

SEA
2020

18th Symposium on Experimental Algorithms
Catania, June 16-18, 2020

The book of abstracts



UNIVERSITÀ
degli STUDI
di CATANIA

GNCS Gruppo Nazionale per
il Calcolo Scientifico

iNSAM Istituto Nazionale di
Alta Matematica F. Severi

SESSION 1 - June 16, 2020 - AM

Indexing Compressed Text: A Tale of Time and Space (Invited Talk)

Authors: Prezza, Nicola

Text indexing is a classical algorithmic problem that has been studied for over four decades. The earliest optimal-time solution to the problem, the suffix tree [Weiner, 1973], dates back to 1973 and requires up to two orders of magnitude more space than the text to be stored. In the year 2000, two breakthrough works [Grossi and Vitter, 2000; Ferragina and Manzini, 2000] showed that this space overhead is not necessary: both the index and the text can be stored in a space proportional to the text's entropy. These contributions had an enormous impact in bioinformatics: nowadays, the two most widely-used DNA aligners employ compressed indexes [Li and Durbin, 2009; Langmead et al., 2009]. In recent years, it became apparent that entropy had reached its limits: modern datasets (for example, collections of thousands of human genomes) are extremely large but very repetitive and, by its very definition, entropy cannot compress repetitive texts [S. Kreft and G. Navarro, 2013]. To overcome this problem, a new generation of indexes based on dictionary compressors (for example, LZ77 and run-length BWT) emerged [S. Kreft and G. Navarro, 2013; Gagie et al., 2020; F. Claude and G. Navarro, 2012], together with generalizations of the indexing problem to labeled graphs [Ferragina et al., 2009; Sirén et al., 2014; Travis Gagie et al., 2017]. This talk is a short and friendly survey of the landmarks of this fascinating path that took us from suffix trees to the most modern compressed indexes on labeled graphs.

High-Quality Hierarchical Process Mapping

Authors: Faraj, Marcelo Fonseca; van der Grinten, Alexander; Meyerhenke, Henning; Träff, Jesper Larsson; Schulz, Christian

Partitioning graphs into blocks of roughly equal size such that few edges run between blocks is a frequently needed operation when processing graphs on a parallel computer. When a topology of a distributed system is known, an important task is then to map the blocks of the partition onto the processors such that the overall communication cost is reduced. We present novel multilevel algorithms that integrate graph partitioning and process mapping. Important ingredients of our algorithm include fast label propagation, more localized local search, initial partitioning, as well as a compressed data structure to compute processor distances without storing a distance matrix. Moreover, our algorithms are able to exploit a given hierarchical structure of the distributed system under consideration. Experiments indicate that our algorithms speed up the overall mapping process and, due to the integrated multilevel approach, also find much better solutions in practice. For example, one configuration of our algorithm yields similar solution quality as the previous state-of-the-art in terms of mapping quality for large numbers of partitions while being a factor 9.3 faster. Compared to the currently fastest iterated multilevel mapping algorithm Scotch, we obtain 16% better solutions while investing slightly more running time.

Probing a Set of Trajectories to Maximize Captured Information

Authors: Fekete, Sándor P. ; Hill, Alexander ; Krupke, Dominik ; Mayer, Tyler ; Mitchell, Joseph S. B. ; Parekh, Ojas ; Phillips, Cynthia A.

We study a trajectory analysis problem we call the Trajectory Capture Problem (TCP), in which, for a given input set T of trajectories in the plane, and an integer $k \geq 2$, we seek to compute a set of k points ("portals") to maximize the total weight of all subtrajectories of T between pairs of portals. This problem naturally arises in trajectory analysis and summarization.

We show that the TCP is NP-hard (even in very special cases) and give some first approximation results. Our main focus is on attacking the TCP with practical algorithm-engineering approaches, including integer linear programming (to solve instances to provable optimality) and local search methods. We study the integrality gap arising from such approaches. We analyze our methods on different classes of data, including benchmark instances that we generate. Our goal is to understand the best performing heuristics, based on both solution time and solution quality. We demonstrate that we are able to compute provably optimal solutions for real-world instances.

