Editorial: ALENEX 2017 Special Issue

This special issue is dedicated to selected papers from the 19th Meeting on Algorithm Engineering and Experiments (ALENEX17), which took place in Barcelona January 17–18, 2017. ALENEX is a leading venue for the presentation of original research results on all aspects of algorithm engineering, and the implementation and experimental evaluation of algorithms and data structures. Out of 50 submissions, the program committee selected 20 for presentation at the meeting and inclusion in the proceedings. This volume contains four outstanding ALENEX'17 papers that were expanded and revised for the journal publication. All papers went through the rigorous review process according to the high standards of the ACM Journal of Experimental Algorithmics.

In the article titled “Computing Critical Nodes in Directed Graphs,” authors Paudel, Georgiadis, and Italiano (<https://doi.org/10.1145/3228332>) consider the “Critical Node Detection Problem” (CNDP) in directed graphs. Given a directed graph G and a positive integer k , CNDP asks for identifying a subset S of at most k vertices or “critical nodes” of G such that $G \setminus S$ has minimum pair-wise strong connectivity. This problem arises in social network analysis, assessing network vulnerabilities, network immunization, and the study of covert terrorist networks among other fields. All prior work on CNDP considered only the undirected version of the problem. Though the problem is NP-hard for both directed and undirected graphs, solving connectivity problems are known to be much harder on directed graphs than on undirected graphs. The current article provides a nontrivial linear-time algorithm for solving CNDP in directed graphs when $k = 1$, which builds on a very recent work by a subset of the authors (Georgiadis and Italiano with Nikos Parotsidis) on strong connectivity under node failures. An efficient heuristic algorithm for general k is obtained by repeatedly removing the most critical node of the current graph using the new linear-time algorithm until k of them are removed. Experimental results show that the new algorithm for $k = 1$ runs orders of magnitude faster than the naïve algorithm that finds the most critical node by evaluating $G \setminus \{v\}$ for every vertex v of G . Also, the new heuristic algorithm outperforms other popular node selection heuristics, such as betweenness centrality, PageRank, and maximum degree, in terms of solution quality.

The article “An Efficient Algorithm for the 1D Total Visibility-Index Problem and Its Parallelization” by Karsin et al. (<https://doi.org/10.1145/3209685>) addresses visibility problems, which are paramount in practice and have been considered in a large variety of settings. In particular, terrains are an important class of geometric domains, intermediate between two-dimensional polygons and full three-dimensional environments. An important problem is computing the total visibility index of points in a set P on a terrain, which is finding, for every point $p \in P$, the number of points of P that are visible from p . Even for one-dimensional terrains (x -monotone chains, or graphs of piecewise-linear functions), no subquadratic-time algorithm for computing the index has been known before. The breakthrough article by Karsin et al. solves this 1D version in nearly-linear time; the solution is based on geometric duality, which reduces the problem to red–blue segment intersections counting. The article also presents a parallel version of the algorithm, as well as extensive experiments mixing and comparing the new algorithms with existing ones. The parallel version is able to handle a terrain with an impressive 2^{24} vertices within 1 minute on as few as 16 cores.

The article titled “Computing the Expected Value and Variance of Geometric Measures” by Tsirogiannis, Staals, and Pellissier (<https://doi.org/10.1145/3228331>) deals with imprecise/stochastic input. Working with such input brings algorithms engineering closer to the reality, since, in practice, acquired data is rarely exact. The article by Tsirogiannis et al. gives algorithms for computing several geometric measures of a set of points under two assumptions of how the input points are obtained from a “master set”: either every point is in the input with some probability (independently of the other points) or a fixed-size subset of the master set is chosen uniformly at random. Computing the measures of imprecise point sets is not only a core computational-geometry problem but has applications also in ecology. The algorithms in the article outperform the standard sampling schemes previously used by ecologists; in addition to exact algorithms, even more efficient approximations are also presented.

In the article titled “I/O-Efficient Generation of Massive Graphs Following the *LFR* Benchmark,” Hamann et al. (<https://doi.org/10.1145/3230743>) present an I/O-efficient version of the Lancichinetti-Fortunato-Radicchi (LFR) benchmark graph generator. The LFR generator is used to evaluate community detection algorithms. Identifying meaningful clusters or communities in large complex networks, such as web graphs or social networks, is a common task in network analysis, and a large number of algorithms exist for the purpose. The LFR benchmark is the de facto standard for evaluating and comparing such algorithms. Though efficient clustering algorithms have already been designed for massive graphs that do not fit in the RAM of a single computer, no efficient generator of large LFR benchmark graphs has been known before for reliable evaluation of those algorithms. The Hamann et al. article fills this gap. The main contributions of the article are the I/O-efficient versions of the two most expensive parts of the LFR algorithm: (i) deterministically generating a graph satisfying a prescribed degree sequence, and (ii) randomizing that graph using random edge switches. The resulting external-memory LFR implementation is capable of handling graphs with more than 37 billion edges on a single desktop computer, is almost an order of magnitude faster than a massively parallel distributed-memory algorithm, and outperforms a state-of-the-art internal-memory implementation even on instances fitting in the main memory of a computer. The new implementation also runs significantly faster than the original LFR implementation while producing networks that are equivalent in terms of community detection algorithm performance and graph properties.

We thank the authors for carefully preparing extended versions of the papers and responding to reviewers comments. We are grateful to the reviewers for a thorough evaluation of the articles and helpful feedback, which helped the authors revise their articles. Finally, we would like to thank JEA and the Editor-in-Chief, Gonzalo Navarro, for making this volume possible, and for their continuous support and guidance.

Rezaul A. Chowdhury
Valentin Polischchuk
Vijaya Ramachandran

Guest Editors