RESEARCH ARTICLE
10.1029/2020MS002362

A Rainfall-Based, Sequential Depression-Filling Algorithm and Assessments on a Watershed in Northeastern Indiana, USA

S. A. Noel1, A. C. Ault2, D. R. Buckmaster1, and J. V. Krogmeier2,1

1Department of Agricultural and Biological Engineering, Purdue University, West Lafayette, IN, USA; 2School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

Abstract The landscapes across much of the Midwestern United States are characterized by glacial activity that left numerous water-holding features. Until recently, traditional watershed algorithms assumed these depressions to be errors in the elevation data and filled them as a means of correction. In reality, many of these features may rarely fill and have the potential to dramatically affect surface flow patterns. These depressions play an important role in hydrology, water management, site planning, and agronomy. An improved sequential depression-filling algorithm (SDFA) was developed which fills these water-holding features sequentially based on their respective retention capacity and contributing area. Outputs reflect the state of connectivity following the application of a user-specified amount of rainfall excess (i.e., in excess of infiltration); depressions that have not been filled will remain as their own hydrologically common subcatchments. A performant algorithm is integral to future delivery and utilization by practitioners both in the office and in the field. The optimal set of subroutines were able to fill all of the depressions in a 239 km² watershed in northeastern Indiana in 42 s and 1.5 h on a consumer desktop computer for digital elevation models (DEMs) at 30 m and 3 m resolutions, respectively (O(n²) overall performance).

Plain Language Summary The landscapes across much of the Midwestern United States are characterized by glacial activity that left numerous water-holding features. Until recently, traditional watershed algorithms assumed these depressions to be errors in the elevation data and filled them as a means of correction. In reality, many of these features may rarely fill and have the potential to dramatically affect surface flow patterns. These depressions play an important role in hydrology, water management, site planning, and agronomy. An improved algorithm was developed which fills these water-holding features sequentially based on their respective storage capacity and drainage area. The algorithm's outputs reflect the state of the water-holding feature's connectivity resulting from a given amount of rainfall. This allows depressions to remain which would not ordinarily be filled given this rainfall amount, which differs from existing algorithms. An efficient algorithm is important to future delivery and use by practitioners both in the office and in the field. The algorithm was assessed and able to fill all of the depressions in a 239 km² watershed in northeastern Indiana in 42 s and 1.5 h on a consumer desktop computer for digital elevation models at 30 m and 3 m resolutions, respectively.

1. Introduction

Processes such as watersheds delineation and the computation of surface flow patterns are commonly performed for watershed management tasks of all sorts, from flood risk management to analyzing agricultural crop potential. The algorithms behind these tasks utilize digital elevation models (DEMs) representing the landscape surface, and must address local minima in the DEM so as to permit water to ultimately reach locations to which it is known to contribute. Such local minima in DEMs are called pit or sink cells, and may be the product of noise introduced by the data collection systems (Lindsay & Creed, 2006) or of legitimate natural origins.

Various alternatives to handling sinks and depressions in DEMs, together referred to as hydrologic conditioning, have been developed with flow direction implementations, are what produce differing watershed delineation algorithms (Baker et al., 2006). Hydrologic conditioning typically consists of filling depressions or breaching (lowering elevations of water-retaining cells along the perimeter of depressions) to remove...
depressions. The earliest algorithms filled all depressions; the assumptions behind this decision are suggested by Lindsay and Creed (2006): "(1) the scale and accuracy of a DEM is inadequate to represent actual depressions, which are generally small landforms; (2) depressions rarely occur in natural landscapes and artifact digital depressions are abundant; and (3) actual depressions have minimal impact on hydrogeomorphic processes since they either fill with water and overflow or find subsurface pathways that are closely approximated by surface topography." Early attempts to acknowledge depressional features applied filling and breaching to depressions greater than user-defined thresholds in depth, volume, or area (Lindsay & Creed, 2005; Martz & Garbrecht, 1998, 1999). However, a single threshold applied to all depressions is arbitrary in effect because it fails to represent dynamics of nested depressions and the hydrologic role that these depressions serve in the landscape.