Storing Set Families More Compactly with Top ZDDs

Authors: Matsuda, Kotaro ; Denzumi, Shuhei ; Sadakane, Kunihiko

Zero-suppressed Binary Decision Diagrams (ZDDs) are data structures for representing set families in a compressed form. With ZDDs, many valuable operations on set families can be done in time polynomial in ZDD size. In some cases, however, the size of ZDDs for representing large set families becomes too huge to store them in the main memory.

This paper proposes top ZDD, a novel representation of ZDDs which uses less space than existing ones. The top ZDD is an extension of top tree, which compresses trees, to compress directed acyclic graphs by sharing identical subgraphs. We prove that navigational operations on ZDDs can be done in time poly-logarithmic in ZDD size, and show that there exist set families for which the size of the top ZDD is exponentially smaller than that of the ZDD. We also show experimentally that our top ZDDs have smaller size than ZDDs for real data.

Fast and Simple Compact Hashing via Bucketing

Authors: Köppl, Dominik ; Puglisi, Simon J. ; Raman, Rajeev

Compact hash tables store a set S of n key-value pairs, where the keys are from the universe $U = \{0, \dots, u-1\}$, and the values are v -bit integers, in close to $B(u, n) + nv$ bits of space, where $\{b(u, n)\} = \log_2 \text{binom}(u, n)$ is the information-theoretic lower bound for representing the set of keys in S , and support operations insert, delete and lookup on S .

Compact hash tables have received significant attention in recent years, and approaches dating back to Cleary [IEEE T. Comput, 1984], as well as more recent ones have been implemented and used in a number of applications. However, the wins on space usage of these approaches are outweighed by their slowness relative to conventional hash tables. In this

paper, we demonstrate that compact hash tables based upon a simple idea of bucketing practically outperform existing compact hash table implementations in terms of memory usage and construction time, and existing fast hash table implementations in terms of memory usage (and sometimes also in terms of construction time).

A related notion is that of a compact Hash ID map, which stores a set \hat{S} of n keys from U , and implicitly associates each key in \hat{S} with a unique value (its ID), chosen by the data structure itself, which is an integer of magnitude $O(n)$, and supports inserts and lookups on \hat{S} , while using close to $B(u,n)$ bits. One of our approaches is suitable for use as a compact Hash ID map.

SESSION 2 - June 16, 2020 - PM

Effect of Initial Assignment on Local Search Performance for Max Sat

Authors: Berend, Daniel ; Twitto, Yochai

In this paper, we explore the correlation between the quality of initial assignments provided to local search heuristics and that of the corresponding final assignments. We restrict our attention to the Max r-Sat problem and to one of the leading local search heuristics - Configuration Checking Local Search (CCLS). We use a tailored version of the Method of Conditional Expectations (MOCE) to generate initial assignments of diverse quality.

We show that the correlation in question is significant and long-lasting. Namely, even when we delve deeper into the local search, we are still in the shadow of the initial assignment. Thus, under practical time constraints, the quality of the initial assignment is crucial to the performance of local search heuristics.

To demonstrate our point, we improve CCLS by combining it with MOCE. Instead of starting CCLS from random initial assignments, we start it from excellent initial assignments, provided by MOCE. Indeed, it turns out that this kind of initialization provides a significant improvement of this state-of-the-art solver. This improvement becomes more and more significant as the instance grows.

