quantified information is Table 1, which only shows the time requirement of implementing this algorithm on data sets in different sizes. To make this paper complete as an individual journal article itself, the authors need to compare the efficiency of running different applications(such as pit filling) without introducing this new depression hierarchy structure. Even with another paper submitted, it only focuses on 6.5 Flow Modelling, but evidence for application in section 6.1-6.4 is still missing.

> We will be happy to include some time comparisons for pit filling with and without the depression hierarchy structure in the updated paper (i.e. application 6.1). However, the algorithm does considerably more work than simple pit filling: it produces a data structure that can be used to analyze and operate on nested depressions. Therefore, a direct comparison of the wall-time of the new algorithm versus simple pit filling is not really appropriate: these are separate operations for separate things. This is also true of depression carving.
>
> We will also include a table with more information about the depression statistics (application 6.4) for the examples processed. These data are retained within the depression hierarchy and would not be available when performing simple pit filling. We will also include an example of depression filtering (6.3) to selectively remove depressions below a certain threshold, and an example of depression carving (6.2).

Table 1 now includes timing comparisons between the depression hierarchy algorithm and a set of depression-filling algorithms. We have added a figure (Figure 6) which depicts the effect of filtering and carving depressions.

> If it is possible, try to reconcile the 1-d topographic profiles used in Figure 1 & 2 and Figure 3 & 4 as a single dataset/profile. Illustrating the points in the context by jumping back and forth between two examples is confusing. For example, the majority of Section 3.4 Hierarchy Construction is explained

with the case presented within Figure 3 and 4. Then in line 12-13 of Page 10, the authors suddenly refer to Figure 1 to illustrate some point. The thing is that the outlet key assignment is only given in Figure 3 and 4. Then the point the authors make ("As an example, in Figure 1, 5 drains into 8, but the cells that actually constitute the outlet will be labeled 2 and 6") is not that obvious to readers.

It was impractical to use the exact same topography (and hence, topology) for all four of these figures, since it was necessary to show several different possible cases in the depression tree in Figure 1. Using this full topography would have made Figures 3 and 4 unwieldy. However, we will experiment with remaking Figures 3 and 4 so that they represent the same topography as seen on the right-hand side of Figures 1 and 2, i.e. the depressions labelled 9–15 in the first two figures. This may make it easier for a reader to follow the changes through these four figures. We will update the references to Figure 1 in these later parts of the text to refer to a similar case in Figure 4, so that the reader does not have to jump back as far. We hope that the point made here will be clearer to a reader when viewing Figure 4, which depicts the colours associated with each depression label.

We have modified Figure 1 for clarity, Figure 2 so that the depressions correspond to those depicted in Figure 1, and Figure 4 so that it shows a worked example of the process being applied to the right-hand side of Figure 2. The text and captions have been updated accordingly. We have also added a new figure (Figure 3) depicting the depressions in a more intuitive form.

Figure 3(f) "an outlet of elevation 3" A specific elevation number ("3") suddenly appears without any indication in the context. If these numbers need to be maintained, please add a y-axis with labels to the subplot. Also, try to use different number formats (like with circles) to differentiate those representing the PQ popup order from those representing the spilling elevations of the outlets.

We have added elevations along the y-axis of the plots in Figures 3 and 4.

We have added elevations along the y-axis of the plots in Figures 4 and 5.

Page 7 Line 29–30 "Figure 3h-i depicts the front of a traversal, in this case, expanding the area that is defined as OCEAN. We discuss both possibilities below." The placement of this sentence seems odd. It is not closely connected to previous statements in this paragraph, which explains cells assigned with given depression labels.

This sentence was referring to a specific case in which cells are assigned the depression label associated with the OCEAN depression. It is one of the possible cases for depression label assignment. Nonetheless, this sentence has now been changed to reflect the changed topography seen in Figure 3.

The sentence has been changed to reflect the new topography generated in response to the reviewers' other comments.

> Page 8 Line 23-24 “If any entry for an outlet is already present, only the outlet of lower elevation is retained; this is important, as it allows for the realistic case of multiple spillways that exist between two depressions.” This statement seems contradictory. The former part states that the value of the lowest joining cell will overwrite the value in the hash map as the outlet value. Since the value of this hash map is a single value instead of an array. How can it keep track of the multiple-spillway case the authors discuss in the later part?

> We have added a sentence to the text clarifying what was meant here. To clarify here, we are referring to cases that would be common in the real world, in which two depressions meet one another at multiple cells. In other words, there is a ridge between two depressions, and each time that a cell along this ridge is processed in the depression hierarchy, it will detect that it is a potential link between the two depressions. Once a potential link has been detected, it will check to see whether an outlet has already been recorded in the hash map. If so, it will replace the recorded value only if the new cell has a lower elevation. In this way, only the true outlet, which has the lowest elevation, is recorded between these two depressions.

We have modified the section to read:

> If any entry for an outlet is already present, only the outlet of lower elevation is retained. Two depressions may share a border across multiple cells (i.e. there are multiple potential spillways), but only the location of the lowest outlet is recorded since this is the only location where overflow from one depression to the other would naturally occur.

> Page 8 Line 24–25 “but the one-dimensional elevation profile in Figure 3 cannot depict the case of multiple outlets of different elevation.” Then can you add a figure of a two-dimensional domain to clarify the multiple outlets case?

> We are not sure that a figure is needed for this concept, which is a relatively small part of the overall algorithm, now that it has been further clarified. The multiple outlets case is simply any case in which depression 1 and depression 2 (for example) border one another at more than one single cell, which will often be the case. Any location at which a cell from depression 1 and depression 2 are adjacent to one another is a potential outlet. Each of these potential outlets may have a different elevation. Only the outlet with the lowest elevation is recorded.

No changes were made in response to this comment, as discussed above.

> Page 8 Line 28 “assigned each of them a flow direction” As a byproduct, the flow directions are rarely discussed during the depression assignment process, which is understandable. The only place I saw that flow directions were mentioned is in Line 10 (P8): “Flowdir(n) is set to point to c”. If I understand it correctly, in this way, the flow directions are assigned locally, which means each cell will drain to the lowest local pit following the assigned directions. This point needs to be emphasized here because they are different from the typical flow directions we have seen draining water to the ocean.

> It is correct that flow directions are assigned such that each cell drains to the lowest local pit following the assigned directions. However, this method of flow direction assignment is not vastly different from other typical flow directions used in other algorithms. While there are some algorithms that always route water to the ocean, for example, those that use a least-cost path to the ocean, ‘typical flow direction algorithms simply assign flow direction in the local downslope direction. These algorithms rely on a user having already filled depressions prior to calculating flow directions and performing flow routing. This is the key difference in our method: we are not simply filling all depressions prior to calculating flow across the landscape. Instead, we are particularly interested in what happens within the depressions. In the revised paper we will clarify this point.

The revised paper now states early on, in the definition of variables:

> The algorithm returns flow directions as an output. They are determined in a standard way by requiring that each cell direct its flow in a D8 fashion to the lowest of its eight neighbors. In the case that the lowest neighbor is not unique, one is chosen arbitrarily.

The section the reviewer refers to has been modified to read

> [...] *Flowdir(n)* is set to point to c. This ensures that flow follows the path of steepest descent since c is the lowest unexplored cell in the DEM.

> Page 9 Figure 4(d) “Were M part of another depression (call it 6) that had previously found an outlet to the ocean, then 5s parent would be the depression identified by the label of M, which would be a leaf of the tree rooted by 6. This would ensure that 5 would drain into the bottom of 6 before overflowing out of it.” An actual figure could be helpful to illustrate this hypothetical scenario. If the authors think its not necessary, remove this statement should be fine.

> This refers to the ‘ocean-linked case and is shown in Figure 1, where depression 5 is linked to the ocean via depression 6. However, this caption has now changed due to the changes to Figure 4.

The caption of Figure 4 has been updated and no longer contains the hypothetical the reviewer mentions.

> Adding a reference to a draft in preparation is not acceptable. Please remove the reference to “Barnes, R., Callaghan, K., and Wickert, A.: Computing water flow through complex landscapes, part 3: Fill-Merge-Spill: Flow routing in depression hierarchies, In preparation, 2019.”

> We have removed reference to this in-preparation paper. We will restore these references if the in-preparation paper is submitted before we submit our revised draft.
>
> We thank the reviewer for their diligence and detailed comments.

The reference is removed.

2 Review #2: Dr. Schwanghart

> A minor issue is that I found the algorithms easier to understand when reading the captions of the figures. Perhaps, the extensive captions might be better placed in the main text.

> We have tried to provide multiple ways to understand the algorithms, including the description in the text, pictures via the figures, figure captions, and extensively-commented source code. Our hope is that at least one of these methods will prove effective for each reader. It might be that in your case the figure captions worked best. In the revised paper, we'll try to duplicate or move material from the captions to the main text where appropriate.

Moving the caption material into the main text seemed to distract from the flow of the explanations there, so we have opted not to change the captions in response to this comment.

> My concern is that the paper may be too technical for the readership of ESURF. While I see that the authors are planning a third part that will highlight how the developed soft-ware can be used to accelerate hydrological models, I think that the paper would benefit from more illustrations/examples/interpretations of the output of these algorithms. How do sink networks differ between different regions (glacially sculpted low-land regions vs. dryland regions) or different DEMs? Illustrating potential geomorphological applications would be a nice addition to the paper and would considerably widen its readership.

> This is a good comment, and indeed one that we wrestled with before deciding to focus on a more abstract approach. From both this comment and some from Reviewer 1, it seems that at least one specific example would be valuable to demonstrate more tangibly the application of the depression hierarchy to real landscapes. We are considering two candidates for this example: the Illinois landscape used by Callaghan and Wickert (2019, a companion paper), and Madagascar, which has diverse topography but is small enough to allow us to describe its exemplary features without diluting the technical focus of this paper. Our choice on which to include in the ultimate analysis in the resubmitted draft will be based on which provides a more useful and intuitive visual description of the depression hierarchy.

We have added a figure (Figure 5) which shows one result of applying the depression hierarchy algorithm to Madagascar.

> The empirical tests are done on an high-performance computer. Why? As far as I understand, the code is not (yet) fully optimized for using parallel infrastructure. I wonder how timings of the algorithm would scale on "normal" desktop computer.

