USING NESTED INTERVAL TREE TO MODELING SINGLE FLOW ACCUMULATION MODEL FOR CHANGING FLOW DIRECTION PROBLEM

Khuu Minh Canh[1], Tran Van Hoai[1] and Le Trung Chon[1]

[1] Ho Chi Minh City University of Technology, Viet Nam
Email: ltchon@hcmut.edu.vn

ABSTRACT

In single flow accumulation model, flow direction value in flow node both indicates the direction of flow just in single node and roots the accumulation of dependent sub-flows. As the result, changing flow direction not only just modifies unique value but also affects to the accumulating values of serial nodes which having related to downstream. By modeling flow as graph forest, each changing in graph could make a large timing to computing the new graph links, then the computation is required speeding up for specifying all new (sub)-flows, location of nodes in flow and new accumulation of nodes, especially in density input DEM as Lidar. This paper presents the applying nested intervals tree model for store the flow trees for solving the updating flow direction for node problem.

1. INTRODUCTION

Single flow D8 algorithm [1] was firstly introduced by O'Callaghan & Mark in 1984. Till now, many enhancements for this algorithm mainly come from the way to speedup. In the real situation, the flow may carry some substance or bacteria. And the flow will lead to some thread if the accumulation over one threshold. So, with one real number given as threshold value may be the factor for checking the over accumulation nodes. The traditional implementation of D8 model could be used to find out the results. But the next problem is how to reduce the accumulation by changing the direction of sub flows. These actions are mapped into the reality as change the terrain elevation, using pumping as well as keeping flow (remove flow by real wall or building regulator lake). And the graph-based model could be used to calculate the flow accumulation as well as detect the set of candidate nodes which may be changed their flow direction….

As the flow be formed, the database for the flow and detecting its component is considered. The main questions or the requests for the design of the database are how to:

- Easily detecting the behavior of the neighbors of a selected node. For example: checking if the nodes in the same tree, tree-branch.
- Easily making the connection and calculating the accumulation of the set of nodes when having changes in direction of flows.
- Easily for implementing parallelizing the system.

To answer these above questions, in this research, the one kind of tree model for database is chosen. That is the nested interval tree.

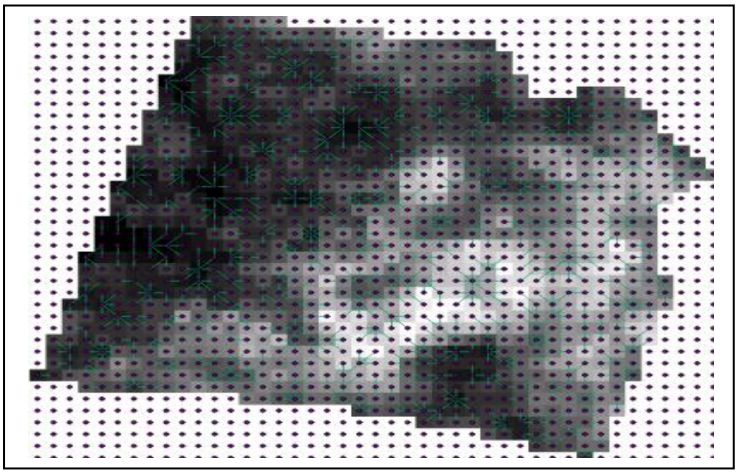

Figure 1. Graph-based flow presentation: forest graph combined by many trees.

2. METHODOLOGY

2.1 Graph-based flow

By building the graph from flow direction then not only the flow accumulation is determined but also the behaviors of graph are inheritance. Especially, when finding the location which changing the elevation to obtaining less than accumulation. With traditional method, explosive combination is occurred. Graph-base may help more information on sub-flow neighbor and boundary of each sub-flow.

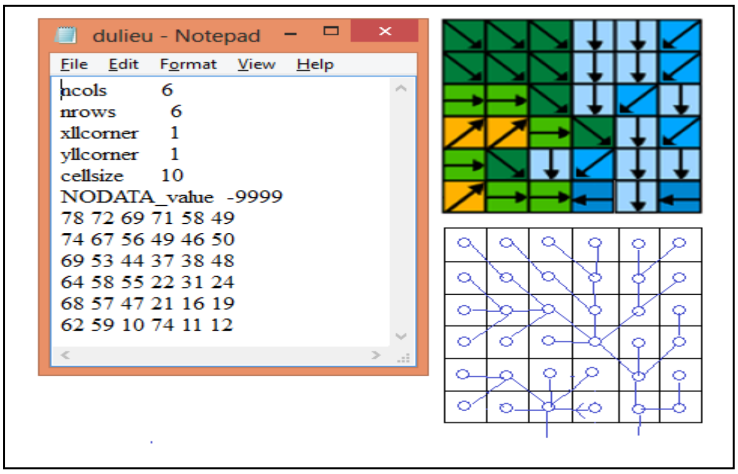

Figure 2. Testing DEM and its traditional flow as well as the graph-based flow

Information supporting in the flow model by graph-based are: set of boundaries points, set of easy flow effect locations ...