Enumerating All Subgraphs Under Given Constraints Using Zero-Suppressed Sentential Decision Diagrams

Authors: Nakahata, Yu ; Nishino, Masaaki ; Kawahara, Jun ; Minato, Shin-ichi

Subgraph enumeration is a fundamental task in computer science. Since the number of subgraphs can be large, some enumeration algorithms exploit compressed representations for efficiency. One such representation is the Zero-suppressed Binary Decision Diagram (ZDD). ZDDs can represent the set of subgraphs compactly and support several poly-time queries, such as counting and random sampling. Researchers have proposed efficient algorithms to construct ZDDs representing the set of subgraphs under several constraints, which yield fruitful results in many applications.

Recently, Zero-suppressed Sentential Decision Diagrams (ZSDDs) have been proposed as variants of ZDDs. ZSDDs can be smaller than ZDDs when representing the same set of subgraphs. However, efficient algorithms to construct ZSDDs are known only for specific types of subgraphs: matchings and paths.

We propose a novel framework to construct ZSDDs representing sets of subgraphs under given constraints. Using our framework, we can construct ZSDDs representing several sets of subgraphs such as matchings, paths, cycles, and spanning trees. We show the bound of sizes of constructed ZSDDs by the branch-width of the input graph, which is smaller than that of ZDDs by the path-width. Experiments show that our methods can construct ZSDDs faster than ZDDs and that the constructed ZSDDs are smaller than ZDDs when representing the same set of subgraphs.

Engineering Exact Quasi-Threshold Editing

Authors: Gottesbüren, Lars ; Hamann, Michael ; Schoch, Philipp ; Strasser, Ben ; Wagner, Dorothea ; Zühlsdorf, Sven

Quasi-threshold graphs are $\{C_4, P_4\}$ -free graphs, i.e., they do not contain any cycle or path of four nodes as an induced subgraph. We study the $\{C_4, P_4\}$ -free editing problem, which is the problem of finding a minimum number of edge insertions or deletions to transform an input graph into a quasi-threshold graph. This problem is NP-hard but fixed-parameter tractable (FPT) in the number of edits by using a branch-and-bound algorithm and admits a simple integer linear programming formulation (ILP). Both methods are also applicable to the general \mathcal{F} -free editing problem for any finite set of graphs \mathcal{F} . For the FPT algorithm, we introduce a fast heuristic for computing high-quality lower bounds and an improved branching strategy. For the ILP, we engineer several variants of row generation. We evaluate both methods for quasi-threshold editing on a large set of protein similarity graphs. For most instances, our optimizations speed up the FPT algorithm by one to three orders of magnitude. The running time of the ILP, that we solve using Gurobi, becomes only slightly faster. With all optimizations, the FPT algorithm is slightly faster than the ILP, even when listing all solutions. Additionally, we show that for almost all graphs, solutions of the previously proposed quasi-threshold editing heuristic QTM are close to optimal.

Advanced Flow-Based Multilevel Hypergraph Partitioning

Authors: Gottesbüren, Lars ; Hamann, Michael ; Schlag, Sebastian ; Wagner, Dorothea

The balanced hypergraph partitioning problem is to partition a hypergraph into k disjoint blocks of bounded size such that the sum of the number of blocks connected by each hyperedge is minimized. We present an improvement to the flow-based refinement framework of KaHyPar-MF, the current state-of-the-art multilevel k -way hypergraph partitioning algorithm for high-quality solutions. Our improvement is based on the recently proposed HyperFlowCutter algorithm for computing bipartitions of unweighted hypergraphs by solving a sequence of incremental maximum flow problems. Since vertices and hyperedges are aggregated during the coarsening phase, refinement algorithms employed in the multilevel setting must be able to handle both weighted hyperedges and weighted vertices - even if the initial input hypergraph is unweighted. We therefore enhance HyperFlowCutter to

handle weighted instances and propose a technique for computing maximum flows directly on weighted hypergraphs. We compare the performance of two configurations of our new algorithm with KaHyPar-MF and seven other partitioning algorithms on a comprehensive benchmark set with instances from application areas such as VLSI design, scientific computing, and SAT solving. Our first configuration, KaHyPar-HFC, computes slightly better solutions than KaHyPar-MF using significantly less running time. The second configuration, KaHyPar-HFC*, computes solutions of significantly better quality and is still slightly faster than KaHyPar-MF. Furthermore, in terms of solution quality, both configurations also outperform all other competing partitioners.