> The largest dataset we test requires approximately 15GB of RAM, which is larger than our laptops (8GB). Since we are located at different institutions, HPC environments are a convenient way to collaborate. The scaling of the algorithm is unaffected by the compute environment, since this is an instrinsic property of the algorithm.

> We thank Dr. Schwanghart for his thoughtful review

As a follow-up on this, in 2015 a near-analogue to the setup used by the Comet computer could be purchased as the HP Z440 Workstation (see https://www.amazon.com/HP-Z440-Workstation-Certified-Refurbished/dp/B07JRDR8QT).

Computing water flow through complex landscapes, Part 2: Finding hierarchies in depressions and morphological segmentations

Richard Barnes[1,2,3], Kerry L. Callaghan[4], and Andrew D. Wickert[4,5]

[1]Energy & Resources Group (ERG), University of California, Berkeley, USA
[2]Electrical Engineering & Computer Science, University of California, Berkeley, USA
[3]Berkeley Institute for Data Science (BIDS), University of California, Berkeley, USA
[4]Department of Earth & Environmental Sciences, University of Minnesota, Minneapolis, USA
[5]Saint Anthony Falls Laboratory, University of Minnesota, Minneapolis, USA

Correspondence: Richard Barnes (richard.barnes@berkeley.edu)

Abstract. Depressions—inwardly-draining regions of digital elevation models—present difficulties for terrain analysis and hydrological modeling. Analogous "depressions" also arise in image processing and morphological segmentation where they may represent noise, features of interest, or both. Here we provide a new data structure—the depression hierarchy—that captures the full topologic and topographic complexity of depressions in a region. We treat depressions as networks, in a way that is analogous to surface-water flow paths, in which individual sub-depressions merge together to form meta-depressions in a process that continues until they begin to drain externally. The hierarchy can be used to selectively fill or breach depressions, or to accelerate dynamic models of hydrological flow. Complete, well-commented, open-source code and correctness tests are available on Github and Zenodo.

1 Introduction

Depressions (see Lindsay, 2015, for a typology) are ~~inwardly-draining~~ inward-draining regions of a DEM that lack an outlet to an ocean, map edge, or some other designated boundary. Quantifying and understanding these depressions and their structure can advance our understanding of wetlands (Wu and Lane, 2016), subglacial hydrology (Humbert et al., 2018) and its links to sea-level rise (Calov et al., 2018), microscale water retention in soils (Valtera and Schaetzl, 2017), and flood extent (Nobre et al., 2016). This is particularly significant because lakes and wetlands host biodiversity, provide ecosystem services including denitrification (Hansen et al., 2018) and recreation (Costanza et al., 2006; Keeler et al., 2015), and impact sediment dynamics (Wickert et al., 2019; Mishra et al., 2019) and drainage-network realignment (Carson et al., 2018).

Likewise, in image processing and segmentation, regions of differing image intensity and color can be modeled as depressions that represent either noise or features of interest. In this context, geomorphological algorithms for depression-handling (e.g., Barnes et al., 2014b) have been applied to the cosmic microwave background radiation (Giri et al., 2017), nanoparticle chemistry (Svoboda et al., 2018), biological membranes (Kulbacki et al., 2017), road-car segmentation (Beucher, 1994), murder and crime statistics (Khisha et al., 2017), remote sensing of buildings (Golovanov et al., 2018), neuron mapping (Iascone et al., 2018), and metal defect mapping (Blikhars'kyi and Obukh, 2018). This multidisciplinary set of uses demonstrates the broad potential of a generalized algorithm that can compute depressions and their topology.

Depressions complicate algorithms for geomorphological and terrain analysis, as well as hydrological modeling. Many common methods route flow using only information about local gradients, and enforce downgradient flow (O'Callaghan and Mark, 1984; Mark, 1987; Freeman, 1991; Quinn et al., 1991; Holmgren, 1994; Tarboton, 1997; Seibert and McGlynn, 2007; Orlandini and Moretti, 2009; Peckham, 2013). As a result, flow entering a depression cannot leave; in an extreme case, this could cause a continent-scale river, such as the entire Mississippi, to disappear into a small depression.

Correctly routing flow in depressions, and flat areas, requires non-local information. Depressions—especially those in high-resolution datasets—are often treated as aberrations. Algorithms to remove these features either flood them until they are filled and flow paths can reconnect (Barnes et al., 2014b); carve deep channels through them either by modifying the DEM's data directly or by altering flow directions to simulate carving (Lindsay, 2015; Martz and Garbrecht, 1998), as in `r.watershed`; or perform some combination of these two options (Grimaldi et al., 2007; Lindsay and Creed, 2005a; Lindsay, 2015; Schwanghart and Scherler, 2017). However, depressions may also represent actual landscape features such as prairie potholes, lakes, wetlands, and soil microrelief (Shaw et al., 2012, 2013; Valtera and Schaetzl, 2017). When this is the case, depressions should be retained and leveraged to improve models ~~(Callaghan and Wickert, 2019; Barnes et al., 2019; Arnold, 2010; Hansen et al., 2018)~~ ~~.~~ (Callaghan and Wickert, 2019; Arnold, 2010; Hansen et al., 2018).

Incorporating depressions into drainage analyses is non-trivial. Depressions may have complex topographic structure. For instance, Vulcan Point is an island within Main Crater Lake, which is on Taal Island in Lake Taal, which itself is on the island of Luzon in the Philippines. As another example, Lake Nipigon (Ontario, Canada) contains Kelvin Island, which in turn contains Firth Lake, which hosts its own islands. High-resolution data can exacerbate the issue by introducing high-frequency noise that cannot be reliably distinguished from actual topographic features (Lindsay and Creed, 2005b, c).

This problem is similar to one in image processing, in which a computer must reassemble multiple distinct-looking features into a meaningful whole. For example, over-segmentation can cause features such as cars to be fragmented into many small pieces (Beucher, 1994). Understanding the relationships between topographic depressions can aid the general goal of building relational hierarchies among adjacent objects, and in so doing can reduce over-segmentation by providing a principled way of merging small features and extracting composite features of interest.

In response to these challenges, we present an efficient method for constructing a *Depression Hierarchy*: a data structure that captures the full topologic and topographic complexity of depressions in a region. The hierarchy can be used to selectively fill or breach depressions, or to accelerate dynamic models of hydrological flow. ~~This latter property is demonstrated in an accompanying paper (Barnes et al., 2019).~~

Prior researchers have developed structures with similar purpose—and in some cases, function—to depression hierarchies, but these either yield nondeterministic results, are not developed in a way to permit dynamic water flow through a set of nested depressions, or are prohibitively slow. Beucher (1994) presents a hierarchical segmentation algorithm for images using a "waterfall" approach that merges adjacent features by filling smaller local minima while maintaining significant minima that can act as a sink over larger regions. However, this "waterfall" algorithm does not produce a persistent data structure to be used in subsequent operations nor does it construct a full hierarchy as an intermediate product. Salembier and Pardas (1994) use a kind of hierarchical segmentation, but generate the hierarchy via repeated simplification of the source image. These simplifications are sufficient to segment features, but, in a hydrological context, can lead to unacceptable degradation of terrain information. Arnold (2010) presents a similar algorithm to the one developed here. However, no source code is provided, the generated hierarchy is not formalized, and the algorithm generates circular topologies that require correction. Wu et al. (2015) and Wu and Lane (2016) develop a method for extracting depression hierarchies by first smoothing a DEM and then extracting vector contour lines from it. They then analyze the topological relationship of the contours. Wu et al. (2018) build on this approach by developing a method to move a horizontal plane upwards through topography and noting the elevations at which depressions combine. Both methods are inaccurate due to their reliance on discrete vertical steps—that is, both the contour intervals and the finite distance over which the plane is shifted upwards before checking for joined depressions. The latter method is also inefficient because it requires every cell of the terrain model to be parsed after each movement of the plane. Cordonnier et al. (2018) present an algorithm based on minimum spanning trees in a planar graph, which can be used to construct a hierarchy of depressions. However, the resulting data structure is not well-described, and the algorithm ~~and~~ has been optimized for use in contexts in which the dynamic flow of water (described at greater length in §6.5) does not need to be modeled explicitly. Callaghan and Wickert (2019), in a companion paper to this, describe an approach to move water among cells across the landscape. This virtual water floods depressions, but its cell-by-cell computation is expensive and slow.

The depression hierarchy presented in this paper is differentiated by several features. (1) Correctness: the DEM does not require preprocessing and no arbitrary step length needs to be defined. (2) Efficiency: the algorithm operates in $O(N)$ time. (3) Degree of documentation: in addition to this paper, 51% of the lines in the accompanying source code are or contain comments.(4) Availability of source code: the completed, well-commented source code for the algorithms described here, along with associated makefiles and correctness tests, is available on both Github and Zenodo (Barnes and Callaghan, 2019). (5) Suitability for dynamic models: ~~an accompanying paper (Barnes et al., 2019) describes how~~ by defining hydrological connectivity across a landscape, the depression hierarchy can be leveraged to accelerate hydrological models.