Initially, studies of the role of surface depressions within the landscape focused on the effects of soil roughness on depression storage, overall hydrologic connectivity, and runoff generation (Antoine et al., 2009; Darboux et al., 2002; Huang & Bradford, 1990; Yang & Chu, 2012). More recently, algorithms that simulate the fill-spill behavior of depressions have been developed, gradually filling depressions as they would actually be filled by water in a rainfall event. Some algorithms filled depressions by incrementally applying small amounts of water and adjusting the fill-levels of every depression dynamically (Appels et al., 2011). Shaw et al. (2012) developed a proof-of-concept model called the Simple Pothole Terrain Analysis Algorithm (SPTA) that fills depressions sequentially (one per iteration), resulting in improved theoretical performance over algorithms that incrementally apply water to every depression. Their results, however, were limited to small numbers of depressions and lacked the programmatic implementation details necessary for replication. Chu et al. (2010) developed a Windows-based tool for delineating depressions and several physical parameters, given a DEM. Yang and Chu (2012); Chu et al. (2013) then detailed a progressive puddle-to-puddle model that incorporates filling, spilling, and merging in addition to a conceptual splitting component where infiltration causes depressions to disconnect from one another. Yang and Chu (2015) introduced overland flow and infiltration into their model. Nasab, Singh, and Chu (2017) integrated this model into the popular Soil and Water Assessment Tool (SWAT). Nasab, Zhang, and Chu (2017) recognizes the stepwise patterns to hydrologic connectivity, depression storage, ponding area, and contributing area that results from topographies with depressions in the D-cubed algorithm developed.

A description of the algorithmic complexity will inform the performance of the algorithm for other potential users, as well as the scalability of such algorithms for generating expansive datasets of depression information. Existing studies of algorithm complexity are limited to those that were devised to remove all depressions as opposed to filling them sequentially (Planchon & Darboux, 2001). Barnes et al. (2014) provides a thorough review. The most efficient of these algorithms, referred to as priority-flooding, begin along the perimeter of the DEM and work inward, raising cells as necessary while prioritizing lower-elevation cells for processing first. Priority-flood algorithms are able to achieve an algorithmic efficiency of $O(m \log m)$, where the number of depressions $m$ does not exceed the number of cells in the DEM, $n$.

Further performance gains were made by leveraging the hierarchical nature of depressions nested within one another in a graph structure where nodes represent each depression and edges represent the potential spillover elevations (i.e., a “pass” in common literature) between adjacent depressions. Motivated by wetland cataloging, Wu et al. (2015) devised a novel contour-tree algorithm where nested depressions are identified as enclosed rings contained within other rings from which the hierarchical relationships can then be extracted (Wu & Chu, 2017). After identifying wetlands of interest, they utilized conventional flow routing and watershed delineation techniques, that is, the contour tree method’s minimum unit is the depression and does not simultaneously handle the complexity of raster flow (re)routing. A later iteration improved performance over the vector-based method using a raster-based level-set method (Wu et al., 2019). Furthermore, the work of Wu and Chu (2017) demonstrates a definition of depressions, commonly used where only the “wet surface” is considered, omitting the contributing area above the high water line. In doing this, a global hierarchy of depressions based on the rainfall events required to fill them cannot be produced, precluding rainfall depth-based sequencing of depression filling.

The algorithms of Barnes et al. (2014); Barnes et al. (2020) utilize a wet surface definition of depressions in their development of filling algorithms optimized for the case, where all depressions must be filled, disregarding sequence. Cordonnier et al. (2019) used a minimum spanning tree (MST) approach to find the
A graph representation of the landscape is developed where all adjacent depressions are linked in the graph, and weight by the minimum spillover elevations between pairs of depressions. Two different MST algorithms were then applied, preserving the minimal set of spillover locations and granting connectivity from each depression to the DEM boundaries (i.e., spanning the tree). However, the process which then traverses this graph, merging depressions, and handling the associated raster datasets is not equipped to sequence these merge operations based on rainfall depth. As a result, these algorithms do not generate the rainfall-based depression hierarchy nor can they produce the set of intermediate states of connectivity resulting from a particular rainfall event. By developing a rainfall-based sequence through which depression merge operations should follow, an algorithm will better replicate the natural process of rainfall filling depressions, producing intermediate flow routes and states of overland flow connectivity that better match the progressions observed in the field. This is critical in landscapes where natural depressions exist and do not readily overflow.

This study seeks to extend existing research with the following objectives:

1. Develop an open source sequential depression-filling algorithm (SDFA) capable of simulating the fill-spill flow patterns of surface depressions based on rainfall depth in an automated, repeatable fashion
2. Describe the resulting set of states of connectivity through a graph-structured, rainfall-based depression hierarchy
3. Assess the theoretical performance of the algorithm using Big O notation to describe algorithmic complexity
4. Assess the performance of SDFA on a 239 km² watershed in northeastern Indiana to demonstrate the effect of rainfall excess and depression storage on the resulting contributing area at the watershed outlet

2. Methods

A sequential depression-filling algorithm is key to simulating the dynamics of water retention within depressions and simulating surface flow connectivity that matches real-world observations. The algorithm fills depressions with a user-specified rainfall excess amount, that is, the runoff-ready portion of rainfall that assumes infiltration and other abstractions have already been taken into account; all rainfall excess is applied to the depression-filling process. Figure 1 presents a flow chart of the primary algorithm steps.