Pattern Discovery in Colored Strings

Authors: Lipták, Zsuzsanna ; Puglisi, Simon J. ; Rossi, Massimiliano

We consider the problem of identifying patterns of interest in colored strings. A colored string is a string in which each position is colored with one of a finite set of colors. Our task is to find substrings that always occur followed by the same color at the same distance. The problem is motivated by applications in embedded systems verification, in particular, assertion mining. The goal there is to automatically infer properties of the embedded system from the analysis of its simulation traces. We show that the number of interesting patterns is upper-bounded by $\mathcal{O}(n^2)$ where n is the length of the string. We introduce a baseline algorithm with $\mathcal{O}(n^2)$ running time which identifies all interesting patterns for all colors in the string satisfying certain minimality conditions. When one is interested in patterns related to only one color, we provide an algorithm that identifies patterns in $\mathcal{O}(n^2 \log n)$ time, but is faster than the first algorithm in practice, both on simulated and on real-world patterns.

Fast and Linear-Time String Matching Algorithms Based on the Distances of q-Gram Occurrences

Authors: Kobayashi, Satoshi ; Hendrian, Diptarama ; Yoshinaka, Ryo ; Shinohara, Ayumi

Given a text T of length n and a pattern P of length m , the string matching problem is a task to find all occurrences of P in T . In this study, we propose an algorithm that solves this problem in $\mathcal{O}((n + m)q)$ time considering the distance between two adjacent occurrences of the same q -gram contained in P . We also propose a theoretical improvement of it which runs in $\mathcal{O}(n + m)$ time, though it is not necessarily faster in practice. We compare the execution times of our and existing algorithms on various kinds of real and artificial datasets such as an English text, a genome sequence and a Fibonacci string. The experimental results show that our algorithm is as fast as the state-of-the-art algorithms in many cases, particularly when a pattern frequently appears in a text.

SESSION 3 - June 17, 2020 - AM

Algorithm Engineering for Sorting and Searching, and All That (Invited Talk)

Authors: Edelkamp, Stefan

We look at several proposals to engineer the set of fundamental searching and sorting algorithms. Aspects are improving locality of disk access and cache access, the efficiency tuning by reducing the number of branch mispredictions, and reducing at leading factors hidden in the Big-Oh notation. These studies in algorithm engineering, in turn, lead to exiting new algorithm designs. On the practical side, we will establish that efficient sorting and searching algorithms are in tight collaboration, as sorting is used for finding duplicates in disk-based search, and heap structures designed for efficient graph search can be exploited in classical and adaptive sorting. We indicate the effects of engineered sorting and searching for combined task and motion planning.

Faster Fully Dynamic Transitive Closure in Practice

Authors: Hanauer, Kathrin ; Henzinger, Monika ; Schulz, Christian

The fully dynamic transitive closure problem asks to maintain reachability information in a directed graph between arbitrary pairs of vertices, while the graph undergoes a sequence of edge insertions and deletions. The problem has been thoroughly investigated in theory and many specialized algorithms for solving it have been proposed in the last decades. In two large studies [Frigioni et al., 2001; Krommidas and Zaroliagis, 2008], a number of these algorithms have been evaluated experimentally against simple, static algorithms for graph traversal, showing the competitiveness and even superiority of the simple algorithms in practice, except for very dense random graphs or very high ratios of queries. A major drawback of those studies is that only small and mostly randomly generated graphs are considered. In this paper, we engineer new algorithms to maintain all-pairs reachability information which are simple and space-efficient. Moreover, we perform an extensive experimental evaluation on both generated and real-world instances that are several orders of magnitude larger than those in the previous studies. Our results indicate that our new algorithms outperform all state-of-the-art algorithms on all types of input considerably in practice.

