Distributed Computing Column 72

Annual Review 2018

Jennifer L. Welch
Department of Computer Science and Engineering
Texas A&M University, College Station, TX 77843-3112, USA
welch@cse.tamu.edu

As with prior December issues, this issue is devoted to a review of notable events related to distributed computing that occurred during the year.

First, congratulations to Bowen Alpern and Fred Schneider, winners of the 2018 Edsger W. Dijkstra Prize in Distributed Computing for their paper "Defining Liveness"! Their paper appeared in *Information Processing Letters* in October 1985. The prize is jointly sponsored by ACM and EATCS, and is given alternately at PODC[1] and DISC[2]; this year it was given at PODC. This paper formally defined liveness properties of concurrent and distributed algorithms for the first time and also proved that every trace property is the conjunction of a safety property and a liveness property. The full citation can be found at `http://www.podc.org/dijkstra/2018-dijkstra-prize/`. I am delighted to include in this column the text of the remarks that Fred and Bowen gave at PODC when the award was presented to them.

Congratulations as well to Rati Gelashvili, who received the 2018 Principles of Distributed Computing Doctoral Dissertation Award! His thesis is entitled "On the Complexity of Synchronization" and was supervised by Professor Nir Shavit at the Massachusetts Institute of Technology. The award is jointly sponsored by PODC and DISC, and was given at DISC this year. The citation appears at `http://www.podc.org/dissertation/2018-dissertation-award/`. The thesis introduces a complexity-based hierarchy for concurrent objects based on combinations of weaker synchronization instructions rather than those considered in the classical consensus hierarchy. His new approach reflects the fact that actual multiprocessors let processes apply supported atomic instructions to arbitrary memory locations. The thesis also includes a linear-space lower bound for anonymous randomized consensus.

[1] ACM Symposium on Principles of Distributed Computing
[2] EATCS Symposium on Distributed Computing

Avery Miller has contributed a review of SIROCCO[3] 2018. The SIROCCO Prize for Innovation in Distributed Computing was given to Zvi Lotker for his work "in network algorithms, but especially for his creative contributions to the theory of wireless and social networks." The full laudatio can be found at `https://sites.google.com/view/sirocco2018/sirocco-prize?authuser=0`. Tal Navon received the best student paper award for her paper "Mixed Fault Tolerance in Server Assignment: Combining Reinforcement and Backup" coauthored with David Peleg. Congratulations to Zvi, Tal, and David!

Next, Naama Ben-David provides us with a review of PODC 2018. Best student paper awards were given to Guy Goren for his paper with Yoram Moses titled "Silence" and to Thibault Rieutord and Yuan He for their paper with Petr Kuznetsov titled "An Asynchronous Computability Theorem for Fair Adversaries". Leonid Barenboim, Michael Elkin and Uri Goldenberg received the best paper award for their paper titled "Locally-Iterative Distributed (Delta + 1)-Coloring below Szegedy-Vishwanathan Barrier, and Applications to Self-Stabilization and to Restricted-Bandwidth Models". Congratulations to Guy, Yoram, Thibault, Yuan, Petr, Leonid, Michael and Uri!

The column closes with a review of DISC 2018 by Aditya Biradavolu and Saptaparni Kumar. Best paper awards were given to Ali Mashreghi and Valerie King for their paper "Broadcast and Minimum Spanning Tree with $o(m)$ Messages in the Asynchronous CONGEST Model" and to Gregory Chockler and Alexey Gotsman for their paper "Multi-Shot Distributed Transaction Commit". Congratulations to Ali, Valerie, Gregory, and Alexey!

Many thanks to Bowen, Fred, Avery, Naama, Aditya and Saptaparni for their contributions!

Call for contributions: I welcome suggestions for material to include in this column, including news, reviews, open problems, tutorials and surveys, either exposing the community to new and interesting topics, or providing new insight on well-studied topics by organizing them in new ways.

[3]International Colloquium on Structural Information and Communication Complexity

History and Context for *Defining Liveness*[4], Winner 2018 Edsger W. Dijkstra Prize

Fred Schneider
Department of Computer and Information Sciences
Department of Computer Science, Gates Hall
Cornell University, Ithaca, New York, 14853
fbs@cs.cornell.edu

It's great to be back at PODC. I attended this conference religiously through 1992. It's interesting to see what has changed but also what has not changed in 25 years. And I can't think of a happier excuse to be back. All of us know how gratifying it is to see that our work is having impact, and that's what this award signifies.

To win an award that carries Dijkstra's name is especially meaningful for me. As a graduate student, I read and reread Dijkstra's work; it changed the way I looked at systems research and the research enterprise. (Bowen will offer some remarks that expand on this theme.) I then had a chance to meet Dijkstra when I served as teaching assistant for a 1-week short course he taught in Santa Cruz. That was the summer following my first year as an assistant professor at Cornell; my colleague David Gries had gotten me the assistantship. I guess I did OK, because Dijkstra and his wife Ria subsequently invited me to visit their home in Nuenen (The Netherlands) for a week. What a thrill. I subsequently saw Dijkstra at technical meetings and socially at least once a year, until he passed away. So, yes, I have many Dijkstra stories to tell. Catch me after dessert for those.

