Research Statement

Anxiao (Andrew) Jiang

Email: jax@paradise.caltech.edu

1 General field of research

My research interest is in the general field of **information networks**. My study and research are in the areas of *algorithms*, *combinatorial and convex optimization*, *distributed systems* and *information theory*. So far my research has focused on two fields — **file storage in networks**, and **wireless ad hoc communication and sensor networks**. I plan to use my research experience and knowledge to explore broader aspects of information networks, including overlay storage/distribution networks, sensor networks and many other forms, all essential for pervasive computing.

Two key components shared by different kinds of information networks are data storage/sharing and network structure design/utilization. The first component, data storage/sharing, requires optimized placement of data for efficient access, even when the users of the data are extensively distributed, mobile or have very different communication and computing capabilities. Information theory can be applied to help both the storage and the retrieval of data to achieve an optimal performance/redundancy tradeoff. Examples include the storage of shared files in networks using erasure codes for high availability, rate allocation for nodes collecting data in sensor networks, fractionally cascading of information for fast data detection and locating, multicast based on Network Coding, etc.

The second component, network structure design/utilization, is on the design of real- or overlay-network structures that enable efficient storage, locating and transmission of data. In this area, combinatorial designs can create network structures that enable effective data lookup and guarantee data delivery. Computational geometry and combinatorial techniques can extract as well as compress the metric information for a set of nodes (such as the pairwise distances for a set of peer-to-peer nodes dispersed in the Internet, or the locations of nodes in a wireless network), which can help design network structures that are more efficient for data storage and transmission. And when the network nodes exhibit mobility or dynamic membership, Kinetic Data Structures can be used to adaptively adjust network structures at the time of critical events.

It is no exaggeration to say that the rising of information networks, with their extensive applications, will fundamentally change our computing world. For the different kinds of information networks, their common key components and their ad-hoc/overlay nature often enable them to be studied in unified ways. Through my research in *File Storage in Networks* and *Wireless Ad Hoc Communication and Sensor Networks*, I have developed a solid background for the two key components of information networks. Together with my study of algorithms, systems and information theory, it prepares me for a comprehensive study of information networks.

2 Overview of past research

2.1 File storage in networks

Network systems use a two-level storage strategy for shared files — *stable placement* and *caching*. With stable placement, file replicas are stored at optimized positions, which are made publicly known. Those file replicas become the targets of data access. The files are reallocated (called Data Movement) only when necessary. With caching, extra memories are used for storing currently hottest files. Caches refresh their contents constantly, and serve as an auxiliary mechanism for data storage.

My research on Stable Placement of files focused on the fresh topic of *Network File Storage Using Error-Correcting Codes (ECC)*. This topic generalizes the traditional file placement approach. Traditionally, a popular file is replicated in its original form, and the replicas are distributively stored for better performance. A more general approach is to encode the file into a codeword, and distributively store replicas of the codeword components in the network. Every user can get the file by retrieving a subset of the codeword components and decoding them — namely, we use the erasure-correcting capability of ECC. Note that the replica of a file can be seen as the most basic encoding of the file; but by using non-trivial ECC, the data storage scheme can have much higher fault tolerance and availability. By using ECC, we also have more freedom to decide how much and which data to place on each node, which can be used to improve performance in many aspects. In practice, obtaining a file by retrieving its pieces from different nodes of the network has becoming increasing popular — for instance, file swarming in peer-to-peer networks like FastTrack and Gnutella, and file retrieval from mirror servers — for which using ECC is a natural next step.

For a network file storage problem using ECC, its solution always consists of two parts — deciding *how many* and *which* codeword component to place on each node. These two procedures are called *Memory Allocation* and *Data Interleaving*, respectively. (The name, Data Interleaving, comes from the fact that for good performance, the locations of different codeword components' replicas always need to be well mixed.) Memory Allocation belongs to the more general field of Facility Location Problems, which, although usually NP hard, can obtain quite good approximation solutions by using LP rounding, primal-dual algorithms, local search and other techniques. Data Interleaving, however, is a field of very little previous study. Therefore, it became the primary focus of my research.

The only seminal work on Data Interleaving had been due to Naor and Roth [13], where they studied Data Interleaving for minimizing the total memory cost. In my studies, I have explored Data Interleaving for optimizing file-retrieval cost, file-recovery complexity and multi-access of data, introduced in the following.