Concurrent Expandable AMQs on the Basis of Quotient Filters

Authors: Maier, Tobias ; Sanders, Peter ; Williger, Robert

A quotient filter is a cache efficient Approximate Membership Query (AMQ) data structure. Depending on the fill degree of the filter most insertions and queries only need to access one or two consecutive cache lines. This makes quotient filters very fast compared to the more commonly used Bloom filters that incur multiple independent memory accesses depending on the false positive rate. However, concurrent Bloom filters are easy to implement and can be implemented lock-free while concurrent quotient filters are not as simple. Usually concurrent quotient filters work by

using an external array of locks - each protecting a region of the table. Accessing this array incurs one additional memory access per operation. We propose a new locking scheme that has no memory overhead. Using this new locking scheme we achieve 1.6× times higher insertion performance and over 2.1× higher query performance than with the common external locking scheme. Another advantage of quotient filters over Bloom filters is that a quotient filter can change its capacity when it is becoming full. We implement this growing technique for our concurrent quotient filters and adapt it in a way that allows unbounded growing while keeping a bounded false positive rate. We call the resulting data structure a fully expandable quotient filter. Its design is similar to scalable Bloom filters, but we exploit some concepts inherent to quotient filters to improve the space efficiency and the query speed. Additionally, we propose several quotient filter variants that are aimed to reduce the number of status bits (2-status-bit variant) or to simplify concurrent implementations (linear probing quotient filter). The linear probing quotient filter even leads to a lock-free concurrent filter implementation. This is especially interesting, since we show that any lock-free implementation of other common quotient filter variants would incur significant overheads in the form of additional data fields or multiple passes over the accessed data. The code produced as part of this submission can be found at <https://www.github.com/Toobiased/lpqfilter>.

Faster Multi-Modal Route Planning With Bike Sharing Using ULTRA

Authors: Sauer, Jonas ; Wagner, Dorothea ; Zündorf, Tobias

We study multi-modal route planning in a network comprised of schedule-based public transportation, unrestricted walking, and cycling with bikes available from bike sharing stations. So far this problem has only been considered for scenarios with at most one bike sharing operator, for which MCR is the best known algorithm [Delling et al., 2013]. However, for practical applications, algorithms should be able to distinguish between bike sharing stations of multiple competing bike sharing operators. Furthermore, MCR has recently been outperformed by ULTRA for multi-modal route planning scenarios without bike sharing [Baum et al., 2019]. In this paper, we present two approaches for modeling multi-modal transportation networks with multiple bike sharing operators: The operator-dependent model requires explicit handling of bike sharing stations within the algorithm, which we demonstrate with an adapted version of MCR. In the operator-expanded model, all relevant information is encoded within an expanded network. This allows for applying any multi-modal public transit algorithm without modification, which we show for ULTRA. We proceed by describing an additional preprocessing step called operator pruning, which can be used to accelerate both approaches. We conclude our work with an extensive experimental evaluation on the networks of London, Switzerland, and Germany. Our experiments show that the new preprocessing technique accelerates both approaches significantly, with the fastest algorithm (ULTRA-RAPTOR with operator pruning) being more than an order of magnitude faster than the basic MCR approach. Moreover, the ULTRA preprocessing step also benefits from operator pruning, as its running time is reduced by a factor of 14 to 20.