2 The Depression Hierarchy

The depression hierarchy consists of a forest of binary trees, as shown in Figure 1 and illustrated in Figures Figure 2 and Figure 3. The leaves of the trees are the smallest, most deeply-nested depressions (Figure 2). During flooding, these would fill first. Non-leaf nodes are formed when two depressions overflow into each other. Here, this non-leaf node is termed a "parent" and each of the overflowing depressions—be they leaves or no—is termed a "child". Eventually, a depression fills to the level at which additional "water" would escape the initial set of depressions and flow into either the ocean or another binary tree of depressions that already has a path to the ocean. For example, in Figure 1, node ~~5~~ 12 flows into leaf-node ~~6~~4, which (indirectly) flows into the ocean. When this happens, one binary tree cannot become the child of the other, since they are not topographically nested. Instead, the root (the topmost node) of the tree that does not yet link to the ocean takes one of the leaf nodes of the other tree as its parent and that leaf node makes an *oceanlink* in the reverse direction. In addition to the primary

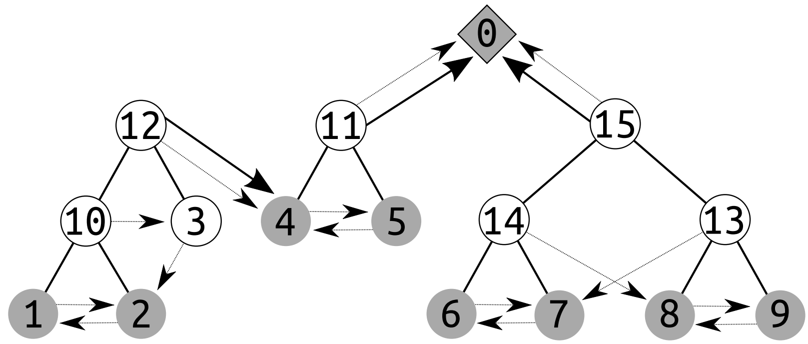


Figure 1. ~~**A depression hierarchy.**~~ **A depression hierarchy** of the topography depicted in Figure 2 generated by a process shown in Figure 4. Dotted arrows indicate *geolinks*, solid lines indicate links between depressions and meta-depressions, solid arrows indicate *oceanlinks*. (~~5~~11), (~~8~~12), and (15) are all *roots* of binary trees. In each of several binary trees, water fills the tree from bottom to top before overflowing into a neighboring tree or the ocean. As (1) fills up, it overflows through its *geolink* (the dotted arrow) into (2). Both of these then begin to fill (~~3~~10), a larger depression containing both, as indicated by the solid lines between (~~3~~10) and both (1) and (2). When (~~3~~10) overflows, it begins to fill (~~4~~3). When (~~4~~3) overflows, it tries to fill (2), but finds it full. Therefore, both (3) and (~~4~~10) begin to fill (~~5~~12). Topologically, (~~5~~12) flows into (~~6~~4); however, the reverse is not true. This is because the depression tree rooted at (~~5~~12) must actually be uphill of (~~6~~4). Thus, (~~5~~12) notes that (~~6~~4) is its parent (solid arrow) and the depression into which it overflows (geolink, dotted arrow), and (~~6~~4) makes an *oceanlink* to (~~5~~12), as implied by the solid arrow, but does not count it as a child. Both (~~8~~11) and (15) flow into the ocean (0), which may have an infinite number of children. A cross-sectional view of the landscape described by this depression hierarchy is shown in Figure 2.

structure of the depression hierarchy (solid lines in Figure 1), we define a set of *geolinks* that tie an overflowing depression with the depression into which it overflows. As in a threaded binary tree (Fenner and Loizou, 1984), these links can be used to accelerate traversals by eliminating recursion.

3 The Algorithm

The depression-hierarchy algorithm proceeds in several stages, as detailed below: (1) ocean identification, (2) pit-cell identification, (3) depression assignment, and (4) hierarchy construction. As a side effect, the algorithm determines flow directions. We describe the algorithm with reference to Figure 4.

Several bookkeeping data structures are required to compute the depression hierarchy. These are:

- *DEM*: A 2D array indicating the elevation of each cell, or, in image segmentation, its intensity. The data type is arbitrary.
- *Label*: An array with the same shape as *DEM* indicating which depression each cell belongs to. Initially, all cells are labeled with the special value NoDep.
- *Flowdir*: An array with the same shape as *DEM* that indicates the flow direction of each cell. Initially, all cells are labeled with the special value NoFlow. The ~~flow directions of cells are determined by the algorithm.~~ algorithm returns flow

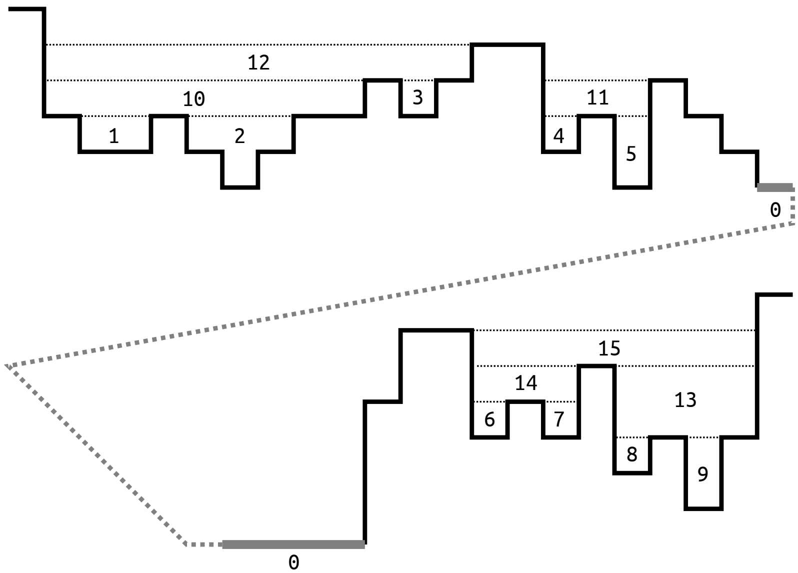


Figure 2. 1D topography representing the depression hierarchy presented in Figure 1. Solid black lines represent topography. The thick gray line represents the ocean, and the dotted line indicates that this figure represents a single continuous profile that has been split to better fit on the page. Following Figure 1, numbers mark depressions and meta-depressions, and "0" marks the ocean.

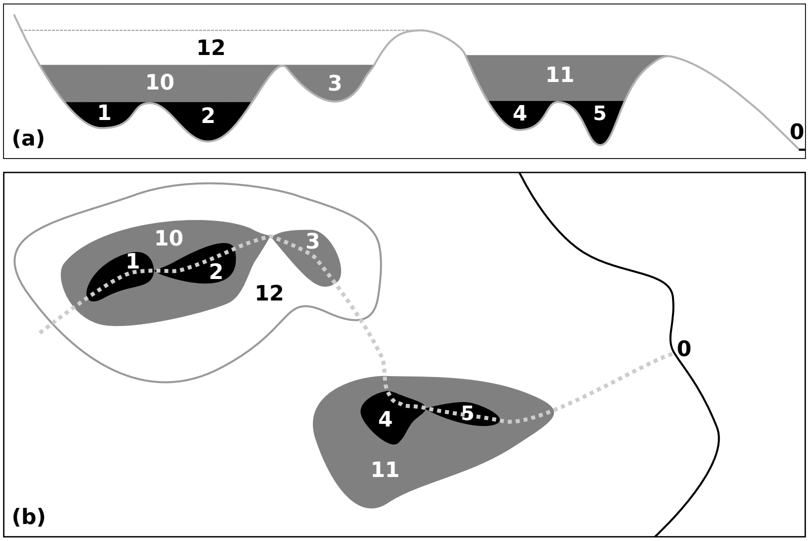


Figure 3. Cartoon of the left-hand side of the depression hierarchy from Figures Figure 1 and Figure 2. In this case, we consider depressions to be represented in topography. **(a)** Cross-sectional view. **(b)** Map view. Numbering follows Figures 1 and 2.

directions as an output. They are determined in a standard way by requiring that each cell direct its flow in a D8 fashion to the lowest of its eight neighbors. In the case that the lowest neighbor is not unique, one is chosen arbitrarily.

- *PQ*: A priority queue that orders cells such that the cell of lowest elevation is always popped (i.e., removed from the queue) first. In the event that two cells have the same elevation, the cell added most recently is popped first.
- *DH*: The depression hierarchy, a forest of binary trees that store the hierarchical relationships among depressions alongside metadata about each depression.
- *OC*: A hash map of depression outlets. The hash map is a relational data structure that links keys to values (Cormen et al., pp. 253–285). Outlets are identified by the two depressions they join, so the depressions' ids are used as the hash map's key while the value contains information such as the spill elevation. Though many potential outlets between two depressions may be found, lower outlets overwrite higher ones so that only the lowest is retained.
- *DS*: A disjoint set data structure (also known as a "union find", "set union", or "merge-find") (Cormen et al., pp. 561–585) is used to quickly determine the root of a tree of depressions.

3.1 Ocean Identification

All cells must have a drainage path to the "ocean". This path may be simple and direct when flow down a river terminates directly in an ocean. It can also be indirect, when flow enters a depression, fills the depression, and then spills out towards the ocean, possibly entering more depressions on the way.

All cells that constitute the ocean must be marked in *Label* with the special value OCEAN. For some applications, OCEAN cells can be determined by comparing the elevations with a value for sea level. In other applications, especially in landlocked regions and image segmentation applications, the edge cells of the DEM can be marked as OCEAN to ensure that flow reaches the edge of the area of interest.

All ocean cells are added to the priority queue *PQ* as they are identified. A single depression representing the entire ocean is added to *DH*. Figure 4a depicts this initial state before the start of the "flooding" process.

3.2 Pit Cell Identification

After the ocean—the ultimate sink—is selected, the depression-hierarchy algorithm must identify all of the pits in the *DEM* that can act as local sinks for water. For the purposes of this paper, a pit cell is a cell that does not drain to any of its neighbours: all of the neighbours' elevations are equal to or greater than that of the pit. All pit cells are added to *PQ* as they are identified, as depicted in Figure 4a.