Perfect Interleaving for optimizing file-retrieval cost The Data Interleaving procedure assumes that the number of codeword components to place on each node is given. Let n denote the number of components in the codeword for the file. Perfect Interleaving is a placement such that for *every* node, the n nearest *distinct* codeword components are just the n nearest components regardless of their distinctness. When the distances between nodes and components have a positive correlation with the cost of retrieving the components, Perfect Interleaving is the *best* solution one can hope for in a fault-free network, because it simultaneously optimizes the file-retrieval cost for every node. It also provides very strong tolerance to data loss. When the network is a cyclic graph, there always exist instances for which Perfect Interleaving does not exist. However, in paper [5] we proved that when the network is an edge-weighted tree, Perfect Interleaving always exists. We presented an efficient Perfect Interleaving algorithm, and proved that the algorithm can be used not just to find one Perfect Interleaving solution, but *all* the Perfect Interleaving solutions. Then, we showed how to trade the algorithm's generality for further reduced complexity.

In [6], we extended the network structures to be trees with parallel edges between adjacent vertices of opposite directions and uncorrelated weights, a model that simulates the asymmetry of transmission costs. A data placement scheme that generalizes Perfect Interleaving is studied for the graceful degradation of file-retrieval performance in faulty environments, and we prove that a solution always exists, too.

t-Interleaving for optimizing file-recovery complexity Consider a torus network where each edge causes transmission delay (distance) 1. A file is to be stored for all the nodes to access, and the maximum delay every node can tolerate is r. For every node in a torus, the number of nodes within distance r is the same, which we denote by B_r. To not only minimize but also perfectly balance the memory requirement for nodes, the best solution is to encode the file with an (n, B_r) maximum-distance separable code, store one codeword component on every node, and require that for each node, all the components within distance r are distinct — which is equivalent to requiring that replicas of the same component must have distance at least $2r + 1$ between them. (n is the number of components in the codeword, and B_r is the number of components sufficient for recovering the file.) Now our objective is to find a data placement where n can be as small as possible, because the smaller n can be, the more power we have to choose a code of lower complexity, especially its file-recovery (decoding) complexity.

Interestingly, this is the long studied *t-Interleaving Problem*. t-Interleaving is to color the nodes of a graph with as few colors as possible such that for any two nodes of the same color, their distance is at least t. t-Interleaving has been studied for about ten years [1], mainly in the Information Theory society. There were numerous works that studied t-Interleaving on arrays and their variations, but the problem of how to t-Interleave graphs that has modular (that is, wrapping-around) structures had been open. In [12], we presented the *first* algorithm for t-Interleaving graphs of modular structures, by solving the problem for two-dimensional tori, for both odd and even t.

Our results were multifold. We proved the necessary and sufficient conditions for the number of colors to achieve its minimum possible value (a lower bound based on a packing argument), and constructively proved that for large tori, that number exceeds the bound by at most one. For tori that are not sufficiently large, we also presented bounds for the color numbers, completing a general characterization of the problem. The t-Interleaving algorithm contributed a novel construction to the general field of interleaving. Particularly, it showed a new approach different from the well known and widely adopted Lattice-Interleaver methods. If we return to the original file storage problem, then the results not only present Data Interleaving solutions that minimize the file-recovery complexity, but also prove that for large tori and tori of certain other sizes, we can use just a simple parity code or sometimes, even only file partition, both of which have extremely low file-recovery complexity.

Multicluster Interleaving for multi-access of data Traditional interleaving problems and the two interleaving problems introduced above ask for only *local properties*, which can be understood as requiring that every small connected subgraph contains enough different codeword components (or colors). In [7], we defined and studied the Multicluster Interleaving problem, which extended the local properties of interleaving to global properties. Multicluster Interleaving requires that multiple connected subgraphs *together* contain enough different codeword components. Here the subgraphs can be almost arbitrarily chosen, and the required number of codeword components is usually larger than the number of components stored in each subgraph. Multicluster Interleaving is motivated by the situation that users have the capability to access a data-storage network through multiple, instead of one, access points, where Multicluster Interleaving can use this multi-access capability to reduce file-retrieval delay. It also has potential applications in data streaming. In [7], we studied the problem for paths and cycles, and presented several families of optimal interleaving constructions. Multicluster Interleaving has the distinct feature that the graph's order is a significant factor for the existence of solutions, and it is related to Block Designs. Our constructions contributed a novel interleaving method based on auxiliary graphs.

Besides Data Interleaving, my study also addressed **Memory Allocation**, the other component of file storage problems using ECC. In papers [6], [8], we presented several algorithms for minimizing the memory cost while guaranteeing the quality of file retrieval.

My research on *caching* focused on **En-Route Web Caching** in the Internet. An en-route caching system attaches caches to routers. It is similar to the Hierarchical Caching System widely deployed in the Internet today, but it needn't forward file requests to proxy servers and has a strong ability to be allocated in the inside, instead of on the edges, of the networks. In paper [9], we showed that existing file placement policies for caching all solved *restricted partial problems* of the file placement problem, therefore gave only sub-optimal solutions. We presented a dynamic programming algorithm of low complexity which computed the optimal solution. It was shown both analytically and experimentally that our algorithm outperformed previous policies. The algorithm can be implemented in a distributed way with a low level of overhead, and the successive caching operations of different nodes aggregately give the optimal file placement solution.