2.1. Initialize Flow Direction Grid (Step 1a)

Figure 2. Example grids demonstrating cell indexing and flow direction implementation. (a) Linear cell indexing definition. (b) Example D8 flow direction indicated with arrows. (c) Flow directions expressed as the index of the destination cell, corresponding to (b). For example, cell 1 has a flow direction value of 7, the cell to which it is directed according to (b). Cell 13 is a pit cell, has no flow direction, and is assigned a value of −1.

D-8 flow direction (Jenson & Domingue, 1988) was chosen due to its simplicity, reliability, and ubiquity in common software. While other methods such as D-Infinity may produce improved localized flow routes, this choice has little impact on the overall area contributing to each depression, only how it arrives there. For a particular cell, if no neighboring cells have a lower elevation, including flats, then that cell is marked as a pit cell in the flow direction grid.

An alternate data structure of flow direction under the D-8 algorithm was implemented whereby each cell’s flow direction is defined as the index of the cell to which it flows (its child) rather than the power of two representation used conventionally and made arbitrary by modern programming language capabilities. Figure 2 demonstrates this implementation. This approach enables flow connections between nonadjacent cells,
providing a means to representing real-world structures such as culverts, drainage tiles, and other conduits given the appropriate datasets. By eliminating the conventional lookup table associating neighbors with unique powers of two and instead referencing cell indices directly, traversals of the flow direction data (e.g., watershed delineation) are performed more rapidly.

2.2. Initialize Flow Direction Parents (Step 1b)

The set of cells that contribute flow into downstream cells (i.e., the flow direction parents) were also stored, making the flow direction tree structure doubly linked. While the relationship between parent and child is one-to-one under the D-8 approach, the relationship between child and parents is one to many, necessitating a different data structure from the downstream flow direction matrix. In MATLAB, this was accomplished using a cell array where each element contains a variable-length array of parent cell indexes.

2.3. Initialize Depression Label Matrix (Step 2)

Each pit cell that was found in the flow direction grid is assigned a unique integer value label beginning with one and counting up in ascending order. Specifically, as shown in Figure 3, we define a depression as a single pit cell and the set of cells that flow to that pit cell; this set of cells are assigned the same value in the depression ID matrix. To initialize this matrix, each pit cell must be located and marked, an $O(n)$ operation by itself. This process also returns the array of pit cell indices as output to be used in preceding steps. The depression label matrix is initialized by traversing upstream from each pit cell via flow direction parents. As the traversal takes place, each visited cell is assigned a label in the label matrix while the set of traversed cell indices are accumulated in an array. The array may then be reused in order to avoid unnecessary traversals in future steps of the algorithm.

2.4. Initialize Spillover Elevation/Location (Step 3)

Each depression has a minimum spillover elevation along the perimeter where water will begin to overflow. This step is perhaps algorithmically unintuitive because it is dependent upon not only the cells along the perimeter of the depression in question, but also the cells of neighboring depressions that are immediately adjacent. As demonstrated in Figure 3, cell F is the cell of minimum elevation along the boundary of a depression. However, cells just beyond the perimeter (cells A and G) belonging to the neighboring depressions may be of greater elevation. Such cells serve as barriers and must also be considered. In Figure 3, the depression cannot overflow until filled to the elevation of cell B. This detail affects the spillover location as well as calculations of retention volume, volume to contributing area ratio (VCA), and ordinal position in the filling process.

Perimeter cells are defined as any cell having an adjacent cell of a different depression label within its 3-by-3 neighborhood. Given the set of cells along the perimeter of two depressions $i$ and $j$, the following equation defines the spillover elevation/location:

$$E_{ij} = \min\{\max\{E_i, E_j\}\}$$

where $E_{ij}$ is the spillover elevation (m), and $[E_i, E_j]$ is the array of paired elevations along the perimeter between depressions $i$ and $j$. $E_i$ is the elevation of a perimeter cell that belongs to the depression of interest and $E_j$ is the elevation of a neighboring cell just outside of the perimeter in an adjacent depression.

2.5. Initialize Depression Volumes, Areas, and VCAs (Step 4)

After finding the spillover elevation, the volume of the depression can be computed:
\[ V_d = V_{pf} + \sum_{i=1}^{n} A_i (E_i - E_s), \forall E_i < E_s \]  

(2)

where:

- \( V_d \) = depression volume, m\(^3\)
- \( V_{pf} \) = volume filled in previous iterations, m\(^3\)
- \( n \) = number of cells with elevations less than the spillover elevation
- \( A_i \) = cell area, (DEM resolution)\(^2\), m\(^2\)
- \( E_s \) = spillover elevation, m
- \( E_i \) = cell elevation, m

Only cells with elevations less than the spillover elevation will retain water and are evaluated in Equation 2. The value of \( V_{pf} \) accounts for the volume that was filled in prior iterations of the filling process (further described in Section 2.6). The full set of cells used in Equation 2 are stored in memory and looked up, avoiding any need to traverse the flow direction data to gather them. Next, the contributing area of the depression is given by:

\[ A_c = \sum_{i=1}^{m} A_i \]  