3.3 Depression Assignment

Once all pit and ocean cells are identified, the depression-hierarchy algorithm places them in *PQ*. The general strategy now is to pop (i.e., select and remove) cells from *PQ*, label the popped cells' unlabeled neighbours, add the previously unlabeled

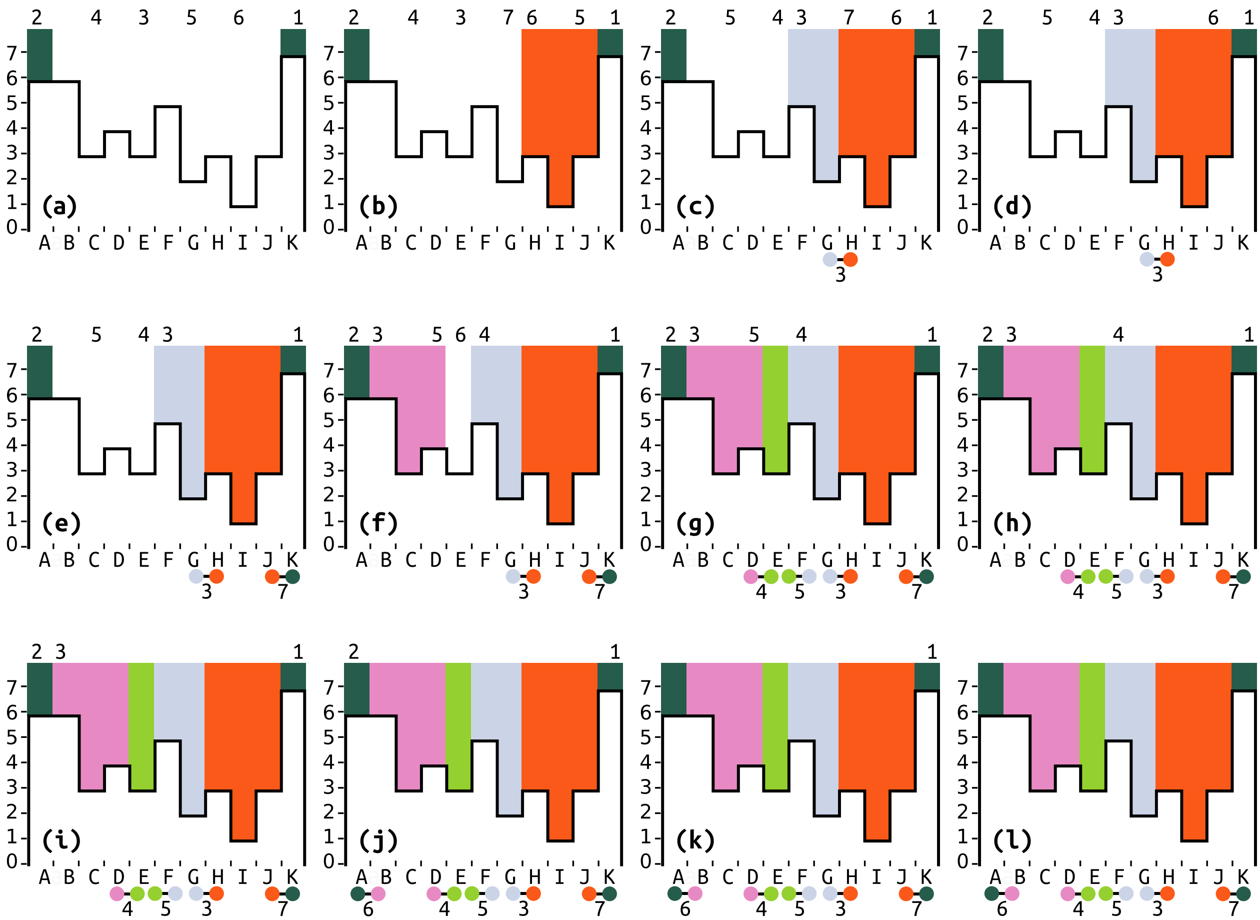


Figure 4. ~~**Illustration of the "Flooding" Process.**~~ **Illustration of the "Flooding" Process** as applied to the right-hand side of the topography shown in Figure 2. Boldface lowercase letters indicate progression through time. Capital letters label cells. Numbers at the top indicate the cells' positions (if any) in a priority-queue PQ. The little barbells indicate outlets between depressions with numbers to indicate their elevations. The black lines outlining the white regions indicate elevation, with values shown on the y-axis. Colors represent labels. **(a)** Initialization. C, E, ~~F~~ G, ~~I,~~ and ~~L~~ I are pit cells (they have no lower neighbours), so they are added to PQ. A and ~~N~~ K are ocean cells, so they are labeled as such and added to PQ. ~~E and F are~~ I is the lowest ~~cells~~ cell and so ~~have~~ has the highest ~~priorities. E is arbitrarily given the highest~~ priority. **(b)** ~~E~~ I is popped. It is not already labeled, so it is a new depression and given a new label. ~~D~~ H and ~~F~~ J are labeled and added to PQ. ~~F was already in PQ~~ H and J have the same elevation as C and E, but ~~had not already~~ since they have been ~~labeled~~ added to the PQ more recently, ~~so it~~ their priority is ~~now in PQ twice~~ higher. Arbitrarily, H is given the higher priority. **(c)** ~~F~~ G is popped. ~~E~~ H shares ~~F~~ I's label, so it is ignored. ~~G~~ F is labeled and added to PQ. An outlet between G ~~has the same elevation as A~~ and ~~N, but, since it has been added to PQ more recently, its priority~~ H is ~~higher.~~ recorded with elevation 3. **(d)** ~~F~~ H is popped ~~again~~. ~~But this time it~~ It is already labeled, so it is not altered. ~~F~~ H's neighbours have already been labeled and so nothing is done to them either. ~~Popping F a second time~~ Nothing new is added to the PQ. The outlet between blue and orange has already been noted, so no ~~effect~~ new outlet is recorded. **(e)** ~~I~~ J is popped. It is ~~not~~ already labeled, so it is ~~a new depression and given a new label. H and J are labeled and added to PQ. **(f)** H is popped. G and I are already labeled, so they are~~ not ~~added to PQ~~ altered. ~~G's~~ Its neighbour, K, has a different label ~~differs from H's~~ (ocean), so an outlet of elevation ~~3~~ 7 between the two depressions is noted. ~~**(g)**~~ **(f)** ~~G~~ C is popped. ~~F~~ B and ~~H~~ D are ~~already labeled~~ a part of C's depression, so they are ~~not~~ given its label and added to the PQ. ~~An outlet between orange and blue has already been noted, so nothing happens.~~ ~~**(h)**~~ **(g)** ~~N~~ E is popped. It ~~is already labeled OCEAN~~ was not yet labelled, so it is ~~not relabeled~~ given a new label. ~~M is labeled and added to PQ.~~ ~~**(i)** A is popped. It is~~ Its neighbours were already labeled ~~OCEAN~~, so ~~it~~ an outlet of elevation 4 is ~~not relabeled. B~~ noted between D and E, and

neighbours to *PQ*, and repeat this process until *PQ* is empty. Once *PQ* is empty, all of the cells of *DEM* will have been visited. This operation is similar to the Priority-Flood algorithm (Barnes et al., 2014b).

For each cell c that is popped, one of three possibilities must be true:

1. *Label(c)*=OCEAN.
2. *Label(c)*=NoDep.
3. Neither of the above.

If *Label(c)*=OCEAN, the cell c is either part of the ocean or has already been proven to flow to the ocean. In this case, nothing more need be done. ~~h and i depict this case.~~

If *Label(c)*=NoDep, cell c is a pit cell. Although all cells begin with the NoDep label, cells label their neighbours as they are popped from *PQ*, the cell most recently added to *PQ* is the next one to be removed, and all OCEAN cells are labeled as OCEAN. Therefore, finding a NoDep cell is possible only if c is a pit cell. Within a flat area that is larger than one cell wide, only one cell will be labeled as the pit~~(as depicted in d)~~. As each pit cell is found, a new depression is added to *DH* and its label is applied to *Label(c)*. Figure 4~~a–b, d–e, and k–l~~a, c, f, and g depict this.

If *Label(c)* is neither OCEAN nor NoDep, cell c has already been assigned to a depression. This means either that: (a) c is on the frontier of the traversal, and will therefore have neighbours that have not yet been seen and must be added to *PQ*, ~~or~~ (b) that c was part of a flat that has already been processed and therefore all its neighbours have been seen and none should be added to *PQ*~~.~~, or (c) c is at the edge of a depression and its neighbour has been labelled as a different depression. In this last case, c may be the outlet between the two depressions, if it is the lowest link between them. Figure 4~~h–i depicts the front of a traversal, in this case, expanding the area that is defined as OCEAN. We discuss both possibilities~~ d, e, and h–l represent the third case, where a previously labeled cell sees neighbours which are part of a different depression. Of these, subfigures e and j included the discovery of a new outlet. We discuss this further below.

After identifying the state of cell c and modifying it as indicated above, *Label(c)* must be either OCEAN or the label of a depression. If it is a depression, it is one of the leaves in the depression hierarchy (gray circles in Figure 1). If it is ocean, we know that it sits at the upper-most end of the depression hierarchy (gray diamond with black border in Figure 1).

From this point, the next step is to consider how the popped cell c interacts with each of its neighbours, n. As before, there are three distinct possibilities:

1. *Label(n)*=NoDep.
2. *Label(n)*=*Label(c)*.
3. Neither of the above.

If *Label(n)*=NoDep, n has not previously been seen. Accordingly, *Label(n)* is set to *Label(c)*, n is placed into *PQ*, and *Flowdir(n)* is set to point to c. This ensures that flow follows the path of steepest descent since c is the lowest unexplored cell in the DEM. Figure 4~~b–c~~ b depicts one example of this, in which the previously-unlabeled ~~cell "G" is~~ cells "H" and "J" are labeled as part of the orange depression. Another example, provided in Figure 4~~h–i~~c, depicts the previously-unlabeled cell "~~B~~F" being labeled as ~~OCEAN~~a part of the light blue depression.

If *Label(n)=Label(c)*, n is skipped because it has either already been visited or has already been added by another cell. This also ensures that flats are processed only once. For example, in Figure 4~~c–d and i–j provide examples of this~~d, neighbour cell "I" already has the same label as target cell "H", and so it is skipped.

If neither of the above is true, *Label(n)*≠NoDep and *Label(n)*≠*Label(c)*. The remaining possibility is that *Label(n)* equals the label of a depression that is not its newly-popped neighboring cell, c. Therefore, this indicates that two different depressions are meeting. For example, in Figure 4d, neighbour cell "G" already has a different label to target cell "H". "G" retains its different label.

In this final case, we note where two different depressions meet by creating a link between them. To do so, we determine whether the elevation of n or c is higher. The higher of the two is the outlet cell, and its elevation is the depression's spill elevation (that is, the elevation to which water must rise in order to flow out of the depression). The depression-hierarchy algorithm then adds an object containing this information to the hash map *OC*. The contents of *OC* are hashed using the labels of the depressions that are joined by an outlet. If any entry for an outlet is already present, only the outlet of lower elevation is retained~~; this is important, as it allows for the realistic case of multiple spillways that exist between two depressions~~. Two depressions may share a border across multiple cells (i.e. there are multiple potential spillways), but only the location of the lowest outlet is recorded since this is the only location where overflow from one depression to the other would naturally occur. Figure 4~~f, l, and n~~ c, e, g, and j are examples of this, but the one-dimensional elevation profile in Figure 4 cannot depict the case of multiple outlets of different elevation.

