Advances in Engineering Software 103 (2017) 38–45

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Analysis and classification of flow-carrying backbones in two-dimensional lattices

Roman Trobec [a,*], Biljana Stamatovic [b]

[a] Jožef Stefan Institute, Jamova 39, Ljubljana 1000 , Slovenia
[b] Faculty of Information Systems and Technologies, University of Donja Gorica, Podgorica 81000, Montenegro

ARTICLE INFO

Article history:
Available online 10 December 2015

Keywords:
Flow
Current
Backbone
Infinite cluster
Percolation
Parallel algorithm
Data-flow

ABSTRACT

The paper proposes a new data-flow based approach for the identification of backbones in infinite clusters on 2-D percolation site lattices of dimension $L \times L$. The infinite cluster is identified first, then a multi step algorithm is applied for the reduction of the infinite cluster to its backbone. Algorithm performances are evaluated theoretically and experimentally. The algorithm is local and can therefore be efficiently implemented on data-flow parallel platforms in $\Theta(L)$ time if applied on percolation lattices near the critical percolation probability or in $\Theta(L^2)$ in the worst case. The proposed methodology could resolve the problem of stack overflow at large systems that can appear with classical graph based algorithms, and has potential for a higher execution speed-up on parallel architectures.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

In the context of percolation theory, the concept of spreading hypothetical fluid particles through a random media [1], e.g. porous materials [2], was introduced in the past. The problem can be formulated as a bond percolation problem or a site percolation problem. In this paper, we deal with the site percolation problem, schematically illustrated by a 2-D lattice of square cells. Finding a spanning cluster, termed also *infinite cluster* (IC), is one of the fundamental tasks in this theory. Such clusters appear if the probability that a percolation *site* is occupied is higher than the critical probability threshold p_c [3]. The *backbone* (BB) can be informally described as parallel pipelines, with no dead ends, that connect two opposite boundaries of the lattice. The BB has an important role in many transport processes through the percolation system, e.g. composite [4] or photovoltaic materials [5], nanoparticles [6], etc. because it enables an accurate evaluation of the non-linear media resistance, permeability or other crucial parameters of various technological processes [7].

Well-known algorithms for the identification of ICs and its parallelization have been presented in [8–10]. Also, several algorithms for BB identification in ICs are known and published in [11–13]. Most of the algorithms for BB identification are recursive, with a risk of stack overflow in large systems. Observing the local characteristics of system elements, e.g. site connectivity, we have explored the system properties and propose a local seven-step algorithm for the determination of the BB.

The algorithm is informally summarized in this paragraph. A lattice of black–white cells is generated, either by a random principle or by a rule. The existence of eventual ICs is confirmed by labeling the white components of the initial lattice, using only the information about the state and connectivity of local cells. If at least one IC exists, the algorithm removes ICs' dead ends using the information about the site connectivity. Then the algorithm locally recognizes so called articulation cells (ACs), that can split an IC into two or more parts, if ACs are removed from the IC. A classification of ACs enables a local identification of the BB based on flooding.

A simple illustrative case is shown in Fig. 1. In its left part, a percolation lattice with dimensions 21×21 cells (sites) is shown. The occupied cells, obtained with the percolation probability $p = 0.6$, are colored black. In the middle, the corresponding IC is shown with black cells. In the right part, the corresponding BB is shown with red cells. The remaining cells are shown in light blue, for better visibility. We see that all dead ends of the BB have been removed. Only one conductive path has been identified. It spans from the top to the bottom lattice boundary and can therefore contribute to a flow or transport of a matter between boundaries.

This paper is based upon Stamatović et al. [14], extended with the following new contributions: (i) the improved algorithm for the identification of a BB in 2-D lattices has been upgraded with two additional steps, to manage all dangling IC parts on boundaries, (ii) the calculation and communication complexities of its data-flow implementation have been evaluated, and (iii) various lattices have been

* Corresponding author. Tel.: +386 1 4773 497.
E-mail address: roman.trobec@ijs.si (R. Trobec).

http://dx.doi.org/10.1016/j.advengsoft.2015.11.002
0965-9978/© 2015 Elsevier Ltd. All rights reserved.