Efficient Route Planning with Temporary Driving Bans, Road Closures, and Rated Parking Areas

Authors: Kleff, Alexander ; Schulz, Frank ; Wagenblatt, Jakob ; Zeitz, Tim

We study the problem of planning routes in road networks when certain streets or areas are closed at certain times. For heavy vehicles, such areas may be very large since many European countries impose temporary driving bans during the night or on weekends. In this setting, feasible routes may require waiting at parking areas, and several feasible routes with different trade-offs between waiting and driving detours around closed areas may exist. We propose a novel model in which driving and waiting are assigned abstract costs, and waiting costs are location-dependent to reflect the different quality of the parking areas. Our goal is to find Pareto-optimal routes with regards to arrival time at the destination and total cost. We investigate the complexity of the model and determine a necessary constraint on the cost parameters such that the problem is solvable in polynomial time. We present a thoroughly engineered implementation and perform experiments on a production-grade real world data set. The experiments show that our implementation can answer realistic queries in around a second or less which makes it feasible for practical application.

SESSION 4 - June 18, 2020 - AM

Algorithm Engineering for High-Dimensional Similarity Search Problems (Invited Talk)

Authors: Aumüller, Martin

Similarity search problems in high-dimensional data arise in many areas of computer science such as data bases, image analysis, machine learning, and natural language processing. One of the most prominent problems is finding the k nearest neighbors of a data point $q \in \mathbb{R}^d$ in a large set of data points $S \subset \mathbb{R}^d$, under same distance measure such as Euclidean distance. In contrast to lower dimensional settings, we do not know of worst-case efficient data structures for such search problems in high-dimensional data, i.e., data structures that are faster than a linear scan through the data set. However, there is a rich body of (often heuristic) approaches that solve nearest neighbor search problems much faster than such a scan on many real-world data sets. As a necessity, the term solve means that these approaches give approximate results that are close to the true k -nearest neighbors. In this talk, we survey recent approaches to nearest neighbor search and related problems.

The talk consists of three parts: (1) What makes nearest neighbor search difficult? (2) How do current state-of-the-art algorithms work? (3) What are recent advances regarding similarity search on GPUs, in distributed settings, or in external memory?

Space and Time Trade-Off for the k Shortest Simple Paths Problem

Authors: Al Zoobi, Ali ; Coudert, David ; Nisse, Nicolas

The k shortest simple path problem (kSSP) asks to compute a set of top- k shortest simple paths from a vertex s to a vertex t in a digraph. Yen (1971) proposed the first algorithm with the best known theoretical complexity of $O(kn(m+n \log n))$ for a digraph with n vertices and m arcs. Since then, the problem has been widely studied from an algorithm engineering perspective, and impressive improvements have been achieved. In particular, Kurz and Mutzel (2016) proposed a sidetracks-based (SB) algorithm which is currently the fastest solution. In this work, we propose two improvements of this algorithm.

We first show how to speed up the SB algorithm using dynamic updates of shortest path trees. We did experiments on some road networks of the 9th DIMAC'S challenge with up to about half a million nodes and one million arcs. Our computational results show an average speed up by a factor of 1.5 to 2 with a similar working memory consumption as SB. We then propose a second algorithm enabling to significantly reduce the working memory at the cost of an increase of the running time (up to two times slower). Our experiments on the same data set show, on average, a reduction by a factor of 1.5 to 2 of the working memory.

An Algorithm for the Exact Treedepth Problem

Authors: Trimble, James

We present a novel algorithm for the minimum-depth elimination tree problem, which is equivalent to the optimal treedepth decomposition problem. Our algorithm makes use of two cheaply-computed lower bound functions to prune the search tree, along with symmetry-breaking and domination rules. We present an empirical study showing that the algorithm outperforms the current state-of-the-art solver (which is based on a SAT encoding) by orders of magnitude on a range of graph classes.