(3)

where:

- \( m \) = number of cells draining to the pit cell, including the pit cell
- \( A_c \) = contributing area, m\(^2\)

Together, the VCA is equivalent to the depth of rainfall excess necessary to fill that particular depression:

\[ VCA = \frac{V_d}{A_c} = P_e \]  

(4)

where

- \( VCA \) = volume to contributing area ratio, m
- \( P_e \) = excess precipitation, m

### 2.6. Iterative Depression Filling (Step 5)

Filling begins as the depression with the minimum VCA overflows, continuing in ascending VCA order until all depressions are filled. The algorithm can also perform filling to a particular rainfall excess value by iterating through depressions until the VCA of the next depression exceeds that particular rainfall excess value. Any depressions that remain will not be filled, they will remain their own hydrologically common watershed and do not contribute to other watersheds.

Two types of depression mergers have been identified based on how the two depressions are oriented. A submerge-type merger (Figure 4a) occurs when the spillover elevation of the resulting composite depression is greater than the spillover elevation identified between the two depressions individually (i.e., the two merging depressions are submerged in the result). The DEM representing the current state of the surface, as it is progressively filled may be referenced to correctly compute volume on each iteration for submerge-type depressions. A cascade-type merger (Figure 4b) occurs where the first depression fills and cascades into the second depression; the new spillover elevation has a lower elevation than one or more depressions that have been previously filled. Because the summation of volume under this new spillover elevation would exclude the volume located above the spillover elevation, a separate sum, \( V_{pf} \) in Equation 2, is necessary and maintained for each depression.

Step 5a begins each iteration by identifying the depression of minimum VCA. Next, Step 5b updates the flow directions of the overflowing pit to reflect the new state of flow connectivity downstream. Flow is rerouted...
from the overflowing depression by iteratively tracing downstream from the spillover location until the pit cell is reached, pointing flow back toward the previous iteratee along the way. By virtue of providing a flow path for the pit cell, every cell in the depression is granted connectivity downstream. Alternatively, an $O(1)$ method assigns the pit cell a flow direction value equal to the spillover index. However, this method does not conform to conventional D-8 and results in discontinuities in the resulting flow routing outputs that can be visually unintuitive.

Step 5c updates the DEM, raising the elevations of the depression and then updating the depression ID matrix such that the overflowing depression takes on the label of the downstream depression to which it now flows. Again, the cached copy of cell indexes is accessed to update the depression ID matrix and DEM using conditionals. In Step 5d, the set of candidate spillover locations are obtained by appending those of the two merging depressions, pruning out spillover locations no longer along the perimeter, and using Equation 1 to then find the new spillover location. For Step 5e, the set of cells comprising the two merging depressions are concatenated, and volume, area, and VCA are computed using the same equations and methods as in Step 4. An optimization for cascading-type mergers (detected when the new spillover location is less than the previous one) where the volumes of the two merging depressions are summed, eliminating the need to visit the set of cells comprising the depressions. This $O(1)$ operation, however, reverts to an upper bound of $O(n)$ (for an $n$-cell depression) in the case of a submerge-type merger.

For Step 5, the algorithmic complexity must consider the number of iterations in addition to the complexity of the step itself. For example, a step that must visit every cell in the DEM on each fill iteration could be described as $O(nd)$, where $n$ is the number of cells in the DEM and $d$ is the number of depressions. At most, Step 5 must compute on the set of cells comprising the merged depression as opposed to all $n$ cells in the DEM. Unfortunately, there exists no analytical description of the depressions in terms of $n$ due to their unpredictable size and shape in a given DEM, but only an upper bound in algorithmic complexity that may occur (though unlikely) when every cell in the DEM is a pit cell. Given a DEM with $n$ cells and therefore $n$ depressions, there will be $n$ total iterations where the first iteration begins with two one-cell depressions merging (visiting $1 + 1$ cells on that iteration). On the next iteration, the two-cell depression will merge with another one-cell depression (visiting $2 + 1$ cells on that iteration), and so forth until no depressions remain. This series can be described as follows:

$$\sum_{i=1}^{n} (i + 1) = \frac{n(n + 1)}{2}$$

where $n$ is the number of cells in the DEM, which reduces to a Big O notation of $O(n^2)$. This result also demonstrates the potential shortcomings in Big O notation comparisons; for example, an algorithm that must visit every cell in the DEM on every iteration would have an equivalent Big O notation despite greater run times.