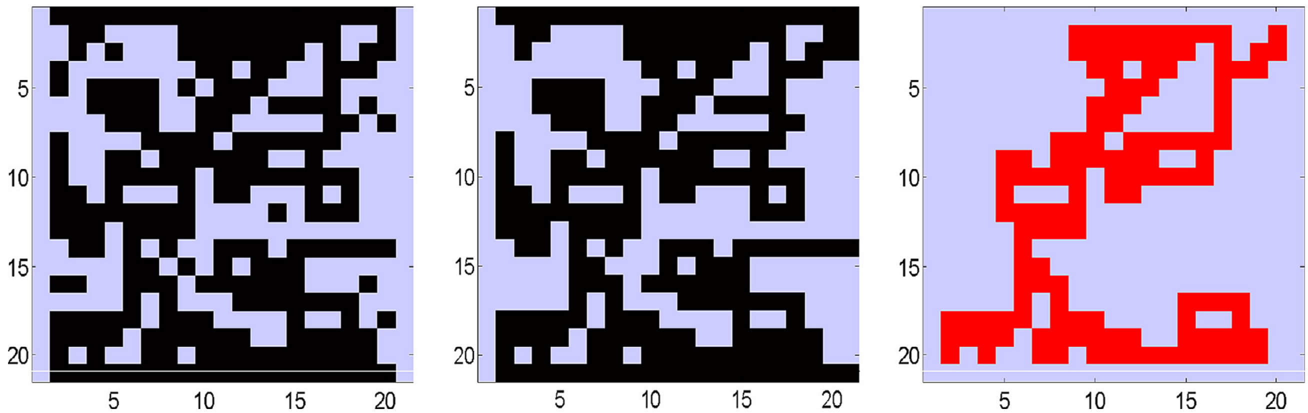

Fig. 1. Examples of a percolation lattice (left), its infinite cluster (middle) and its backbone (right).

experimentally characterized regarding the number of time-steps and the sizes of ICs and BBs.

In the rest of the paper, some essential definitions are given, supported by illustrative examples. A new algorithm for the identification of backbones in an infinite cluster on 2-D percolation lattices with open boundary conditions is proposed. Similar problems, or at least parts of them, have been considered in [8,11,12] but none of the suggested algorithms is entirely local. Our algorithm relies on local properties of the percolation lattice cells and is not limited by the size of the lattice. The proposed solution can resolve the problem of stack overflow that could appear with the classic algorithms for the backbone identification, if implemented on the standard sequential computers. An extensive experimental evaluation of 2-D lattices is done regarding the percolation probability, the size of BBs and the number of algorithmic time-steps. We confirmed that the proposed algorithm has a high potential for a scalable and efficient parallel implementation on data-flow computing platforms, e.g. graphics processing units (GPUs), many-core accelerators (Intel Phi), field programmable gate arrays (FPGAs), systems on chips (SoC), or other processing arrays with high radix interconnection networks [15].

2. Definitions

We consider a 2-D lattice network of unite squares (cells) whose centers are in an integer lattice. For simplicity, we suppose that the lattice has $N = L \times L$ cells $c_{i,j}$ with positions determined by the indices $i, j = 1, \ldots, L$ in x and y directions, respectively, with $L \geq 3$. Each cell can exist in a finite number of states, marked by colors. The cells can change their states at the end of individual time-steps, i.e. at that are discrete moments in time that mark the completion of a computation. The state of all lattice cells in a time-step t is denoted by C_t, $t \geq 0$.