Algorithms for New Types of Fair Stable Matchings

Authors: Cooper, Frances ; Manlove, David

We study the problem of finding "fair" stable matchings in the Stable Marriage problem with Incomplete lists (SMI). For an instance I of SMI there may be many stable matchings, providing significantly different outcomes for the sets of men and women. We introduce two new notions of fairness in SMI. Firstly, a regret-equal stable matching minimises the difference in ranks of a worst-off man and a worst-off woman, among all stable matchings. Secondly, a min-regret sum stable matching minimises the sum of ranks of a worst-off man and a worst-off woman, among all stable matchings. We present two new efficient algorithms to find stable matchings of these types. Firstly, the Regret-Equal Degree Iteration Algorithm finds a regret-equal stable matching in $O(d_o nm)$ time, where d_o is the absolute difference in ranks between a worst-off man and a worst-off woman in the man-optimal stable matching, n is the number of men or women, and m is the total length of all preference lists. Secondly, the Min-Regret Sum Algorithm finds a min-regret sum stable matching in $O(d_s m)$ time, where d_s is the difference in the ranks between a worst-off man in each of the woman-optimal and man-optimal stable matchings. Experiments to compare several types of fair optimal stable matchings were conducted and show that

the Regret-Equal Degree Iteration Algorithm produces matchings that are competitive with respect to other fairness objectives. On the other hand, existing types of "fair" stable matchings did not provide as close an approximation to regret-equal stable matchings.

Crystal Structure Prediction via Oblivious Local Search

Authors: Antypov, Dmytro ; Deligkas, Argyrios ; Gusev, Vladimir ; Rosseinsky, Matthew J. ; Spirakis, Paul G. ; Theofilatos, Michail

We study Crystal Structure Prediction, one of the major problems in computational chemistry. This is essentially a continuous optimization problem, where many different, simple and sophisticated, methods have been proposed and applied. The simple searching techniques are easy to understand, usually easy to implement, but they can be slow in practice. On the other hand, the more sophisticated approaches perform well in general, however almost all of them have a large number of parameters that require fine tuning and, in the majority of the cases, chemical expertise is needed in order to properly set them up. In addition, due to the chemical expertise involved in the parameter-tuning, these approaches can be biased towards previously-known crystal structures. Our contribution is twofold. Firstly, we formalize the Crystal Structure Prediction problem, alongside several other intermediate problems, from a theoretical computer science perspective. Secondly, we propose an oblivious algorithm for Crystal Structure Prediction that is based on local search. Oblivious means that our algorithm requires minimal knowledge about the composition we are trying to compute a crystal structure for. In addition, our algorithm can be used as an intermediate step by any method. Our experiments show that our algorithms outperform the standard basin hopping, a well studied algorithm for the problem.

SESSION 5 - June 18, 2020 - PM

Variable Shift SDD: A More Succinct Sentential Decision Diagram

Authors: Nakamura, Kengo ; Denzumi, Shuhei ; Nishino, Masaaki

The Sentential Decision Diagram (SDD) is a tractable representation of Boolean functions that subsumes the famous Ordered Binary Decision Diagram (OBDD) as a strict subset. SDDs are attracting much attention because they are more succinct than OBDDs, as well as having canonical forms and supporting many useful queries and transformations such as model counting and Apply operation. In this paper, we propose a more succinct variant of SDD named Variable Shift SDD (VS-SDD). The key idea is to create a unique representation for Boolean functions that are equivalent under a specific variable substitution. We show that VS-SDDs are never larger than SDDs and there are cases in which the size of a VS-SDD is exponentially smaller than that of an SDD. Moreover, despite such succinctness, we show that numerous basic operations that are supported in polytime with SDD are also supported in polytime with VS-SDD. Experiments confirm that VS-SDDs are significantly more succinct than SDDs when applied to classical planning instances, where inherent symmetry exists.

Engineering Fused Lasso Solvers on Trees

Authors: Kuthe, Elias ; Rahmann, Sven

The graph fused lasso optimization problem seeks, for a given input signal $y=(y_i)$ on nodes $i \in V$ of a graph $G=(V,E)$, a reconstructed signal $x=(x_i)$ that is both element-wise close to y in quadratic error and also has bounded total variation (sum of absolute differences across edges), thereby favoring regionally constant solutions. An important application is denoising of spatially correlated data, especially for medical images. Currently, fused lasso solvers for general graph input reduce the problem to an iteration over a series of "one-dimensional" problems (on paths or line graphs), which can be solved in linear time. Recently, a direct fused lasso algorithm for tree graphs has been presented, but no implementation of it appears to be available.