Figure 5 depicts a set of depressions in a 2-D topography as well as the corresponding depression hierarchy in a graph representation. It should be noted that as a depression overflows, it takes on the label of the depression into which it merges, as denoted in the graph. The appropriate depression merges can be determined by tracing along the edges of the graph until an edge is reached which exceeds a given rainfall.
2.7. Flow Accumulation (Step 6) and Watershed Delineation (Step 7)

Flow accumulation is a challenging computation because it requires global knowledge regarding the other cells flowing toward any particular cell of interest. Flow accumulation is computed using a recursive traversal upstream along flow direction parents. As each recursive call returns, flow accumulation values are incremented and this sum is returned to be used in the next flow direction child. Finally, watershed delineation (Step 7) utilizes the traversal method described in Step 2 on a selected outlet cell to generate a Boolean matrix repressing the watershed consisting of ones for contributing cells and zeros for all remaining cells. Because of the redundancy of this subroutine with respect to others, this step is omitted from performance evaluation.

2.8. Test Watershed and Data Used

A watershed was selected to evaluate the scalability of the model in terms of performance on a broad scale. A subportion of the St. Joseph River Watershed (HUC 0410000306) located in northeastern Indiana was chosen for analysis (St. Joseph River Watershed Initiative, 2005). It has an area of 239 km² and glacial activity played a major role in the development of the soils, topography, and hydrology of the region. The landscapes are flat to rolling and feature a number of potholes and depressions that retain water. It is representative of the glaciated landscapes across much of the Midwestern United States. It is largely undeveloped with agriculture as the primary land use. Figure 6 shows imagery and elevations of the watershed. A watershed of this size was selected for testing in order to (1) demonstrate the scalability of the algorithm in terms of algorithmic performance and (2) evaluate an area of sufficient size that captures the frequency of occurrence of depression features across the landscape (so as to avoid confirmation bias).

DEM of the watershed at 30, 10, and 3 m resolutions were obtained for analysis. The 30 and 10 m DEMs were obtained from the USGS National Elevation Data set (U.S. Geological Survey, 2009). The 3 m DEM was derived from county mosaics of the Indiana statewide vendor-supplied LiDAR 1.5 m resolution DEMs, subsampled to 3 m (Indiana Map Framework Data, 2012). The DEMs were clipped to the extent of the excess amount of interest. Each step along the graph represents a change in the state of connectivity over the DEM surface.
watershed, and cells outside of the watershed were assigned NaN values. The DEMs were processed in MATLAB 2018a running on a Windows 10 desktop machine with an Intel Core i7-4770K processor and 6.7 GB of allocated Java Heap Memory. The MATLAB profiler was used to track performance and outputs of run time, and peak memory usage were recorded. A parallel pool consisting of six workers was configured for parallel processes. Other outputs of the algorithm itself also serve to illustrate the hydrologic role of depressions in the landscapes of this region. The percentage of the total watershed area connected to the outlet was recorded on each iteration for the three DEM resolutions.

Finally, a mass balance was performed to verify the conservation of mass through this simple model, omitting infiltration and other abstractions. The overall mass balance can be described as follows:

\[ V_t = V_s + V_r \]  

where \( V_t \) is the total rainfall excess volume applied, \( V_s \) is the volume stored, and \( V_r \) is the volume that ran off. \( V_t \) can be computed as the rainfall depth applied uniformly over the area of interest, \( A_{\text{AOI}} \).

\[ V_t = P_t(A_{\text{AOI}}) \]  

Next, the filled DEM surface can be compared with the original DEM surface to compute the volume stored in depressions:

\[ V_s = \sum_{i=1}^{n} A_i(E_{i,\text{filled}} - E_{i,\text{original}}) \]  

While depressions are typically ignored once they begin flowing off of the DEM surface, the paired array of contributing area versus rainfall excess was used to compute the runoff volume for the study watershed:

\[ V_r = \sum_{i=1}^{n} A_i(VCA_i - VCA_{i-1}) \]  

where

- \( V_t \) = runoff volume; \( m^3 \)
- \( n \) = number of iterations
- \( A_i \) = contributing area at iteration \( i \); \( m^2 \)
- \( VCA_i \) = volume to contributing area ratio at iteration \( i \), \( m \), and
- \( VCA_{i-1} \) is the initial VCA of the region contributing to the outlet when iterative filling begins.
3. Results and Discussion

3.1. Algorithm Performance Results

Table 1 displays the performance results in terms of run time and peak memory usage associated with each subroutine comprising the algorithm. The high peak memory usage of Step 1 can be attributed to the additional resources needed for parallel processing.

Step 3 was grouped with Step 4 and performed in parallel across depressions, resulting in both reduced run time and increased peak memory usage. Unfortunately, MATLAB does not provide detailed profiling results for code within parfor loops, only summary information.

For Step 5a, MATLAB utilizes an optimized min function that provides performance comparable to that of finding the minimum of a sorted list (finding the minimum of a sorted list was expected to perform as $O(n)$).