You can read our paper "Defining Liveness" (it's only 5 pages!) if you are interested in the technical details. I thought that instead I would talk about how we developed those ideas and where things stand today. I'm always fascinated by the history behind discoveries, and I suspect that I'm not alone in enjoying this "academic gossip".

The idea of proving programs was discussed by Turing in 1949 (where he proved that a subroutine for factorial computed the desired result).[1] Starting in the early 1960's through the late 1970's, the CS research community embraced the idea of writing proofs for programs. In those days, you would "prove the program correct". And there was a debate about whether to prove

[4] Bowen Alpern and Fred B. Schneider. Defining Liveness. *Information Processing Letters* 21, 4 (October 1985), 181–185.

[1] Turing, A.M. Checking a large routine. *Report of a Conference on High Speed Automatic Calculating Machines*, University Mathematics Laboratory, Cambridge, 67-69. For a discussion of the proof, see Morris, F.L., and C.B. Jones. An early program proof by Alan Turing. *Annals of the History of Computing* 6, 2 (April 1984), 139–143.

"partial correctness" versus "total correctness". Floyd's 1967 paper[2] had showed how to attach assertions to the edges of a flowchart; Hoare's 1969 paper[3] implemented this basic approach as a logic involving pre- and postconditions.

Lamport in a 1977 paper[4] showed how to "prove correctness" of multiprocess programs "to solve synchronization problems". He extended Floyd's work, introducing the terms "safety property" and "liveness property" to abstractly characterize different kinds of things you might want to prove. Synchronization protocols didn't produce answers, and termination was generally considered a failing, so proving partial or total correctness was not a useful goal. Petri nets were all the rage back then, as a specification language and as a way to simulate concurrent systems. The theory of Petri nets introduced the terms "liveness" and "boundedness" for describing how the assignment of a Petri net's "tokens" to its "places" could evolve; Petri net "safety" was a specific form of boundedness. Lamport borrowed those names for use in his verification work, but he gave the terms different meanings.

Fast forward to the mid 1980's. CS now had a formal methods research community that was devoted to program verification, and those researchers understood that the real problem was not verifying partial or total correctness but the more general problem of proving that a program satisfied a given specification. Some researchers were exploring temporal logics for this; others were exploring automata, because you could perform automated analysis with automata.

Bowen Alpern, who was my Ph.D. student at Cornell, was engaged in thesis research that had a foot in both camps. He understood that you could specify rich sets of program executions by using so-called Buchi automata (which were known to be models for temporal logic formulas). A Buchi automaton was a finite-state automaton that accepted infinite sequences. Specifically, it rejected sequences of input symbols that forced the automaton to make an undefined transition or that did not infinitely-often enter the automaton's accepting states. Not all input symbols had transitions defined in every automaton state, and not every automaton state would be an accepting state; one would formulate a Buchi automaton to accept exactly those program executions (modeled as infinite sequences of program states) that satisfied the property of interest.

But Bowen didn't pursue automated analysis for Buchi automata. Instead, he showed how to perform program verification by creating a correspondence between (i) program states and automaton states, and (ii) program transitions and automaton transitions. This correspondence was validated by discharging proof obligations that resembled the verification conditions you would have with Floyd's method or Hoare's logic.

First, you would show that the automaton wouldn't make an undefined transition when reading the state sequence corresponding to a program execution. The obligations here involved constructing invariants that related program states and automaton states. Second, you would show that the automaton could not remain forever in non-accepting states. These obligations involved exhibiting variant or well-founded functions, which decreased in value with each program step and evaluated to a minimal element when the next program step was guaranteed to cause transition into an accepting state.

As it happens, Leslie Lamport and I were both speakers at a 2-week NATO Advanced Course on Distributed Systems at Technical University of Munich in April 1984. Lamport had been using

[2]Floyd, R.W. Assigning meanings to programs. *Proc. Symposia in Applied Mathematics* 19, 1967, 19–31.

[3]Hoare, C.A.R. An axiomatic basis for computer programming. *CACM* 12, 10 (Oct. 1969), 576–580.

[4]Lamport, L. Proving the correctness of multiprocess programs. *IEEE Trans. on Software Engineering* SE-3, 2 (March 1977), 125–143.

a temporal logic for reasoning about concurrent programs, and he planned to talk about that work in Munich. (Ironically, my lectures in Munich concerned Lamport's state machine approach, and those presentations were the genesis for my well known survey paper on the subject.) It was natural in motivating the use of his temporal logic for Lamport to give a formal definition of safety properties, since that enabled him to argue that his temporal logic had sufficient expressive power. He shared with me this formal definition of safety properties. He also wanted to give an expressiveness argument for liveness properties, but he had been unable to devise a formal definition of liveness. He shared that challenge with me, too.

