Distributed Algorithm For Shortest Path Problem In

MIDDLETON PHELPS

Incremental Computation of Topological Properties in Distributed Networks Morgan & Claypool Publishers
Networks constitute the backbone of complex systems, from the human brain to computer communications, transport infrastructures to online social systems and metabolic reactions to financial markets. Characterising their structure improves our understanding of the physical, biological, economic and social phenomena that shape our world. Rigorous and thorough, this textbook presents a detailed overview of the new theory and methods of network science. Covering algorithms for graph exploration, node ranking and network generation, among others, the book allows students to experiment with network models and real-world data sets, providing them with a deep understanding of the basics of network theory and its practical applications. Systems of growing complexity are examined in detail, challenging students to increase their level of skill. An engaging presentation of the important principles of network science makes this the perfect reference for researchers and undergraduate and graduate students in physics, mathematics, engineering, biology, neuroscience and the social sciences.
A Minimum-hop Path Failsafe and Loop-free Distributed Algorithm Springer
This paper presents two distributed algorithms for finding shortest paths from a source node to all other nodes in an N-node network. These algorithms are executed at individual nodes using only local information. Algorithm 1 works in networks where there are no topological changes such as link failures, link recoveries or changes of link lengths. Algorithm 2 is a mofification of Algorithm 1 for networks where there are topological changes. Algorithm 1 determines the optimal shortest paths in at most N3/4 steps, which is only one-half of the computational upper bounds of Abram and Rhodes' and Segall, Merlin and Gallager's algorithms. After the last topological change, Algorithm 2 determines the optimal shortest paths in the same number of steps as Algorithm 1. There are many situations where the present algorithms will work up to N/2 times faster than the algorithms proposed by these authors. (Author).
Distributed Algorithms Cambridge University Press
The algebraic path problem is a generalization of the shortest path problem in graphs. Various instances of this abstract problem have appeared in the literature, and similar solutions have been independently discovered and rediscovered. The repeated appearance of a problem is evidence of its relevance. This book aims to help current and future researchers add this powerful tool to their arsenal, so that they can easily identify and use it in their own work. Path problems in networks can be conceptually divided into two parts: A distillation of the extensive theory behind the algebraic path problem, and an exposition of a broad range of applications. First of all, the shortest path problem is presented so as to fix terminology and concepts: existence and uniqueness of solutions, robustness to parameter changes, and centralized and distributed computation algorithms. Then, these concepts are generalized to the algebraic context of semirings. Methods for creating new semirings, useful for modeling new problems, are provided. A large part of the book is then devoted to numerous applications of the algebraic path problem, ranging from mobile network routing to BGP routing to social networks. These applications show what kind of problems

can be modeled as algebraic path problems; they also serve as examples on how to go about modeling new problems. This monograph will be useful to network researchers, engineers, and graduate students. It can be used either as an introduction to the topic, or as a quick reference to the theoretical facts, algorithms, and application examples. The theoretical background assumed for the reader is that of a graduate or advanced undergraduate student in computer science or engineering. Some familiarity with algebra and algorithms is helpful, but not necessary. Algebra, in particular, is used as a convenient and concise language to describe problems that are essentially combinatorial. Table of Contents: Classical Shortest Path / The Algebraic Path Problem / Properties and Computation of Solutions / Applications / Related Areas / List of Semirings and Applications

Distributed Computing McGill-Queen's Press - MQUP

Distributed algorithms have been the subject of intense development over the last twenty years. The second edition of this successful textbook provides an up-to-date introduction both to the topic, and to the theory behind the algorithms. The clear presentation makes the book suitable for advanced undergraduate or graduate courses, whilst the coverage is sufficiently deep to make it useful for practising engineers and researchers. The author concentrates on algorithms for the point-to-point message passing model, and includes algorithms for the implementation of computer communication networks. Other key areas discussed are algorithms for the control of distributed applications (wave, broadcast, election, termination detection, randomized algorithms for anonymous networks, snapshots, deadlock detection, synchronous systems), and fault-tolerance achievable by distributed algorithms. The two new chapters on sense of direction and failure detectors are state-of-the-art and will provide an entry to research in these still-developing topics.

Shortest Path Solvers. From Software to Wetware Cambridge University Press

This text is based on a simple and fully reactive computational model that allows for intuitive comprehension and logical designs. The principles and techniques presented can be applied to any distributed computing environment (e.g., distributed systems, communication networks, data networks, grid networks, internet, etc.). The text provides a wealth of unique material for learning how to design algorithms and protocols perform tasks efficiently in a distributed computing environment.