Step 5b, in which two options were described, one constrained by the convention of adjacent flow paths and the other not, is shown in Table 1. While the latter is superior due to its $O(1)$ complexity, the former provides more intuitive output, it is compatible with conventional D-8 implementations, and is not a bottleneck in the algorithm.

Step 5d of the algorithm is one of the costliest steps in the algorithm as each potential spillover location must be evaluated. Step 5e, however, has the largest impact on the performance of the algorithm with exception to flow accumulation, which though it has been evaluated, is not considered part of the proposed filling algorithm. While cascading-type mergers are an $O(1)$ operation, the computation of depression volume is much more unpredictable and inefficient for submerge-type mergers. For this reason, the resulting run time and memory consumption can be largely attributed to the submerge-type depression mergers. Finally, Step 6, flow accumulation, serves as a proxy for the similar process of watershed delineation. While this step is the costliest in terms of run time, it is included for reference, not as part of the proposed algorithm.

The performance results for the algorithm run on three different DEM resolutions are summarized in Table 1. Peak memory usage occurs during Steps 3 and 4 where depression parameters are initialized in parallel, resulting in six times the memory demand had it been performed serially (due to the six parallel workers). The algorithm utilizes 3.15 GB of memory at peak on the 3 m DEM, meaning that an even larger watershed could be processed on the hardware used for testing before the constraint becomes memory.

| Step 1a | Initialize flow direction | 0.834 | 21,100 | $O(n)$ |
|--------|--------------------------|-------|--------|--------|
| Step 1b | Initialize flow direction parents | 1.75 | 5,270 | $O(n)$ |
| Step 2 | Initialize depression label matrix | 2.04 | 5,594 | $O(n)$ |
| Step 3 | Initialize spillover location | 1.92 | 32,400 | $O(n)$ |
| Step 4 | Initialize volume, area, and VCA | 0.632 | 256 | $O(n)$ |
| Step 5a | Identify depression iteratee | 0.335 | 40 | $O(p)$ |
| Step 5b | Reroute flow direction | | | |
| Approach a: Direct path update | 0.140 | 5,270 | $O(n^2)$ |
| Approach b: Single cell update | 0.0190 | 5,270 | $O(n)$ |
| Step 5c | Update DEM and depression labels | 0.346 | 5,270 + 3,830 | $O(n^2)$ |
| Step 5d | Identify spillover location | 1.17 | 528 + 10,500 | $O(n^2)$ |
| Step 5e | Compute volume, area, and VCA | 1.60 | 1,960 + 3,830 | $O(n^2)$ |
| Step 6 | Flow accumulation | 4.24 | 5,270 | $O(n)$ |

* $n =$ number of cells in the DEM; ** $p =$ number of depressions.

Table 1

Performance Results for Each Step of the Implemented Algorithm
3.2. Other Algorithm Outputs and Interpretations

The plot in Figure 7, used in tandem with Figure 8, demonstrates the discrepancy between previous assumptions that depressions are artifacts that can be negated and the claim herein that depressions can play an appreciable role in hydrology at a range of scales. These depressions are important to consider in regard to agricultural, drainage, flood control, and other decisions. This plot in Figure 7 takes a stepwise form, where after a threshold amount of rainfall excess, a depression is filled, overflows, and begins to contribute downstream to the watershed outlet. While many depressions are filled as rainfall excess is applied, only those depressions that contribute to the selected watershed outlet point are captured in this plot. After 50 mm of applied rainfall excess, less than 20% of the watershed area is connected to the outlet using the 3 m DEM (compared to 70% for the 30 m DEM). After 150 mm of applied rainfall excess, only 60% of the potential watershed area achieves connectivity to the watershed outlet according to the 10 and 3 m DEMs (80% for the 30 m DEM). Finally, a rainfall excess amount of 600 mm is necessary to fill all depressions and achieve 100% watershed connectivity, exceeding the equivalent of a 100 years, 24 h rainfall event, approximately 140 mm according to Hershfield (1961).

Table 2,3 shows the results of the mass balance performed on the study site, computed from Equation 7 through 9, on the 30 m DEM. This successfully verifies that mass is conserved through this simplified model when considering only rainfall excess applied, water stored in depressions, and runoff that occurs at the watershed outlet while other depressions remain to be filled. The three independently computed volumes satisfy the overall balance using Equation 6. Figure 7 can also be used to visualize the runoff versus storage relationship with increasing rainfall excess. The area below the curve can be integrated to compute runoff (as a percentage of the total watershed area in its current form), while the area above the curve represents

![Figure 7. Percent of the potential total watershed area draining to the outlet after each iteration for the three Digital Elevation Models (DEMs) analyzed.](image-url)

![Figure 8. Hydrologically connected regions as depressions resulting from increasing amounts of applied rainfall excess (as captioned) on the 10 m resolution digital elevation models (DEM).](image-url)
the water stored. Furthermore, the shape of the curve shows the rate at which the watershed transitions from storing water to generating runoff.