2.2 Nested interval tree

Offered by V. Tropashko, nested interval tree is one kind of tree-like structures. Each node on the nested interval tree with four prime integer numbers forming the reducing fraction. The reducing fraction is the result of continued fractions and matrices. This tree does not need traverse through all not for query. Because the relationship between ancestor and descendant is reflected.

For example: the code for node 1.1.2 performed by continued fractions and matrices are below:

$$1.1.2 = \mathbf{1} + \cfrac{1}{\mathbf{1} + \cfrac{1}{\mathbf{2} + \cfrac{1}{x}}} = \frac{5x+2}{3x+1}$$

$$\begin{bmatrix} \mathbf{1} & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 3 & 1 \end{bmatrix} \Leftrightarrow \left(\frac{5}{3}, \frac{2}{1}\right);$$

Using numbers in system of residual classes method has been presented by A. Malikov and A. Turyev (2011). The method avoid using big numbers in intervals (big trees) by arithmetical operations. In this method, the tree nodes quantity is estimated by $\prod p_i$, with p_i are prime numbers.

Table 1. Maximum coded value by prime number

Id	Prime number	Maximum coded value (nodes)
1	2	2
2	2, 3	6
3	2, 3, 5	30
4	2, 3, 5, 7, 11, 13, 17	510,510
5	2, 3, 5, 7, 11, 13, 17, 19	9,699,690

3. PROPOSED ALGORITHM

3.1 Remark from single flow

Properties and advantages/disadvantages of nested interval coding path for single flow are:

- The <.value> range from 1 to 8. The higher values indicate the more flows in the next receivers (flow to). This comment is especially true when we build more a nested interval tree which the root is the sink node.
- The length of paths reveals the flow topology. In real, the statistics of length of path gives the information about the terrain, sub flow trees, … The length of path indicates the depth of the flow tree. That means the length of path is not exceed (n+1), which n is number of the forest nodes.
- The number of nodes of subtrees will be easily calculated by SQL (or string) query.
- Easily check the relation of accumulation when choosing the new flow direction in one node. For example: when changing an node label .1.x.y flow direction into .1.w.z, the part accumulation of ".1" will not be affected (changed).

3.1 Proposed algorithm

As mentioned, designing database to adapt the analyses is the most important. The three main problems are separation/gathering tree branches, search on tree/forest as well as parallel processing. Among treelike models in relational database, nested interval tree [5] is more fitted because of the ease of tracking changing flow. But to build the nested interval tree, the root node must be selected. Despite the flow is directed, the nested interval tree must be undirected. But the tree may be balanced for easily detecting the effect of node having changed the flow direction.

This below is the proposed algorithm for path coding outline.
Input: Undirected graph, a threshold value.
Output: Nested interval coding path start at the center node of graph.
Phase 1: Find the eccentricity nodes of graph (forest):

- Step 1: Reading the graph (forest). Separate the graph into disjointed parts (trees).
- Step 2: For each part, estimate the length of any two vertices in undirected graph. In the set of maximum length from one node, the minimum values indicate the center point of graph (called eccentricity). Prioritize choosing the nodes having more connections among the set of candidate nodes.
- Step 3: Set the path 'pi' for the path of each center node of part pi.

Phase 2: Build nested interval coding path:

- Step 4: For each part pi in graph:
- Step 5: Check the 8 neighbor nodes to set the path (add into candidate list). Check the available connection.
- Step 6: Determine the value add to path (by the existence of the connected neighbor nodes) and remove the node in list when check all its neighbor nodes.
- Step 7: Repeat the step 5 until all of the node of the part $p_i$found.