Complex Networks Springer

Many applications in different domains need to calculate the shortest-path between two points in a graph. In this paper we describe this shortest path problem in detail, starting with the classic Dijkstra's algorithm and moving to more advanced solutions that are currently applied to road network routing, including the use of heuristics and precomputation techniques. Since several of these improvements involve subtle changes to the search space, it may be difficult to appreciate their benefits in terms of time or space requirements. To make methods more comprehensive and to facilitate their comparison, this book presents a single case study that serves as a common benchmark. The paper also compares the search spaces explored by the methods described, both from a quantitative and qualitative point of view, and including an analysis of the number of reached and settled nodes by different methods for a particular topology.

On the Average Communication Complexity of Asynchronous Distributed Algorithms Springer

This book offers advanced parallel and distributed algorithms and experimental laboratory prototypes of unconventional shortest path solvers. In addition, it presents novel and unique algorithms of solving shortest problems in massively parallel cellular automaton machines. The shortest path problem is a fundamental and classical problem in graph theory and computer science and is frequently applied in the contexts of transport and logistics, telecommunication networks, virtual reality and gaming, geometry, and social networks analysis. Software implementations include distance-vector algorithms for distributed path computation in dynamics networks, parallel solutions of the constrained shortest path problem, and application of the shortest path solutions in gathering robotic swarms. Massively parallel algorithms utilise cellular automata, where a shortest path is computed either via matrix multiplication in automaton arrays, or via the representation of data graphs in automaton lattices and using the propagation of wave-like patterns. Unconventional shortest path solvers are presented in computer models of foraging behaviour and protoplasmic network optimisation by the slime mould Physarum polycephalum and fluidic devices, while experimental laboratory prototypes of path solvers using chemical media, flows and droplets, and electrical current are also highlighted. The book will be a pleasure to explore for readers from all walks of life, from undergraduate students to university professors, from mathematicians, computers scientists and engineers to chemists and biologists.

Guide to Graph Algorithms Cambridge University Press

We present a distributed protocol for obtaining the shortest paths between all pairs of nodes in a network with weighted links. The protocol is based on an extension of the Dijkstra (centralized) shortest path algorithm and uses collaboration between neighboring nodes to transfer the information needed at the nodes for the successive construction of the shortest paths. A formal description of the protocol is given by indicating the exact algorithm performed by each node. The validation proofs are greatly simplified by separating the communication mechanism from the computation at the nodes, the latter being the transposition of the Dijkstra shortest path algorithm to the decentralized protocol. (Author).

Introduction to Distributed Algorithms MIT Press

This volume contains papers presented at the First International Workshop on Distributed Algorithms. The papers present solutions to a wide spectrum of problems (leader election, resource allocation, routing, etc.) and focus on a variety of issues that influence communications complexity.

Distributed Algorithms Springer Science & Business Media

This monograph presents the outcome of a GI-Dagstuhl Seminar held in Dagstuhl Castle in November 2005. It gives a first overview of algorithmic results on wireless ad hoc and sensor networks. Many chapters deal with distributed algorithms. Importance is attached to topics that combine both interesting aspects of wireless networks and attractive algorithmic methods. Each chapter provides a survey of some part of the field, while selected results are described in more detail.

Algorithms for Dynamic and Distributed Networks Springer Nature

Abstract: "In an execution of a distributed program, processes communicate among themselves by exchanging messages. The execution speed of the program could be expedited by a faster message delivery system, transmitting messages to their destinations through their respective shortest paths. Some distributed algorithms have been proposed in recent years for determining all pairs shortest paths for an arbitrary computer network. The best known algorithm uses O(n2 log n) messages, where n is the number of computers in the network. This paper presents a new distributed algorithm for the same problem using 2n2 messages in the worst case. This algorithm uses a strategy quite different from those of the other algorithms for the same problem."

A Distributed Shortest Path Protocol MIT Press

One routing strategy frequently used in computer networks assigns traffic dependent distances to the links of the network and then increases the traffic flow on shortest paths. If a central facility monitors all network traffic, classical algorithms can be readily employed to compute shortest paths. If traffic is only locally monitored, we wish to have distributed procedures in which the nodes begin with only local information and compute shortest paths by communicating with one another. We present several such distributed shortest path algorithms and analyze their communication cost. Since the transmission of control information required for network operation reduces the bandwidth available to users, we concentrate on finding algorithms that use a minimum of information exchange. (Author).

Distributed Decomposition of P-plane Networks, Single-source Shortest Paths, and Adaptive Message Routing LibreDigital

The problem of routing in a data network is often treated by assigning traffic dependent lengths to the links of the network and routing traffic from node i to node j along the shortest path from i to j. A distributed algorithm is presented in which the nodes cooperate to find all shortest paths. It runs asynchronously in every node and does not require the network topology, or even the number of nodes in the network, to be known a priori by the nodes.