We here present a simplified exact algorithm and additionally a fast approximation scheme for trees, together with engineered implementations for both. We empirically evaluate their performance on different kinds of trees with distinct degree distributions (simulated trees; spanning trees of road networks, grid graphs of images, social networks). The exact algorithm is very efficient on trees with low node degrees, which covers many naturally arising graphs, while the approximation scheme can perform better on trees with several higher-degree nodes when limiting the desired accuracy to values that are useful in practice.

Finding Structurally and Temporally Similar Trajectories in Graphs

Authors: Grossi, Roberto ; Marino, Andrea ; Moghtasedi, Shima

The analysis of similar motions in a network provides useful information for different applications like route recommendation. We are interested in algorithms to efficiently retrieve trajectories that are similar to a given query trajectory. For this task many studies have focused on extracting the geometrical information of trajectories. In this paper we investigate the properties of trajectories moving along the paths of a network. We provide a similarity function by making use of both the temporal aspect of trajectories and the structure of the underlying network. We propose an approximation technique that offers the top-k similar trajectories with respect to a query trajectory in an efficient way with acceptable precision. We investigate our method over real-world networks, and our experimental results show the effectiveness of the proposed method.

Zippering Segment Trees

Authors: Barth, Lukas ; Wagner, Dorothea

Stabbing queries in sets of intervals are usually answered using segment trees. A dynamic variant of segment trees has been presented by van Kreveld and Overmars, which uses red-black trees to do rebalancing operations. This paper presents zippering segment trees - dynamic segment trees based on zip trees, which were recently introduced by Tarjan et al. To facilitate zippering segment trees, we show how to uphold certain segment tree

properties during the operations of a zip tree. We present an in-depth experimental evaluation and comparison of dynamic segment trees based on red-black trees, weight-balanced trees and several variants of the novel zipping segment trees. Our results indicate that zipping segment trees perform better than rotation-based alternatives.

Fast and Stable Repartitioning of Road Networks

Authors: Buchhold, Valentin ; Delling, Daniel ; Schieferdecker, Dennis ; Wegner, Michael

We study the problem of graph partitioning for evolving road networks. While the road network of the world is mostly stable, small updates happen on a relatively frequent basis, as can be observed with the OpenStreetMap project (<http://www.openstreetmap.org>). For various reasons, professional applications demand the graph partition to stay roughly the same over time, and that changes are limited to areas where graph updates occur. In this work, we define the problem, present algorithms to satisfy the stability needs, and evaluate our techniques on continental-sized road networks. Besides the stability gains, we show that, when the changes are low and local, running our novel techniques is an order of magnitude faster than running graph partitioning from scratch.

Path Query Data Structures in Practice

Authors: He, Meng ; Kazi, Serikzhan

We perform experimental studies on data structures that answer path median, path counting, and path reporting queries in weighted trees. These query problems generalize the well-known range median query problem in arrays, as well as the 2d orthogonal range counting and reporting problems in planar point sets, to tree structured data. We propose practical realizations of the latest theoretical results on path queries. Our data structures, which use tree extraction, heavy-path decomposition and wavelet trees, are implemented in both succinct and pointer-based form. Our succinct data structures are further specialized to be plain or entropy-compressed. Through experiments on large sets, we show that succinct data structures for path queries may present a viable alternative to standard pointer-based realizations, in practical scenarios. Compared to naive approaches that compute the answer by explicit traversal of the query path, our succinct data structures are several times faster in path median queries and perform comparably in path counting and path reporting queries, while being several times more space-efficient. Plain pointer-based realizations of our data structures, requiring a few times more space than the naive ones, yield up to 100-times speed-up over them.