After completing this process, the depression assignment algorithm then selects the next cell c from the priority queue and repeats the above set of steps until *PQ* contains no more cells. Upon completion of the depression assignment phase, the algorithm will have visited and labeled all of the cells, assigned each of them a flow direction, and identified the lowest outlet between each adjacent pair of depressions.

3.4 Hierarchy Construction

At this point *Label* associates every cell with the label of a depression corresponding to an entry in *DH*. These entries will form the leaves of the depression hierarchy (gray circles in Figure 1). Each depression contains all of the cells lower than its spill elevation as well as all cells whose flow ultimately terminates somewhere within the depression. Such a set of cells can also be termed a "basin" (Cordonnier et al., 2018). Figure 5a depicts this.

The next order of business is to identify the structure of flow among the depressions. Pairs of depressions that flow into one another—that is, those connected by links in Figure 4—will join to form meta-depressions. The elevations of these meta-depressions extend from the spill elevation (i.e. the height of the sill) between the two depressions to the elevation of the next-lowest contiguous sill. Pairs of meta-depressions can join to form meta-meta-depressions, and so on to the requisite number of nested meta^n-depressions to represent the structure of depressions in the landscape. Not all depressions flow into each other because the binary tree stops growing when its root finds an outlet to the ocean. Therefore, *DH* is a forest of binary

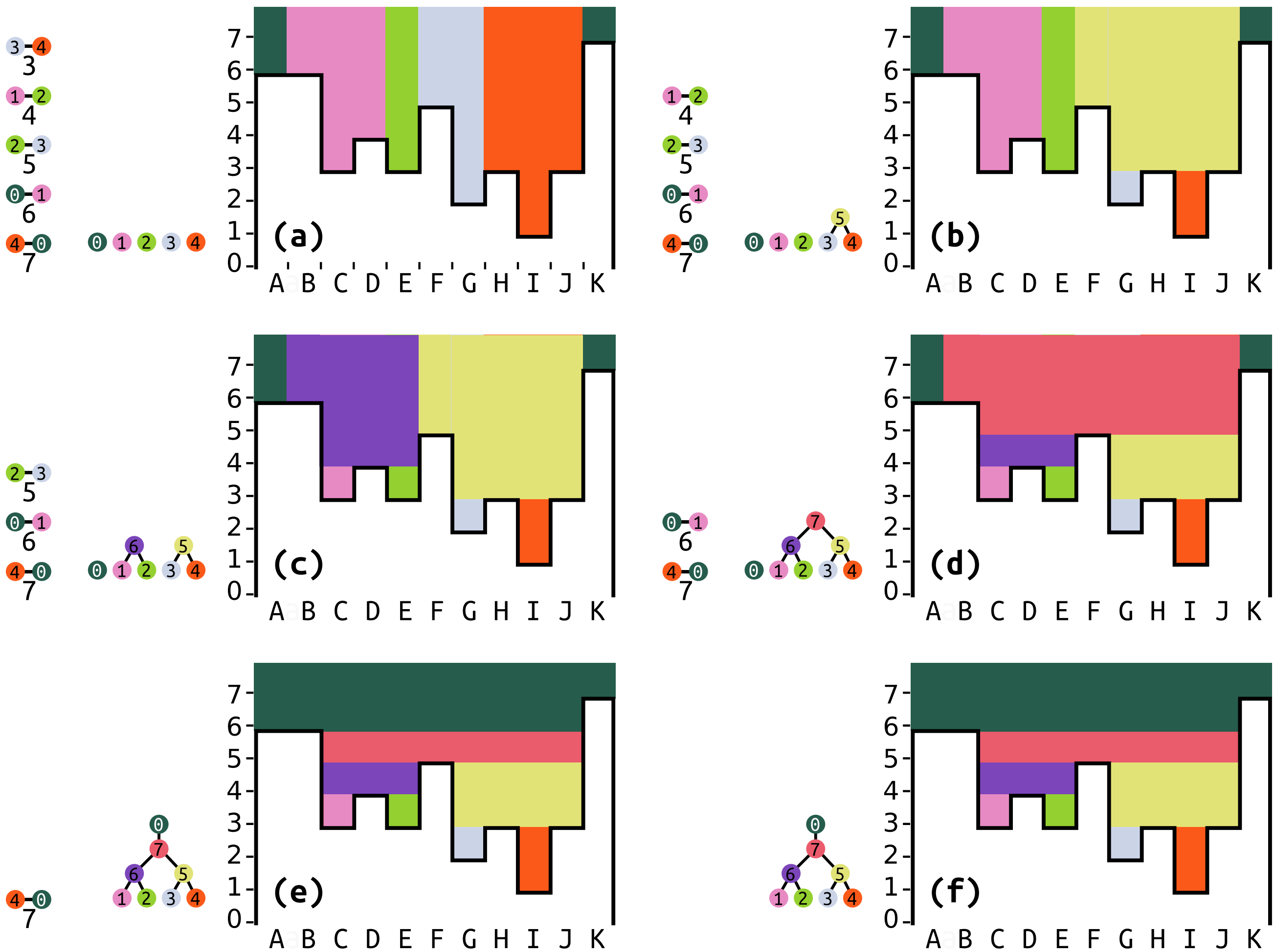


Figure 5. Illustration of the Hierarchy Construction. Boldface lowercase letters indicate progression through time. Capital letters label cells. ~~Numbers at the top indicate the cells' positions (if any) in a priority-queue PQ.~~ The little barbells indicate outlets between depressions with numbers to indicate their elevations. The order of the outlets on the left represent the outlets' positions (if any) in the priority queue PQ. The tree that is progressively built represents the depression hierarchy. The black lines outlining the white regions indicate elevation, with values along the y-axis. Colors represent labels, and the barbels on the left also indicate the depression number associated with each label color. **(a)** Initialization. This reflects the state at the end of Figure 4. The ~~four~~ five outlets have been sorted in order of increasing elevations. ~~Four~~ Five depressions are in the hierarchy, but none of them are connected yet. **(b)** The lowest outlet (between ~~1~~ 3 and ~~2~~4) is popped. A new meta-depression, labeled ~~4~~5, is made and becomes the parent of ~~1~~ 3 and ~~2.~~ 4. All cells in ~~1~~ 3 and ~~2~~ 4 with elevations equal to or greater than the outlet's elevation implicitly become a part of ~~4.~~ 5. **(c)** The new lowest outlet (between ~~3~~ 1 and 2) is popped. ~~We note that 2 now has a parent and should actually be referred to as 4 (the disjoint-set DS accelerates this look-up).~~ A new meta-depression, labeled ~~5~~6, is made and becomes the parent of ~~3~~ 1 and ~~4.~~ 2. All cells in ~~3 and 4 (and, following the depression hierarchy, implicitly including all cells in~~ 1 and 2 ~~)~~ become part of ~~5.~~ 6. **(d)** The lowest outlet is now between ~~3~~ 2 and ~~0. 3~~ 3. We note that 2 now has a parent ~~, so we refer~~ and should actually be referred to ~~it~~ as ~~5~~ 6 (the disjoint-set DS accelerates this look-up) ~~. Because 5 connects~~ while 3 also has a parent and should be referred to ~~the ocean, no~~ as 5. A new meta-depression, labelled 7, is ~~made~~created. ~~5's parent simply becomes 0.~~ **(e)** The lowest outlet is now between ~~cells L~~ 0 and ~~M. Were M part of another depression (call it 6) that had previously found an outlet~~ 1. We refer to 1 by its parent's label, 7. 0 is the ocean, ~~then 5~~so no new meta-depression is made; 7's parent ~~would be the depression identified by the label of M, which would be a leaf of the tree rooted by 6. This would ensure that 5 would drain into the bottom of 6 before overflowing out of it.~~ simply becomes 0. ~~(e)~~**(f)** The

trees, where "forest" refers to the fact that multiple binary trees of depressions and meta-depressions may exist that do not link directly.

All outlets are labeled with reference to the leaves of the binary trees. However, some outlets will drain meta-depressions rather than the leaf depressions that have been used to label the outlets. As an example, in Figure 5, 5 drains into ~~8~~6, but the cells that actually constitute the outlet will be labeled 2 and ~~6.~~ 3.

A fast way to determine the hierarchical structure of a depression set—such as determining that depression ~~5 in~~ 6 in Figure 5 contains depression 2—is to implement a disjoint-set data structure (Galler and Fischer, 1964; Tarjan and van Leeuwen, 1984). A disjoint set, also known as a "union find", "set union", or "merge-find", quickly identifies which of its elements belong to the same set. In the case of the depression hierarchy each depression is an element of the disjoint set, and each of these elements is initially marked as being its own set. Pairs of these sets may be merged such that one set becomes the parent of another. Repeating these merges forms the aforementioned forest of trees.

Merges in a disjoint set are usually performed using "union by rank", but this discards information that is critical to building a depression hierarchy. When combining depressions following "union by rank", the shorter tree is made a child of the taller tree, thereby ensuring that the height of any tree is logarithmically bounded. While this is computationally advantageous, the downside of "union by rank" is that it relabels the root nodes of trees in a way that would prevent us from building the binary trees of the depression hierarchy. We therefore use disjoint-set without "union by rank".

To determine which set hosts an element, we use disjoint-set rules. In so doing, we follow the chain of parents from that element upwards until we encounter an element that is its own parent. For the depression hierarchy, this ultimate parent is a cell that contains an *oceanlink*. Critically for computational efficiency, the disjoint set then points all elements to the appropriate root, ensuring that future queries on any element in the path execute in $O(1)$ time, a technique known as "path compression". With the disjoint set in hand, an outlet's depressions can be updated to reflect the current state of the binary tree by querying each depression label in the disjoint set.

We now sort the outlets in order of increasing elevation and loop over them. Let the depressions linked by a given outlet be called A and B; A and B are both leaf depressions in the binary tree. Further, let $R(A)$ and $R(B)$ be the result of querying the disjoint-set; that is, $R(A)$ and $R(B)$ are the meta-depressions at the roots of the trees to which A and B belong. Based on this starting point, one of the following three options must be true:

1. $R(A) = R(B)$. In this case, the depressions are already part of the same meta-depression and nothing needs to be done (see Figure 5~~e~~f).

2. $R(A)$=OCEAN or $R(B)$=OCEAN. Due to the previous condition, only one of these two depressions may link to the ocean.