Highly Parallel Distributed Algorithm for Shortest Path Routing in Dynamically Reconfigurable Networks

This volume contains the proceedings of the fifth International Workshop on Distributed Algorithms (WDAG '91) held in Delphi, Greece, in October 1991. The workshop provided a forum for researchers and others interested in distributed algorithms, communication networks, and decentralized systems. The aim was to present recent research results, explore directions for future research, and identify common fundamental techniques that serve as building blocks in many distributed algorithms. The volume contains 23 papers selected by the Program Committee from about fifty extended abstracts on the basis of perceived originality and quality and on thematic appropriateness and topical balance. The workshop was organizedby the Computer Technology Institute of Patras University, Greece.

A Comparison of Two Distributed Shortest Path Algorithms Based on Dijkstra's Algorithm

Computation of shortest paths is one of the classical problems in theoretical computer science. Given a pair of nodes s and t in a graph G, the goal is to find a path of minimum weight from s to t. Most graphs that commonly occur in practice are sparse graphs. In this work, we deal with several computational problems related to shortest paths in sparse graphs and we present algorithms that provide significant improvements in performance in both sequential and distributed settings. We also present fine-grained reductions that establish fine-grained hardness for several problems related to shortest paths. In the sequential context, we consider the fine-grained complexity of sparse graph problems whose time complexities have stayed at Õ(mn) over the past several decades, where m is the number of edges and n is the number of vertices in the input graph. All of these problems are known to be subcubic equivalent and this shows that achieving sub-mn running time is hard, but only for dense graphs where $m = [Theta] (n2). We introduce the notion of a sparse reduction which preserves the sparsity of graphs, and we present near linear-time sparse reductions between various pairs of graph problems in the Õ(mn) class. We also introduce the MWC-hardness conjecture, which states that Minimum Weight Cycle problem cannot be solved in sub-mn time. We establish that several important graph problems in the Õ(mn) class such as APSP, second simple shortest path (2-SiSP), Radius, and Betweenness Centrality are MWC-Hard, establishing sub-mn fine-grained hardness for these problems. A well-known generalization of the shortest path problem is the k-simple shortest paths (k-SiSP) problem, where we want to find k simple paths from s to t in a non-decreasing order of their weight. In this thesis we present a new approach for computing all pairs k simple shortest paths (k-APSiSP), which is based on forming suitable path extensions to find simple shortest paths; this method is different from the 'detour finding' technique used in all prior work on computing multiple simple shortest paths, replacement paths, and distance sensitivity oracles. The Õ(mn) time bound of our 2-APSiSP algorithm matches the fine-grained time complexity for the simpler 2-SiSP problem, which is the single source-sink version of this problem. Computing APSP is one of the most fundamental problems in distributed computing. We present a simple Õ(n [superscript 3/2]) rounds deterministic algorithm for computing APSP in the well-known CONGEST model which is the first Õ(n2) round deterministic algorithm for this problem. We then improve this further by reducing the round complexity to Õ(n [superscript 4/3]). We also present a faster algorithm for graphs with moderate integer edge weights. We develop several derandomization techniques for our deterministic APSP algorithms. These include efficient deterministic distributed algorithms for computing a small blocker set, which is a set that intersects a desired collection of shortest paths, and several deterministic pipelined approaches for computing the shortest path distance values as well as for propagating the messages in the network. Aside from our deterministic results, all non-trivial distributed algorithms currently known for computing APSP are randomized

Design and Analysis of Distributed Algorithms

Abstract: "Designers of distributed algorithms must deal with a variety of issues including sequential algorithms design, communication protocols, fault tolerance. The distributed design must also include a proof step of the whole algorithm features. This paper gives a new scheme for the design of distributed algorithms. In this approach the design step is performed simultaneously with the proof step. Our distributed design method is mainly based upon parallel recursive schemes, but recursivity is used in a distributed environment so we use two existing and widely available tools: remote procedure call, and the PAR instruction parallel execution of threads."

Decentralized shortest path algorithms in distributed systems

Abstract: "Efficient distributed algorithms are presented for three closely-related problems on asynchronous p-plane networks, i.e., plane networks in which a set of p> 1 faces cover all the nodes. An optimal algorithm is given for decomposing a p-plane network into outerplane networks. The algorithm uses O(n) messages and time for an n-node network. The decomposition is used with an efficient single-source shortest path algorithm for outerplane networks to design a single-source algorithm for p-plane networks, which uses O(pn) messages and time. The latter algorithm and certain other properties of p-plane networks are then incorporated in the design of a communication-, time-, and space-efficient message routing scheme which adapts to changing link conditions and still routes along near-shortest paths."