Figure 8 shows the set of depressions produced on the 10 m DEM after applying 0, 25, 75, 150, 300, and 500 mm of rainfall excess. Connectivity, observed as contiguous regions of a particular color, gradually grow in spatial extent with increasing rainfall excess. The areas immediately adjacent to rivers and drainage ditches begin to connect in the early stages of the process (beginning with Figure 8b). As filling continues, this region eventually grows into the full watershed Figure 8. Subcatchment boundaries with straight edges corresponding to interstate roads and highways can be identified, for example, Figure 8e; such features erroneously retain water that is likely drained by culvert and bridge features that have not yet been hydrologically corrected in the elevation data. While the nonadjacent flow connections outlined in this study serve to address these sorts of features, the focus of this study is on the sequential depression-filling algorithm.

4. Conclusions, Limitations, and Future Work

An algorithm was developed which routes water across a modeled landscape (using the DEM), simulates retention, and fills depressions sequentially given an input rainfall excess amount. Such an algorithm is useful where elevation data reveals surface depressions that are not artifactual and may dramatically affect surface flow patterns as present in much of the Midwestern United States as well as other glaciated topographies the world over. Requiring only a value for rainfall in excess of infiltration provides a straightforward, intuitive parameterization of the algorithm, and should serve to provide a better understanding of the dynamics of potholes and depressional features on existing hydrologic models. In its current form, the algorithm is limited in terms of its ability to handle spatiotemporally vary rainfall events; while the algorithm can be augmented to better handle this, the complexity, intuitiveness, and performance of the algorithm may be impacted. Future work will focus on the integration of this algorithm into traditional hydrologic modeling frameworks to assess the overall hydrologic role of depressions in these landscapes and within agricultural systems in particular. This also means integrating additional model processes such as infiltration, routing, and other abstractions.

End-to-end, the algorithm produced a total run time of under 1 min for a 30 m DEM and about 1.5 h for a 3 m DEM of a 239 km² watershed (on a consumer desktop machine). Given a machine equipped with additional system hardware resources, DEMs that are several times larger can be processed. Statewide or nationwide datasets may be viable given a week’s time of processing with a small cluster of machines. Assuming a similar frequency of depressions as the study watershed, the land area of the entire state of Indiana (94,322 km²) at 3 m resolution is estimated to take about 5 days to process on a 16 GB machine. Shifting run time toward data initialization prior to filling provides future opportunities to serve the initialized depression parameters, allowing end-use applications to alter flow directions and perform filling to user-specified levels.

Corrugated drainage tiles are buried below the ground surface to drain excess water from heavy, poorly drained soils in much of the Midwestern United States; these tiles also provide an outlet for isolated depressions that produce standing water conditions. Tile inlets, culverts, and other conduits fundamentally alter the depression filling results as well as field-scale flow patterns and should be incorporated to provide accurate flow predictions beyond what can be achieved using only topographic data. Nonadjacent flow connections were implemented and utilized to redirect water after each spillover event, but also provide a means to include the aforementioned drainage features if supplied with this input data set.

Data Availability Statement

Source code and input data to replicate the results found herein are available via the Purdue University Research Repository with DOI 10.4231/S2B6-J628 (Buckmaster & Noel, 2020).
Acknowledgments

Funding: This work was supported by the National Institute of Food and Agriculture grant 2011-5130-31119 and Foundation for Food and Agriculture Research award 534662.

References

Antoine, M., Javoux, M., & Bielders, C. (2009). What indicators can capture runoff-relevant connectivity properties of the micro-topography at the plot scale? *Advances in Water Resources*, 32, 1297–1310. https://doi.org/10.1016/j.advwatres.2009.05.006

Appéls, W. M., Bogaart, P. W., & van der Zee, S. E. A. T. M. (2011). Influence of spatial variations of microtopography and infiltration on surface runoff and field scale hydrological connectivity. *Advances in Water Resources*, 34, 303–313. https://doi.org/10.1016/j.advwatres.2010.12.003

Baker, M. E., Weller, D. E., & Jordan, T. E. (2006). Comparison of automated watershed delineations. *Photogramm Eng Remote Sensing*, 72, 159–168. https://doi.org/10.14358/ paris.72.2.159

Barnes, R., Callaghan, K. L., & Wickert, A. D. (2020). Computing water flow through complex landscapes - Part 2: Finding hierarchies in depressions and morphological segmentations. *Earth Surface Dynamics*, 8, 431–445. https://doi.org/10.5194/esurf-2019-34.10.5194/esurf-8-431-2020

Barnes, R., Lehman, C., & Mulla, D. (2014). 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