3. Neither of the above is true. In this case, two depressions are meeting and must be joined into a meta-depression.

For Case 2 above—either $R(A)$=OCEAN or $R(B)$=OCEAN and $R(A) \neq R(B)$—a few additional steps must be taken to properly build the depression hierarchy. First, for simplicity, we may swap A and B to ensure that B is the depression that links to the ocean ($R(B)$=OCEAN). This means that $R(A)$ will connect to the ocean through $R(B)$. We make a note that $R(A)$

is ocean-linked (linked to the ocean) through B, and also geolinked (physically overflows) into B. This ensures that flow from $R(A)$ has an opportunity to fill the $R(B)$ tree from the bottom up. In *DS*, $R(A)$ is merged as a child of the ocean. Figure 5d depicts this.

For Case 3 above—$R(A) \neq$OCEAN, $R(B) \neq$OCEAN, and $R(A) \neq R(B)$—the algorithm recognizes that two depressions are meeting and that a meta-depression must be formed. To do so, the algorithm adds a new depression to *DH* with children $R(A)$ and $R(B)$, and performs a similar operation on *DS*. Finally, the algorithm notes that $R(A)$ and $R(B)$ overflow into each other through the current outlet, and that $R(A)$ geolinks to B and $R(B)$ geolinks to A. Figure 5b and c depict this.

4 Theoretical Analysis

In computer science, the performance of algorithms can be analyzed based on how they will scale as the amount of data they process increases. In particular, if $f(N)$ is the exact run-time of some complicated algorithm, then $f(N) = O(g(N))$ implies this run-time has an upper bound of $c \cdot g(N)$ for some constant c and some $N \geq N_0$. The notation $f(N) = \Theta(g(N))$ implies both an upper and lower bound, for appropriate constants. Such bounds are referred to as the *time complexity* or time of the algorithm (Skiena, 2012). This same notation can be used to measure the *space complexity* of an algorithm: the amount of memory it requires.

We apply this to the algorithms described here. Let the number of cells in *DEM* be N. The time complexity of finding the ocean is then $O(N)$, since this requires a single pass across the data. Similarly, the time required to find pit cells is $O(N)$. For depression assignment, all N cells must pass through the priority queue. Following Barnes et al. (2014b), we use a radix heap (Akiba, 2015) constructed to have $O(1)$ operations for both integer and floating-point data. Therefore, depression assignment takes $O(N)$ time for both integer and floating-point data. *OC* is a hash table, so additions and accesses are $O(1)$. Additions and accesses to *DS* using only path compression are $\Theta(n + f \cdot (1 + log_{2+f/n} n))$ for n set and f find operations (Cormen et al., pp. 571–572). Since depression merges are always directly preceded by find operations, n and f are small constants, so manipulations on *DS* take $O(N)$ time. Finally, all of the outlets need to be processed in order to build the forest of binary trees. The number of outlets is unknown, but certainly has an $O(N)$ worst case. Therefore, the entire algorithm runs in $O(N)$ space and time.

5 An Alternative Design

Using a priority queue, even one that is $O(N)$, serializes the algorithm. Steps 1–8 of the following alternative design can each be parallelized. The design involves three stages: identifying flats, identifying basins, and building the hierarchy. This can be done as follows: (1) Cells are assigned flow directions. (2) Cells without flow directions are identified—these are flats. (3) Each cell in the flat performs a disjoint-set merge with all its neighbours of the same elevation using the cells' array indices as their keys. If a cell's neighbour has a flow direction (meaning that the particular cell is on the edge of the flat), the neighbouring cell is added to a queue and a note is made that this flat can drain. (4) At this point, all flats are represented by the index of a single one of their member cells. If a flat cannot drain, this representative cell is also added to the queue. (5) A breadth-first traversal is begun for the cells in the queue and used to apply shortest-path flow directions to all the flat cells. (6) At this point, all flats either drain to the ocean or a single, unique pit cell. (7) The ocean and each pit cell each have a unique label. A breadth- or depth-first traversal can be used to apply this label to every cell flowing into a given pit cell or the ocean, forming basins. (8) Exactly as above, the lowest outlet between each basin is identified and (9) the depression hierarchy is constructed.

Unfortunately, load balancing the parallel traversals can be non-trivial. Therefore, we include preliminary source code for a parallel implementation here, but defer developing a performant algorithm for future work.

6 Applications

Once the hierarchy has been generated, it can be used to rapidly produce a number of outputs of interest. This includes three different methods for DEM preconditioning, such as those used for hydrological calculations: filling depressions, carving depressions, and depression filtering. In addition, this approach can be used to compute depression statistics and to model water flow across a landscape.

6.1 Depression Filling

Depression filling raises the elevation of all cells within a depression to the level of the depression's lowest outlet. This ensures that all cells have a monotonically-descending flow path to the edge of the DEM. Barnes et al. (2014b) review depression-filling algorithms and offer a general algorithm unifying previous work. This has since been accelerated for serial execution (Zhou et al., 2016; Wei et al., 2018) and parallelized for large datasets (Barnes, 2016a).

The depression hierarchy algorithm can be used to perform depression filling by raising each cell c of the DEM to the elevation of its ultimate outlet to the ocean (i.e., the outlets above ~~5, 8~~11, 12, or 15 in Figure 1, or the elevation of meta-depression ~~5~~ 7 in Figure 5). This operation will leave flat areas behind which can be resolved by other algorithms (Barnes et al., 2014a). Alternatively, since the location of the outlet is known, a breadth-first traversal from that point over the depression's cells will yield a drainage surface.

6.2 Depression Carving

Depressions can be removed in $O(N)$ time by carving paths from the pit cells of the depression hierarchy's leaves to the ocean. To do so, the elevation of each depression's pit cell should be noted. Since the location of the depression's outlet is known and every cell has been assigned a flow direction, these flow directions can be followed from the outlet to the pit cell. To remove the depression, the flow directions along this path should be reversed (if they flow away from the ocean) or retained (if they flow towards the ocean). Furthermore, once the reversed path has been built, the original DEM can be altered to enforce drainage by traversing the path from the pit cell to the ocean and decrementing each cell along the way, being careful to use a function similar to C++'s `std::nextafter` to prevent floating-point cancellation. This will produce flow fields similar to those resulting from previous works (Braun and Willett, 2013; Lindsay, 2015). See Figure 6 for an example.

6.3 Filtering Depressions

Depressions can be selectively removed by traversing the depression hierarchy. Typically, small or shallow depressions are considered to be artifacts; these can be identified by checking whether a depression's area or volume falls below a threshold. If so, the depression can be filled to the level of its outlet or breached (Lindsay, 2015) by using a priority-queue seeded with any of the depression's pit cells in a way that is similar to Priority-Flood (Barnes et al., 2014b). See Figure 6 for an example.

6.4 Depression Statistics

The number of cells in a depression, the area the depression covers, and the volume of the depression can all be calculated by adapting the depression-filling method above. To do so, a cell c's elevation is compared with the outlet elevations of the depressions in the hierarchy. The lowest such depression-containing cell c is identified. This depression's *cell count* is then incremented and the cell's areas and elevation are added to the depression's *summed elevation* and *summed area*.

The foregoing process produces marginal values: the areas, volumes, and cell counts associated uniquely with each node in the depression hierarchy. To generate totals, the values of each depression below a given node in the hierarchy must be summed. To do so, the depression hierarchy is traversed in depth-first fashion from its leaf depressions upwards to the ocean. Each depression's *cell count* D_c, *summed elevation* D_e, and *summed area* D_a are then the sum of those cells that belong uniquely to the depression (per the above) and those that belong to the depression's children. If the outlet elevation of the depression is D_o, the volume of the depression is then given by $D_a(D_c \cdot D_o - D_e)$.

6.5 Flow Modeling

When water falls on a landscape, it flows downhill to the pit cells of depressions. Depressions then begin to fill up until they spill over into neighboring depressions. The combined depression then fills until it too spills over. This continues until the water finds an outlet to the sea. The depression hierarchy described here, with its geolinks, has been optimized to model this dynamic process of filling, spilling, and merging~~, as described in an accompanying paper (Barnes et al., 2019).~~ .

7 Empirical Tests

We have implemented the algorithm described above in `C++17` using the Geospatial Data Abstraction Library (GDAL) (GDAL Development Team, 2016) to read and write data. For efficiency we use a radix heap (Akiba, 2015) and an optimized hash table (Popovitch, 2019). There are 981 lines of code of which 51% are or contain comments. The code, along with correctness tests and a makefile, can be acquired from Github (https://github.com/r-barnes/Barnes2019-DepressionHierarchy) or Zenodo (Barnes and Callaghan, 2019).

Tests were run on the Comet machine of the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns et al., 2014). Each node of Comet has 2.5 GHz Intel Xeon E5-2680v3 processors with 24 cores per node and 128 GB of DDR4 RAM. Code was compiled using GNU g++ 7.2.0 with full optimizations enabled. The datasets used and timing results are

Dataset	Dimensions	Cells	Time (s)	Barnes2014 (s)	Zhou2016 (s)	Wei2018 (s)
Madagascar	2000 x 1000	$2.0 \cdot 10^6$	0.2	0.2	0.2	0.1
U.S. Great Basin	1920 x 2400	$4.6 \cdot 10^6$	1.0	1.0	0.9	0.4
Australia	5640 x 4200	$2.3 \cdot 10^7$	2.4	2.0	2.6	1.2
Africa	9480 x 9000	$8.5 \cdot 10^7$	17.7	16.2	11.8	5.6
N&S America	18720 x 17400	$3.2 \cdot 10^8$	47.7	48.3	37.8	19.0
Minnesota 30m topobathy	34742 x 23831	$8.2 \cdot 10^8$	117.3	119.7	101.4	39.4
GEBCO_14 global 30" topobathy	86400 x 43200	$3.7 \cdot 10^9$	1881.5	1879.9	1508.5	629.1