2.2 Wireless ad hoc communication and sensor networks

My research on Wireless Ad Hoc Communication and Sensor Networks mainly focused on the network topologies and their utilization. The specific topics included localization, planar spanner graph, topology control, geographical and compact routing, and rate and power allocation with signal interference. Those topics are basic, and many of them have counterpart problems in other ad hoc or overlay networks.

Localization, Planar Spanner and Routing In [2], we studied for the *first* time the using of local angles (the angles between adjacent edges) for localization of anchor-free sensor networks. The locations of sensor nodes are important for the meaning of collected data and for routing. The localization problem is to determine the node positions based only on the information obtained through the nodes' interactions, such as connectivity, edge lengths and angles. The study on localization has been extensive, but before our work, all the study on anchor-free localization focused on using connectivity and link lengths. Hence using angles creates a new dimension for this field. Local angles can be measured by using multiple ultrasound receivers or by using directional antennas. In [2], we studied both the power and the limitation of the angle information. We modelled the sensor network as a unit-disk graph (UDG), and proved that the localization problem is NP-hard. What's more, the $\sqrt{2}$-approximate localization (which relaxes the constraint on edge length by a factor of $\sqrt{2}$) and the topologically-equivalent localization (which focuses on edge crossing and completely relaxes the constraint on edge length) are NP-hard as well. Despite the hardness of localization, however, we proposed a practical localization algorithm that works surprisingly well. The algorithm is based on linear programming, and differs from all previous methods (e.g., Multidimensional Scaling, semi-definite programming, kinetic simulations, and various numerical and combinatorial optimization approaches). It recovers the network geometry almost truthfully in practice and excels the performance of previously known results. It is also demonstrated to be robust to both measurement noises and the variation of network models.

We presented in [2] an algorithm for finding a planar spanner of UDG, which is valuable for routing and topology control, by using local angles. This is the *first* algorithm that derives planar spanners by using less information than the nodes' coordinates. We combined the planar spanner with our localization algorithm for geographical routing and graph-labelling-based routing. The routing schemes achieved the same good performance as if the nodes' real locations were known.

In [3], we presented a novel naming and routing scheme for wireless and sensor networks. The scheme builds a name space that effectively captures the crucial geometric information of the networks and enables smooth, efficient and load-balanced routing. Although this scheme is location-free, its performance substantially exceeds that of existing geographical routing schemes based on known node coordinates.

Topology control Topology control is the process of controlling the topology of a wireless network by adjusting the coverage ranges of its wireless nodes. In [10], we explored a topology control method primarily based on local angles. Specifically, each node decides on its coverage range subject to the constraint that the angles between its incident out-edges are all below a certain threshold. One main advantage of using local angles is that they enable coverage ranges to adapt to the variation of node densities well. While other works have focused on topology control algorithms for achieving certain pre-specified objectives, we used the new approach of adjusting the constraints on local angles, and study how the resulting network topology's properties change accordingly. Using rigorous combinatorial and probabilistic analysis, we studied various topological properties (e.g., connectivity, routing path, degree, local minimum for geographical routing, etc.), and presented several variations of the topology control method. The findings showed that such a method achieves very good balance between the various aspects of network performance.

My research has also addressed the localization of mobile ad hoc networks through kinetic simulation, and overlay multicast [11]. Ongoing research includes rate and power allocation for sensor networks with signal interference, network design for routing, and several other topics.

3 Future research

My past research has armed me with the understanding of data storage/sharing and network structure design/utilization, as well as the more general background in algorithms, systems and information theory. As a career objective, I plan to explore broader topics in Information Networks. In the following, I present examples of the research directions that I intend to pursue soon.

Generalized geographical routing Geographical routing is a very important routing method, especially for ad hoc networks, mainly due to its high scalability. It is widely used for wireless networks, where it guides routing by using nodes' coordinates. It is also being used for overlay networks in the Internet, where overlay nodes are embedded into certain metric spaces to obtain their virtual coordinates. (A typical example is the Chord peer-to-peer network.) Currently, the performance of geographical routing is limited by the hardness of the embedding process, the metric distortion caused by embedding, and its sensitivity to particular network models (such as the UDG model for wireless networks). I am interested in studying new routing methods that generalize geographical routing, which use new backbone structures to guarantee message delivery and new shortcut links to guarantee efficiency. I am exploring such schemes that are robust and much less dependent on embedding.