Rapport

In this thesis we study the problem of computing Betweenness Centrality in dynamic and distributed networks. Betweenness Centrality (BC) is a well-known measure for the relative importance of a node in a social network. It is widely used in applications such as understanding lethality in biological networks, identifying key actors in terrorist networks, supply chain management processes and more. The necessity of computing BC in large networks, especially when they quickly change their topology over time, motivates the study of dynamic algorithms that can perform faster than static ones. Moreover, the current techniques for computing BC requires a deeper understanding of a classic problem in computer science: computing all pairs all shortest paths (APASP) in a graph. One of the main contributions of this thesis is a collection of dynamic algorithms for computing APASP and BC scores which are provably faster than static algorithms for several classes of graphs. We use n = |V| and m = |E| to indicate respectively the number of nodes and edges in a directed positively weighted graph G = (V,E). Our bounds depend on the parameter [nu]* that is defined as the maximum number of edges that lie on shortest paths through any single vertex. The main results in the first part of this thesis are listed below. - A decrease-only algorithm for computing BC and APASP running in time O([nu]* n) that is provably faster than recomputing from scratch in sparse graphs. - An increase-only algorithm for computing BC and APASP that runs in O([nu]*2 log n) per update for a sequence of at least [Omega](m*/[nu]*) updates. Here m* is the number of edges in G that lie on shortest paths. This algorithm uses O(m* [nu]*) space. - An increase-only algorithm for computing BC and APASP that runs in O([nu]*2 log n) but improves the computational space to O(m*n). - A fully dynamic algorithm for computing BC and APASP that runs in O([nu]*2 log3 n) amortized time per update for a sequence of at least [Omega](n) updates. - A refinement of our fully dynamic algorithm that improves the amortized running time to O([nu]*2 log2 n), saving a logarithmic factor. In the second part of this thesis, we study the case when the input graph is a distributed network of machines and the BC score of each machine, considering its location within the network topology, needs to be computed. In this scenario, each node in the input graph is a self-contained machine with limited knowledge of the network and communication power. Each machine only knows the (virtual) location of the neighbors machines (adjacent nodes in the input graph). The messages, exchanged in each round between machines, cannot exceed a bounded size of at most O(log n) bits. In this distributed model, called CONGEST, we present algorithms for computing BC in near-optimal time for unweighted networks, and some classes of weighted networks. Specifically, our main results are: - A distributed BC algorithm for unweighted undirected graphs completing in at most min(2n+O(D[underscore u]); 4n) rounds, where D[underscore u] is the diameter of the undirected network. - A distributed BC algorithm for unweighted directed graphs completing in at most min(2n + O(D); 4n) rounds, where D is the diameter of the directed network. - A distributed APSP algorithm for unweighted directed graphs completing in at most min(n + O(D); 2n) rounds. - A distributed BC algorithm for weighted directed acyclic graphs (dag) completing in at most 2n + O(L) rounds, where L is the longest length of a path in the dag. - A distributed APSP algorithm for weighted dags completing in at most n + O(L) rounds.

An 'all Pairs Shortest Paths' Distributed Algorithm Using 2n2 Messages

This clearly structured textbook/reference presents a detailed and comprehensive review of the fundamental principles of sequential graph algorithms, approaches for NP-hard graph problems, and approximation algorithms and heuristics for such problems. The work also provides a comparative analysis of sequential, parallel and distributed graph algorithms – including algorithms for big data – and an investigation into the conversion principles between the three algorithmic methods. Topics and features: presents a comprehensive analysis of sequential graph algorithms; offers a unifying view by examining the same graph problem from each of the three paradigms of sequential, parallel and distributed algorithms; describes methods for the conversion between sequential, parallel and distributed graph algorithms; surveys methods for the analysis of large graphs and complex network applications; includes full implementation details for the problems presented throughout the text; provides additional supporting material at an accompanying website. This practical guide to the design and analysis of graph algorithms is ideal for advanced and graduate students of computer science, electrical and electronic engineering, and bioinformatics. The material covered will also be of value to any researcher familiar with the basics of discrete mathematics, graph theory and algorithms.

Systematic Building of a Distributed Recursive Algorithm. Example : the Shortest Path Algorithm

The thesis consists of four chapters. Chapter one lays out the foundations of distributed computing. Chapter two presents new decentralized algorithms for sorting files of integers stored in a distributed and asynchronous network. An optimal communication complexity O(n log$\sb2$d) is obtained under the assumption that extra memory is available at each site in an almost balanced binary tree network, where n is the file size and d is the number of nodes in the network. When extra memory is removed from the nodes, our algorithm still manages to have a very good complexity of O(n log$\sb2\sp2$d). Chapter three presents a new decentralized algorithm in shortest path finding in a distributed and synchronous network, in particular we focus on the termination detection problem. The analysis of this algorithm shows an improved complexity over previous methods. Moreover, in chapters two and three we discuss some tradeoffs in the design of these two algorithms. Finally in chapter four we summarize our results and present some unsolved problems arising from this work. (Abstract shortened by UMI.).