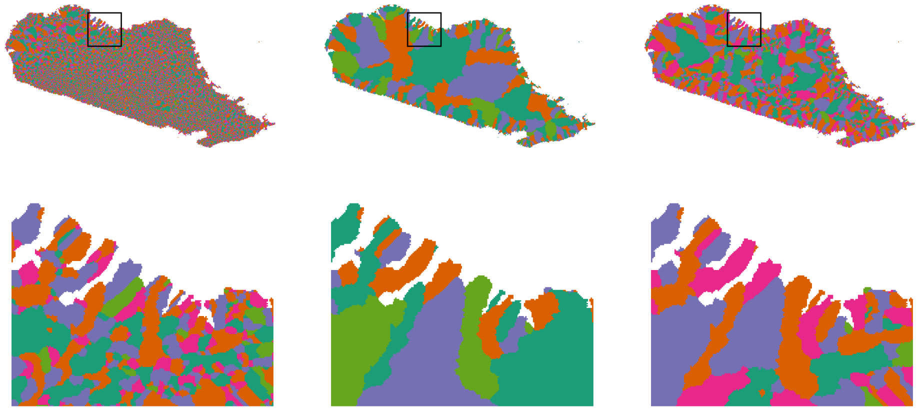
Table 1. Datasets used, their dimensions, and algorithm wall-time on the Comet cluster run by XSEDE (see main text for full specifications). Topographic data for Madagascar, the U.S. Great Basin, Australia, Africa, and North & South America, were clipped from the global GEBCO_08 30-arcsecond global combined topographic and bathymetric elevation data set (GEBCO, 2010). The Minnesota 30m topobathy data is the merged result of two data sources. The topography is resampled from the Minnesota Geospatial Information Office's 1m LiDAR Elevation Dataset (Office, 2019). Bathymetric data were provided by the Minnesota Department of Natural Resources (of Natural Resources, 2014). Richard Lively of the Minnesota Geological Survey merged and combined these data sets. The GEBCO_14 global 30" topobathy data set was drawn directly from GEBCO (2014). Wall-times are compared against several depression-filling algorithms, as described in the text.

shown in Table 1. Datasets were chosen for the large number of depressions they contained. Runtime scales linearly across datasets ranging in size over three orders of magnitude, in agreement with theory. The smaller datasets run quickly enough that they indicate that the depression-hierarchy algorithm may be suitable for use in landscape evolution models.

Wall-times of the depression hierarchy algorithm are compared against RichDEM's (Barnes, 2016b) implementations of several depression-filling algorithms. The structure of the depression hierarchy algorithm is most directly comparable to the improved variant of the Priority-Flood algorithm presented by Barnes et al. (2014b) and exhibits almost no overhead in comparison, showing that constructing the depression hierarchy data structure is inexpensive. Later algorithms from Zhou et al. (2016) and Wei et al. (2018) improve on Priority-Flood by using more complex logic to decrease the number of cells that need to be processed by the priority queue. Incorporating these improvements into the depression hierarchy algorithm would have made it more difficult to describe and verify, so we do not pursue them here.

8 Conclusions

In summary, this paper presents a data structure—the depression hierarchy—that captures the topologic and topographic complexities of depressions in the context of natural landscapes with potential extensions to image processing. The algorithm used to generate this data structure offers advantages in efficiency, correctness, documentation, and reuseability when compared against previous work. ~~An accompanying paper describes~~ A follow-on paper will describe how the depression hierarchy can be leveraged to accelerate hydrological models and rapidly compute the effects of depression structures on drainage networks.



(a) Leaf Labels (b) Top Labels (c) Filtered Labels

Figure 6. Depression hierarchies applied to Madagascar: depression labels. The *Label* array of the depression hierarchy algorithm is shown here for three situations. The top row depicts all of Madagascar while the bottom row depicts the zoomed areas identified by the black boxes. Since there are too many labels to show in distinct colors the labels have instead been colored so that no two adjacent depressions have the same color using a largest-first greedy algorithm (Kosowski and Manuszewski, 2004; Hagberg et al., 2008). **(a)** depicts the labels assigned to the leaf nodes of the depression hierarchy. **(b)** depicts the labels assigned to the uppermost parent depressions—those which connect directly to the ocean. These are the top-level watersheds of the island. **(c)** depicts the labels after depressions less than a given threshold (30 cells in area) are detected by filtering and removed via carving.

Code availability. Complete, well-commented source code, an associated makefile, and correctness tests are available from Github (https://github.com/r-barnes/Barnes2019-DepressionHierarchy) and Zenodo (Barnes and Callaghan, 2019).

Author contributions. KLC and ADW conceived the problem. RB conceived the algorithm and developed initial implementations. KC and RB debugged and tested the algorithm. RB prepared the manuscript with contributions from all authors.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. RB was supported by the Department of Energy's Computational Science Graduate Fellowship (Grant No. DE-FG02-97ER25308) and, through the Berkeley Institute for Data Science's PhD Fellowship, by the Gordon and Betty Moore Foundation (Grant GBMF3834) and by the Alfred P. Sloan Foundation (Grant 2013-10-27).

KLC was supported by the University of Minnesota Department of Earth Sciences Junior F Hayden Fellowship, and by start-up funds awarded to ADW by the University of Minnesota.

Empirical tests and results were performed on XSEDE's Comet supercomputer (Towns et al., 2014), which is supported by the National Science Foundation (Grant No. ACI-1053575). Portability and debugging tests were performed on the Mesabi machine at the Minnesota Supercomputing Institute (MSI) at the University of Minnesota (http://www.msi.umn.edu).

This collaboration resulted from a serendipitous meeting at the Community Surface Dynamics Modeling System (CSDMS) annual meeting, which RB attended on a CSDMS travel grant.

References

Akiba, T.: Software: Radix-Heap, https://github.com/iwiwi/radix-heap Commit f54eba0a19782c67a9779c28263a7ce680995eda, 2015.

Arnold, N.: A new approach for dealing with depressions in digital elevation models when calculating flow accumulation values, Progress in Physical Geography: Earth and Environment, 34, 781–809, https://doi.org/10.1177/0309133310384542, https://doi.org/10.1177/0309133310384542, 2010.

Barnes, R.: Parallel Priority-Flood Depression Filling For Trillion Cell Digital Elevation Models On Desktops Or Clusters, Computers & Geosciences, https://doi.org/10.1016/j.cageo.2016.07.001, 2016a.

Barnes, R.: RichDEM: Terrain Analysis Software, https://doi.org/10.5281/zenodo.1295618, http://github.com/r-barnes/richdem, 2016b.

Barnes, R. and Callaghan, K.: Depression Hierarchy Source Code, https://doi.org/10.5281/zenodo.3238558, 2019.

Barnes, R., Lehman, C., and Mulla, D.: An efficient assignment of drainage direction over flat surfaces in raster digital elevation models, Computers & Geosciences, 62, 128 – 135, https://doi.org/10.1016/j.cageo.2013.01.009, 2014a.

Barnes, R., Lehman, C., and Mulla, D.: Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models, Computers & Geosciences, 62, 117 – 127, https://doi.org/10.1016/j.cageo.2013.04.024, 2014b.

Barnes, R., Callaghan, K., and Wickert, A.: Computing water flow through complex landscapes, part 3: Fill-Merge-Spill: Flow routing in depression hierarchies, In preparation, 2019.

Beucher, S.: Watershed, Hierarchical Segmentation and Waterfall Algorithm, in: Mathematical Morphology and Its Applications to Image Processing, edited by Viergever, M. A., Serra, J., and Soille, P., vol. 2, pp. 69–76, Springer Netherlands, Dordrecht, https://doi.org/10.1007/978-94-011-1040-2_10, 1994.

Blikhars'kyi, Z. Y. and Obukh, Y. V.: Influence of the Mechanical and Corrosion Defects on the Strength of Thermally Hardened Reinforcement of 35GS Steel, Materials Science, 54, 273–278, 2018.

Braun, J. and Willett, S. D.: A very efficient O(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution, Geomorphology, 180-181, 170–179, https://doi.org/10.1016/j.geomorph.2012.10.008, 2013.

Callaghan, K. L. and Wickert, A. D.: Computing water flow through complex landscapes, part 1: Incorporating depressions in flow routing using FlowFill, Earth Surface Dynamics Discussions, 2019, 1–25, https://doi.org/10.5194/esurf-2019-11, https://www.earth-surf-dynam-discuss.net/esurf-2019-11/, 2019.

Calov, R., Beyer, S., Greve, R., Beckmann, J., Willeit, M., Kleiner, T., Rückamp, M., Humbert, A., and Ganopolski, A.: Simulation of the future sea level contribution of Greenland with a new glacial system model, The Cryosphere, 12, 3097–3121, 2018.

Carson, E. C., Rawling III, J. E., Attig, J. W., and Bates, B. R.: Late Cenozoic evolution of the upper Mississippi River, stream piracy, and reorganization of North American Mid-Continent drainage systems, GSA Today, 28, 2018.

Cordonnier, G., Bovy, B., and Braun, J.: A Versatile, Linear Complexity Algorithm for Flow Routing in Topographies with Depressions, Earth Surface Dynamics Discussions, 2018, 1–18, https://doi.org/10.5194/esurf-2018-81, 2018.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C.: Introduction to Algorithms, The MIT Press, 3rd edn.

Costanza, R., Wilson, M., Troy, A., Voinov, A., Liu, S., and D'Agostino, J.: The Value of New Jersey's Ecosystem Services and Natural Capital, Tech. rep., Gund Institute for Ecological Economics, University of Vermont, 2006.

Fenner, T. I. and Loizou, G.: Loop-free Algorithms for Traversing Binary Trees, BIT, 24, 33–44, https://doi.org/10.1007/BF01934513, 1984.

Freeman, T.: Calculating catchment area with divergent flow based on a regular grid, Computers & Geosciences, 17, 413–422, https://doi.org/10.1016/0098-3004(91)90048-I, 1991.

Galler, B. A. and Fischer, M. J.: An improved equivalence algorithm, Communications of the ACM, 7, https://doi.org/10.1145/364099.364331, 1964.

GDAL Development Team: GDAL - Geospatial Data Abstraction Library, Open Source Geospatial Foundation, available at http://www.gdal.org., 2016.

GEBCO: General Bathymetric Chart of the Oceans (GEBCO), GEBCO_08 grid, version 20100927, http://www.gebco.net, 2010.

GEBCO: GEBCO 30 arc-second grid, The GEBCO_2014 Grid, version 20150318, www.gebco.net, 2014.

Giri, S. K., Mellema, G., Dixon, K. L., and Iliev, I. T.: Bubble size statistics during reionization from 21-cm tomography, Monthly Notices of the Royal Astronomical Society, 473, 2949–2964, 2017.