Topology control for overlay networks Overlay networks constitute a major portion of the emerging Information Networks in the Internet. Typical examples include peer-to-peer networks and publish/subscribe service networks. People have devoted considerable effort to study the metric properties of the Internet, and a number of recent works [4] indicate that the metric of overlay nodes in the Internet often exhibits low dimensionality. I am interested in exploring the design of overlay network topologies that suit such metric properties, which should be very different from the currently popular topologies for overlay networks including regular and random expanders, trees and other combinatorial structures. It is likely that we can construct overlay networks that have efficient routing paths, good load balancing abilities and small well-separated pair decomposition (which can effectively reduce the complexity of many metric-related problems), as what happens for unit-disk graphs. The properties of the topology are fundamental for the overlay network's performance.

Kinetic Data Structures for data storage and transmission Data placement is a critical procedure of data storage systems in networks. The placement of data needs to suit the users' access patterns (e.g., retrieval frequencies). I am interested in using Kinetic Data Structures (KDS) to achieve efficient adjustment of the data locations (which is a field called Data Movement). KDS has been studied in computational geometry, with a main focus on maintaining the combinatorial or algebraic structures of moving objects. In practice, users' data access patterns have the property that their short-term behavior is predictable to some extent, which is suitable for KDS. KDS can help the system monitor the changes in users' access patterns more effectively and less frequently by removing the unnecessary computation when non-critical events happen. Besides data storage systems, KDS can also help data transmission by monitoring the states of network flows.

Network storage system People are having various types of digital equipments, and they want to access their data at different places. Both ask for the storage of data in the Internet, instead of on offline machines. Besides availability, network storage systems also have the important advantages of jointly storing people's data, optimizing the placement of shared files, and stronger abilities in data searching and categorization. Unlike peer-to-peer file-sharing networks, which use the best-effort-search approach, network storage systems need to guarantee the safety of data. I am interested in selecting good coding schemes for data, and in designing storage schemes that balance well between efficiency and robustness. I am also interested in many other aspects, such as data lookup and system growth.

Combining Information Theory with network problems Many structures and algorithms designed in computer science for network problems have not taken information theory into account. For example, in wireless networks, signal interference is a very important factor that affects the simultaneous availability of links. However, it is seldom taken into account in related algorithms and designed structures. I am interested in studying the combination of information theory and network problems, with which the performance of the algorithms can be evaluated better.

Information Networks is a developing field of vast demand. I am interested in understanding and solving its key challenges, which cross different disciplines. I am especially interested in novel designs that have fundamental impact on information systems. This general field, with its great research values and tremendous applications, is also a subject that can benefit from the collaboration of researchers in many different areas.

References

[1] M. Blaum and J. Bruck, Interleaving for Two-Dimensional Cluster Errors, *Workshop on Data Encoding of Page-Oriented Optical Memories*, Phoenix, Arizona, March 1996.

[2] J. Bruck, J. Gao and A. Jiang, Localization and Routing in Sensor Networks by Local Angle Information, *Proc. the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, pp. 181-192, May 2005.

[3] J. Bruck, J. Gao and A. Jiang, MAP: Medial Axis Based Geometric Routing in Sensor Networks, *Proc. the 11th ACM International Conference on Mobile Computing and Networking (MobiCom'05)*, August–September, 2005.

[4] F. Dabek, R. Cox, F. Kaashoek, R. Morris, Vivaldi: A Decentralized Network Coordinate System, *Proceedings of SIGCOMM'04*, Portland, OR, Aug. 2004.

[5] A. Jiang and J. Bruck, Diversity Coloring for Distributed Data Storage in Networks, submitted to *IEEE Transactions on Information Theory.*

[6] A. Jiang and J. Bruck, Network File Storage with Graceful Performance Degradation, *ACM Transactions on Storage*, March, 2005, to appear.

[7] A. Jiang and J. Bruck, Multicluster Interleaving on Paths and Cycles, *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 597-611, February 2005.

[8] A. Jiang and J. Bruck, Memory Allocation in Information Storage Networks, *Proc. IEEE International Symposium on Information Theory (ISIT'03)*, pp. 453, June–July, 2003.

[9] A. Jiang and J. Bruck, Optimal Content Placement for En-route Web Caching, *Proc. the 2nd IEEE International Symposium on Network Computing and Applications*, pp. 9-16, April 2003.

[10] A. Jiang and J. Bruck, Monotone Percolation and The Topology Control of Wireless Networks, *Proc. the 24th IEEE INFOCOM*, March 2005.

[11] A. Jiang and J. Bruck, Geometric Embedding for Mobile Ad Hoc Networks, manuscript.

[12] A. Jiang, M. Cook and J. Bruck, Optimal t-Interleaving on Tori, *Proc. IEEE International Symposium on Information Theory (ISIT'04)*, pp. 22, June–July, 2004.

[13] M. Naor and R. M. Roth, Optimal File Sharing in Distributed Networks, *SIAM Journal on Computing*, vol. 24, no. 1, pp. 158-183, 1995.