We use the Moore neighborhood with twenty-four neighbors. Each cell has four nearest neighbors (*nn*), four next-nearest neighbors (*nnn*) and sixteen not next-nearest neighbors (*nnnn*). Two cells $c_{i,j}$ and $c_{k,l}$ are *nn*-neighbors if $|i-k| + |j-l| \leq 1$, *nnn*-neighbors if $(i-k)^2 + (j-l)^2 = 2$ and *nnnn*-neighbors if $4 \leq (i-k)^2 + (j-l)^2 \leq 8$. For short notation, the relative positions of cells are denoted by the compass notation, e.g. E for the right *nn*-cell, NE for the right-upper *nnn*-cell, WNN for the left-upper-upper *nnnn*-cell, etc. The *nn-path* (*nnn-path*) is a sequence of black cells p_i, $1 \leq i \leq n$, $n \geq 2$, such that each pair, p_i and p_{i+1}, is *nn*-connected (*nnn*-connected). More definitions about connectivity can be found in [16].

The initial configuration of the percolation sites C_0 can be represented by a 2-D lattice of square cells and each cell can exist in two different states, marked by the white or the black color. The left (W) and the right (E) boundary cells of the lattice are white, while the remaining cells of the top (N) and the bottom (S) boundaries are black. The lattice is surrounded by one additional layer of white cells, to simplify the analysis of cells near boundaries when the *nnnn*-neighborhood has to be examined. This layer is not shown in illustrations of our examples. All the remaining inner cells of the lattice, are colored black with the site probability p, and white with the probability $1 - p$. The probabilities are independent for each cell. A black *nn*-cluster is a group of black cells that are *nn*-connected. Note that the critical probability for such a percolation lattice of *nn*-connected sites is $p_c \approx 0.59$, which is known from the site percolation theory [1]. Site percolation probabilities $p < p_c$ should not result in an IC.

The *infinite cluster* (IC) is a large black *nn*-cluster that spans from the top to the bottom boundary of the above defined lattice. The neighboring cells of *nn*-clusters are connected by edges termed also as *bonds*. In the same way, an *nnn*-cluster is a group of cells in the same state that are *nn*- or *nnn*-connected. In particular, a white *nnn*-cluster is a group of white cells that are *nn*- or *nnn*-connected.

The *backbone* (BB) is a set of black cells from the interior of the lattice that are *nn*-connected with the top and the bottom boundary by two *nn*-paths that have no bonds in common [18]. The BB is a subset of cells from ICs that are *nn*-connected and spans from the top to the bottom boundary of the percolation lattice. Some cells that are members of the IC are dangling, i.e. *nn*-connected with the IC through just a single cell. Consequently, they cannot be members of the BB. The shapes of such dangling parts can be single cells, branches or loops of cells, termed in our paper as *dangling ends* or *dangling loops*.

The BB has an important role in the transport processes of percolation systems. For example, if we study a fluid flow across the bonds of a lattice that connect black cells, we will find out that some of the bonds experience the flow, while the others are dangling ends or loops with no flow. Another example of the applicability of BB identification is in conductivity studies of composite materials made from resistive nano particles, where the BB could be defined as a part of the cluster that can carry electric current if the top and the bottom lattice boundaries are on different electric potentials.

The *articulation cell* (AC) is a black cell of a black IC if its removal (changing its state to white) splits the existing IC into two or more parts, with at least one part becoming unconnected with, or isolated from, the top and the bottom boundaries. The key for finding the BB is to find its ACs. However, local rules, limited to the *nnnn*-neighborhood, should be applied for the data-flow determination of ACs' relations to the BB. Some specific cases of ACs are illustrated and informally explained in the left part of Fig. 2 on an example of the IC from Fig. 1. In the right part of Fig. 2, the identified BB is colored red.

The lattice shown in Fig. 2, with 21×21 cells, incorporates the top and the bottom boundaries, a single black IC (*nn*-cluster), thirteen white *nnn*-clusters O_i, $i = 1, 2, \ldots, 13$, and thirteen ACs a_j, $j = 1, 2, \ldots, 13$, of which two are dangling ends a_1 and a_2. We know that a removal of an AC can isolate a part of IC from the top and the

Download English Version:

https://daneshyari.com/en/article/4977994

Download Persian Version:

https://daneshyari.com/article/4977994

Daneshyari.com