Golovanov, S., Neuromation, O., Kurbanov, R., Artamonov, A., Davydow, A., and Nikolenko, S.: Building Detection from Satellite Imagery Using a Composite Loss Function, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 219–2193, IEEE, 2018.

Grimaldi, S., Nardi, F., Di Benedetto, F., Istanbulluoglu, E., and Bras, R. L.: A physically-based method for removing pits in digital elevation models, Advances in Water Resources, 30, 2151–2158, https://doi.org/10.1016/j.advwatres.2006.11.016, 2007.

Hagberg, A., Swart, P., and S Chult, D.: Exploring network structure, dynamics, and function using NetworkX, in: Proceedings of the 7th Python in Science Conference (SciPy2008), 2008.

Hansen, A. T., Dolph, C. L., Foufoula-Georgiou, E., and Finlay, J. C.: Contribution of wetlands to nitrate removal at the watershed scale, Nature Geoscience, 11, 127–132, https://doi.org/10.1038/s41561-017-0056-6, https://doi.org/10.1038/s41561-017-0056-6, 2018.

Holmgren, P.: Multiple flow direction algorithms for runoff modelling in grid based elevation models: An empirical evaluation, Hydrological Processes, 8, 327–334, https://doi.org/10.1002/hyp.3360080405, 1994.

Humbert, A., Steinhage, D., Helm, V., Beyer, S., and Kleiner, T.: Missing evidence of widespread subglacial lakes at Recovery Glacier, Antarctica, Journal of Geophysical Research: Earth Surface, 123, 2802–2826, 2018.

Iascone, D. M., Li, Y., Sumbul, U., Doron, M., Chen, H., Andreu, V., Goudy, F., Segev, I., Peng, H., and Polleux, F.: Whole-neuron synaptic mapping reveals local balance between excitatory and inhibitory synapse organization, bioRxiv, p. 395384, 2018.

Keeler, B. L., Wood, S. A., Polasky, S., Kling, C., Filstrup, C. T., and Downing, J. A.: Recreational demand for clean water: evidence from geotagged photographs by visitors to lakes, Frontiers in Ecology and the Environment, 13, 76–81, https://doi.org/10.1890/140124, https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/140124, 2015.

Khisha, J., Zerin, N., Choudhury, D., and Rahman, R. M.: Determining Murder Prone Areas Using Modified Watershed Model, in: International Conference on Computational Collective Intelligence, pp. 307–316, Springer, 2017.

Kosowski, A. and Manuszewski, K.: Classical coloring of graphs, Contemporary Mathematics, 352, 1–20, 2004.

Kulbacki, M., Segen, J., and Bak, A.: Analysis, Recognition, and Classification of Biological Membrane Images, in: Transport Across Natural and Modified Biological Membranes and its Implications in Physiology and Therapy, pp. 119–140, Springer, 2017.

Lindsay, J. and Creed, I.: Removal of artifact depressions from digital elevation models: towards a minimum impact approach, Hydrological processes, 19, 3113–3126, https://doi.org/10.1002/hyp.5835, 2005a.

Lindsay, J. B.: Efficient hybrid breaching-filling sink removal methods for flow path enforcement in digital elevation models: Efficient Hybrid Sink Removal Methods for Flow Path Enforcement, Hydrological Processes, https://doi.org/10.1002/hyp.10648, 2015.

Lindsay, J. B. and Creed, I. F.: Removal of artifact depressions from digital elevation models: towards a minimum impact approach, Hydrological Processes, 19, 3113–3126, https://doi.org/10.1002/hyp.5835, https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.5835, 2005b.

Lindsay, J. B. and Creed, I. F.: Sensitivity of digital landscapes to artifact depressions in remotely-sensed DEMs, Photogrammetric Engineering & Remote Sensing, 71, 1029–1036, 2005c.

Mark, D.: Chapter 4: Network models in geomorphology, in: Modelling Geomorphological Systems, edited by Anderson, M., pp. 73–97, 1987.

Martz, L. and Garbrecht, J.: The treatment of flat areas and depressions in automated drainage analysis of raster digital elevation models, Hydrological processes, 12, 843–855, https://doi.org/10.1002/(SICI)1099-1085(199805)12:6<843::AID-HYP658>3.0.CO;2-R, 1998.

Mishra, K., Sinha, R., Jain, V., Nepal, S., and Uddin, K.: Towards the assessment of sediment connectivity in a large Himalayan river basin, Science of The Total Environment, 661, 251 – 265, https://doi.org/https://doi.org/10.1016/j.scitotenv.2019.01.118, http://www.sciencedirect.com/science/article/pii/S0048969719301354, 2019.

Nobre, A. D., Cuartas, L. A., Momo, M. R., Severo, D. L., Pinheiro, A., and Nobre, C. A.: HAND contour: a new proxy predictor of inundation extent, Hydrological Processes, 30, 320–333, 2016.

O'Callaghan, J. and Mark, D.: The extraction of drainage networks from digital elevation data, Computer Vision, Graphics, and Image Processing, 28, 323–344, https://doi.org/10.1016/S0734-189X(84)80011-0, 1984.

of Natural Resources, M. M. D.: Lake Bathymetric Outlines, Contours, Vegetation, and DEM, https://gisdata.mn.gov/dataset/water-lake-bathymetry, 2014.

Office, M. M. G. I.: LiDAR Elevation Data for Minnesota, http://www.mngeo.state.mn.us/chouse/elevation/lidar.html, 2019.

Orlandini, S. and Moretti, G.: Determination of surface flow paths from gridded elevation data, Water Resources Research, 45, https://doi.org/10.1029/2008WR007099, http://dx.doi.org/10.1029/2008WR007099, 2009.

Peckham, S.: Mathematical Surfaces for which Specific and Total Contributing Area can be Computed: Testing Contributing Area Algorithms, Nanjing, China, 2013.

Popovitch, G.: Software: The Parallel Hashmap, https://github.com/greg7mdp/parallel-hashmap Commit 9fa76bd5d7d5d49aedda8b1a8278f0e47425f235, 2019.

Quinn, P., Beven, K., Chevallier, P., and Planchon, O.: The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models, Hydrological Processes, 5, 59–79, https://doi.org/10.1002/hyp.3360050106, 1991.

Salembier, P. and Pardas, M.: Hierarchical morphological segmentation for image sequence coding, IEEE Transactions on Image Processing, 3, 639–651, https://doi.org/10.1109/83.334980, 1994.

Schwanghart, W. and Scherler, D.: Bumps in river profiles: uncertainty assessment and smoothing using quantile regression techniques, Earth Surface Dynamics, 5, 821–839, https://doi.org/10.5194/esurf-5-821-2017, https://www.earth-surf-dynam.net/5/821/2017/, 2017.

Seibert, J. and McGlynn, B.: A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models: A New Triangular Multiple-flow Direction, Water Resources Research, 43, n/a–n/a, https://doi.org/10.1029/2006WR005128, http://doi.wiley.com/10.1029/2006WR005128, 2007.

Shaw, D. A., Vanderkamp, G., Conly, F. M., Pietroniro, A., and Martz, L.: The Fill–Spill Hydrology of Prairie Wetland Complexes during Drought and Deluge, Hydrological Processes, 26, 3147–3156, https://doi.org/10.1002/hyp.8390, https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.8390, 2012.

Shaw, D. A., Pietroniro, A., and Martz, L.: Topographic analysis for the prairie pothole region of Western Canada, Hydrological Processes, 27, 3105–3114, https://doi.org/10.1002/hyp.9409, https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.9409, 2013.

Skiena, S.: The Algorithm Design Manual, Springer, 2nd edn., 2012.

Svoboda, O., Fohlerova, Z., Baiazitova, L., Mlynek, P., Samouylov, K., Provaznik, I., and Hubalek, J.: Transfection by Polyethyleneimine-Coated Magnetic Nanoparticles: Fine-Tuning the Condition for Electrophysiological Experiments, Journal of biomedical nanotechnology, 14, 1505–1514, 2018.

Tarboton, D. G.: A new method for the determination of flow directions and upslope areas in grid digital elevation models, Water Resources Research, 33, 309–319, https://doi.org/10.1029/96WR03137, 1997.

Tarjan, R. E. and van Leeuwen: Worst-case analysis of set union algorithms, Journal of the ACM, 31, https://doi.org/10.1145/62.2160, 1984.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., et al.: XSEDE: accelerating scientific discovery, Computing in Science & Engineering, 16, 62–74, 2014.

Valtera, M. and Schaetzl, R. J.: Pit-mound microrelief in forest soils: Review of implications for water retention and hydrologic modelling, Forest ecology and management, 393, 40–51, 2017.

Wei, H., Zhou, G., and Fu, S.: Efficient Priority-Flood depression filling in raster digital elevation models, International Journal of Digital Earth, 0, 1–13, https://doi.org/10.1080/17538947.2018.1429503, 2018.

Wickert, A. D., Anderson, R. S., Mitrovica, J. X., Naylor, S., and Carson, E. C.: The Mississippi River records glacial-isostatic deformation of North America, Science Advances, 5, https://doi.org/10.1126/sciadv.aav2366, https://advances.sciencemag.org/content/5/1/eaav2366, 2019.

Wu, Q. and Lane, C. R.: Delineation and quantification of wetland depressions in the Prairie Pothole Region of North Dakota, Wetlands, 36, 215–227, 2016.

Wu, Q., Liu, H., Wang, S., Yu, B., Beck, R., and Hinkel, K.: A localized contour tree method for deriving geometric and topological properties of complex surface depressions based on high-resolution topographical data, International Journal of Geographical Information Science, 29, 2041–2060, https://doi.org/10.1080/13658816.2015.1038719, https://doi.org/10.1080/13658816.2015.1038719, 2015.

Wu, Q., Lane, C. R., Wang, L., Vanderhoof, M. K., Christensen, J. R., and Liu, H.: Efficient Delineation of Nested Depression Hierarchy in Digital Elevation Models for Hydrological Analysis Using Level-Set Method, Journal of the American Water Resources Association, 0, https://doi.org/10.1111/1752-1688.12689, 2018.

Zhou, G., Sun, Z., and Fu, S.: An efficient variant of the Priority-Flood algorithm for filling depressions in raster digital elevation models, Computers & Geosciences, 90, Part A, 87 – 96, https://doi.org/http://dx.doi.org/10.1016/j.cageo.2016.02.021, 2016.