3.2 Applied nested interval tree in flow problem

The result of building forest from the eccentricity node (for undirected graph of the flow). In this example, the eccentricity node is the node having 22 elevation height locate in 4th row and 4th column:

Table 2. Nested intervals coding: Path

.1.3.1.1	.1.4.1.1	.1.4.2.1	.1.4.2.2	.1.5.1.1	.1.5.1.2
.1.3.2.1	.1.3.1	.1.4.1	.1.4.2	.1.5.1	.1.5.2
.1.3.2.2	.1.3.2	.1.3	.1.4	.1.5	.1.1.2.1
.1.3.2.3	.1.3.3	.1.2	**.1**	.1.1.1	.1.1.2
.2.1.1	.2.1	.2.2	.2.3	.1.1	.1.1.3.1.1
.2.1.2	.2.5	.2	.2.4	.1.1.3	.1.1.3.1

Table 3. Nested intervals coding: Interval

(13/8, 8/5)	(19/12,11/7)	(27/19,17/12)	(46/29,19/12)	(31/20,17/11)	(48/31,17/11)
(18/11,13/8)	(8/5,5/3)	(11/7,8/5)	(19/12, 8/5)	(17/11,14/9)	(31/20, 14/9)
(31/19,13/8)	(13/5, 8/3)	(5/3, 3/2)	(8/3, 5/2)	(14/3, 9/2)	(9/7, 5/4)
(44/27,13/8)	(18/11,5/3)	(11/7,3/2)	(3/2, 2/1)	(5/3, 2/1)	(7/4, 2/1)
(5/2,3/1)	(3/1, 2/1)	(5/2, 2/1)	(7/3, 2/1)	(2/1, 1/1)	(4/3, 3/2)
(8/3, 3/1)	(11/5, 2/1)	(2/1)	(9/4, 2/1)	(1/1)	(3/2, 1/1)

Nested interval tree has some advantages:

- Easily calculate the subtrees and their length (by SQL query).
- Easily replace related branches when updating flow.

3.3 Some implementation codes

Here below some codes in Python to implement

Input: a integer number A and a vector/array of primes pi, A must satisfied the condition A<P=p1p2…

Output: an array of residual number, by alphai=A-[A/pi] pi.

For example: residual (17, [2,3,5]) = [1,2,2]

This below code is written in Python:

```
def residual (number, prime_arr)
    res = []
    for prime in prime_arr:
        res.append(number – (number/prime)*a)
    return res
```

3.4 Database additional consideration

In addition, for the reality using, there are groups of cells having spatial relationship. As on the same roads, the cells have to have the same height values, so the database must point out these relations. When the elevation in one cell is changed, the thresholds of input and output could be considered to detect the changing in flows. Each threshold contains two values max value and min value, so could be named min_threshold_input, max_threshold_input, min_threshold_output, max_threshold_output. The same group cells may be considered as the same level of one branch of abstraction tree.

Besides, the other attributes in each table in database are proposed. Each attribute stands for a step in graph computation. For example: the values of keeping flow in an cell (default value is 0).

All in all, the following attributes are listed below:

- For topology relation part:
  - Set of relation_node_indices values.
  - Set of separation_node_indices values.
  - Value indicating without-changing elevation locations.
  - Value indicating no-flow-adaption, i.e. locations could not for any more flow in.
- For terrain properties part:
  - Set of tree indices (in nested interval tree) values.
  - keeping_flow: double value. Default value: 0.
  - min_input: double value.
  - max_input: double value.
  - min_output: double value.
  - max_output: double value.
- For flows relation part:
  - Set of tree indices (in nested interval undirected/directed tree) values.
  - Value indicating without-pump-bridging locations. Single value, combination of eight directions value (1 xor 2 xor 4 xor 8… xor 128).
  - Eight directions ranking value for next the flow changing direction. Domain is in integer which range [-8. 8].

4. CONCLUSION

The nested interval coding method could help us model the flow for easily solving the changing flow to decrease the accumulation problem. The proposed method using eccentricities of the trees (one kind of graphs) as root. By putting the roots in the eccentricities, the partitions of flow could be easily to analyze. The information while changing in flow direction of nodes could be quickly archived because of the paths. For the large terrains, the system of residual classes method for nested interval coding could be employed to reduce the string manipulating and store.

5. REFERENCES

O'Callaghan & Mark, 1984, D8 algorithm.

C. Wallis, et al., 2009, *Hydrologic Terrain Processing Using Parallel Computing*, 18th World IMACS/MODSIM Congress, Cairns, Australia

D. Hiep-Thuan, L. Sébastien, and M. Emmanuel, 2011, *Parallel Computing Flow Accumulation in Large Digital Elevation Models*, in International Conference on Computational Science, ICCS 2011, ELSEVIER.

Bédard, Y.,1999, *Principles of Spatial Database Analysis and Design*, GIS: Principles, Techniques, Applications & Management, Wiley, Vol. 2nd Ed., No. Chap. 29, 413-424.

A. Malikov and A. Turyev, 2011, *Nested Intervals Tree Encoding with System of Residual Classes*, Special Issue of International Journal of Computer Applications (0975 – 8887) on Electronics, Information and Communication Engineering – ICEICE No. 2

Vadim Tropashko, 2005, *Nested Interval Tree Encoding in SQL*, SIGMOD Record, Vol. 34, No. 2.