Buckmaster, D., & Noel, S. A. (2020). A rainfall-based, sequential depression-filling algorithm and applications on a watershed in northeastern Indiana, USA. Purdue University Research Repository. https://doi.org/10.4231/S2B6-J628

Chu, X., Yang, J., Chi, Y., & Zhang, J. (2013). Dynamic puddle delineation and modeling of puddle-to-puddle filling-spilling-merging-splitting overland flow processes. *Water Resources Research*, 49, 3825–3829. https://doi.org/10.1002/wrcr.20286

Chu, X., Zhang, J., Chi, Y., & Yang, J. (2010). An improved method for watershed delineation and computation of surface depression storage. *Watershed management 2010: Innovations in watershed management under land use and climate change* (pp. 1113–1122). https://doi.org/10.1061/41143(394)100

Cordonnier, G., Bovy, B., & Braun, J. (2019). A versatile, linear complexity algorithm for flow routing in topographies with depressions. *Earth Surface Dynamics*, 7, 549–562. https://doi.org/10.5194/esurf-7-549-2019

Darboux, F., Davy, F., Gascuel-Odoux, C., & Huang, C. (2002). Evolution of soil surface roughness and flowpath connectivity in overland flow experiments. *Catena*, 46, 125–139. https://doi.org/10.1016/s0341-8162(01)00162-8

Hershfield, D. M. (1961). *Rainfall frequency atlas of the United States, technical paper*, 40 (pp. 1–61). Weather Bureau, US Department of Commerce.

Huang, C.-h., & Bradford, J. M. (1990). Portable laser scanner for measuring soil surface roughness. *Soil Science Society of America Journal*, 54, 1402–1406. https://doi.org/10.2136/ssaaj1990.0361599500540005003x

Indiana Map Framework Data. (2012). *Indiana statewide imagery and lidar program*. Retrieved from https://gisdb.uits.indiana.edu/portal/map/m10000.html, 10-10-2020

Jenson, S. K., & Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, 54, 1593–1600.

Lindsay, J. B., & Creed, I. F. (2006). Distinguishing actual and artifact depressions in digital elevation data. *Computers & Geosciences*, 32, 1192–1204. https://doi.org/10.1016/j.cageo.2005.11.002

Martz, L. W., & Garbrecht, J. (1998). 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)12:5<843::aid-hyp658>3.0.co;2-r

Martz, L. W., & Garbrecht, J. (1999). An outlet breaching algorithm for the treatment of closed depressions in a raster DEM. *Computers & Geosciences*, 25, 835–844. https://doi.org/10.1016/S0098-3004(99)00018-7

Nasab, M. T., Singh, V., & Chu, X. (2017a). SWAT modeling for depression-dominated areas: How do depressions manipulate hydrologic modeling? *Water*, 9, 2–16. https://doi.org/10.3390/w9010058

Nasab, M. T., Zhang, J., & Chu, X. (2017b). A new depression-dominated delineation (D-cubed) method for improved watershed modeling. *Hydrological Processes*, 31. https://doi.org/10.1002/hyp.12161

Plancho, O., & Darboux, F. (2001). A fast, simple and versatile algorithm to fill the depressions of digital elevation models. *Catena*, 46, 159–176.

Shaw, D. A., Pietroniro, A., & Martz, L. W. (2012). Topographic analysis for the prairie pothole region of Western Canada. *Hydrological Processes*, 27, 3105–3114. https://doi.org/10.1002/hyp.9409

St. Joseph River Watershed Initiative. (2005). Cedar creek watershed management plan. Retrieved from https://www.in.gov/idem/rps/files/wmp_stjosewri-maumes_02-502.pdf

U.S. Geological Survey. (2009). *National elevation dataset*. Retrieved from http://nationalmap.gov

Wu, Q., & Lane, C. R. (2017). Delineating wetland catchments and modeling hydrologic connectivity using lidar data and aerial imagery. *Hydrology and Earth System Sciences*, 21, 3579–3595. https://doi.org/10.5194/hess-21-3579-2017

Wu, Q., Lane, C. R., Li, X., Zhao, K., Zhou, Y., Clinton, N., et al. (2019). Integrating lidar data and multi-temporal aerial imagery to map wetland inundation dynamics using Google earth engine. *Remote Sensing of Environment*, 228, 1–13. https://doi.org/10.1016/j.rse.2019.04.015

Wu, Q., Liu, H., Wang, S., Yu, B., Beck, R., & Hinckel, K. (2015). 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

Yang, J., & Chu, X. (2012). Effects of DEM resolution on surface depression properties and hydrologic connectivity. *Journal of Hydrologic Engineering*, 18, 1157–1169. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000731

Yang, J., & Chu, X. (2015). A new modeling approach for simulating microtopography-dominated, discontinuous overland flow on infiltrating surfaces. *Advances in Water Resources*, 78. https://doi.org/10.1016/j.advwatres.2015.02.004