Safety properties assert that "bad things" don't happen. That can be formalized as eschewing irremediable finite prefixes of executions; the "bad thing" is thus formalized as a set of finite prefixes. When I told Bowen about this definition, he quickly saw the connection to his work on Buchi automata: The "bad prefixes" of safety properties were those prefixes that caused the Buchi automaton to make an undefined transition.

By returning to the full acceptance criteria for Buchi automata, a formal definition of liveness now became obvious. A liveness property had to be something that stipulated every prefix would have a continuation that caused acceptance by the Buchi automaton—that is, a continuation that infinitely-often would cause the automaton to enter accepting states. So the essence of liveness was: no matter what the finite prefix, an extension could be accepted by the Buchi automaton.

I told Lamport about this proposed formal definition for liveness. He, in turn, told Gordon Plotkin, who had been experimenting with using topology to reason about classes of properties for concurrent programs. Plotkin hated our liveness definition, and he mailed us a letter, giving a topological formulation (which he attributed to a paper by Mike Smyth[5]) to present his objections. That letter was 5 $\frac{1}{2}$ handwritten pages, and I still have a copy (albeit scanned).

Bowen and I were not convinced that the concerns Plotkin raised were reasons to abandon our proposed liveness definition. Moreover, Plotkin's letter explained that Lamport's formal definition of safety properties corresponded to the "closed sets" in a natural topology, and our liveness definition corresponded to the "dense sets". To somebody well versed in topology (which is not me), the fact that any set is the intersection of a closed set with a dense set follows trivially from the definitions of closed and dense sets. This meant that we not only had a liveness definition but we also had a proof that every property was the intersection (conjunction, when formalized in a logic) of a safety property and a liveness property! Such a decomposition result seemed to us to be a truly compelling rationale in support of our liveness definition.

So let's take stock of where we were.

Bowen and I now had formal definitions along with a proof that safety and liveness were an orthogonal basis for all "properties" (though I will return to the notion of properties shortly). Bowen also had shown that safety properties required invariance proofs whereas liveness properties required variant functions.[6] Thus, there was a benefit to performing that decomposition—it told you what proof technique to use for verifying each of the different pieces of an arbitrary property. Finally, we had also established that temporal logics were not required for proving arbitrary properties of concurrent programs; the kinds of proof obligations used in Floyd's original paper were sufficient. (Recall, showing that temporal logics were not needed was the original goal of Bowen's Ph.D. thesis. Those results can be found in a TOPLAS paper[7] that has mostly been ignored.)

---

[5]Michael B. Smyth. Power Domains and Predicate Transformers: A Topological View. *ICALP* 1983, 662–675.

[6]Bowen Alpern and Fred B. Schneider. Recognizing safety and liveness. *Distributed Computing* 2, 3 (1987), 117–126.

[7]Bowen Alpern and Fred B. Schneider. Verifying temporal properties without temporal logic. *TOPLAS* 11, 1

Now, let's skip ahead 30 years. Safety and liveness have decidedly entered into the vernacular—which is to say, they are used without citation, alas. But, after 30 years, it has also become clear that the definition we had been using for "property" was simplistic. The defining characteristic of a "property" (today, often called a "trace property") is a predicate that says whether each single execution in isolation is in that property. Yet many important aspects of system execution cannot be formalized in terms of such predicates: confidentiality, integrity, and service-level agreements, are examples. These are sets of executions that cannot be formalized as checks on an individual execution in isolation—they involve checks on pairs of executions or larger subsets. For example, confidentiality involves pairs of executions because it is a statement about correlation between the value of some variable and the value of some secret; checking for correlation requires looking at pairs of executions.

There's now some good news and some bad news. The good news: the safety/liveness orthogonal basis still works. Michael Clarkson and I introduced[8] the idea of "hyper-properties" as sets of sets of execution sequences, and we showed that all hyper-properties could be decomposed into hyper-safety (finite set of finite sequences) and hyper-liveness (set of infinite sequences can extend a set of prefixes). The bad news: "refinement" which was roughly "subset" for trace properties, becomes much more complicated for hyper-properties. Researchers are starting to develop proof methods for classes of hyper-properties, but nobody has identified a small set of building-block proof obligations like invariance and variant functions. So there is much work still to be done. And it is important work, because proving system security grows ever more crucial as we come to depend more and more on networked information systems.

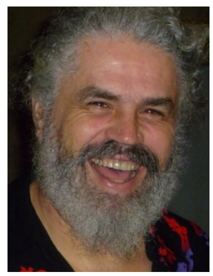
With that, you are up to date on safety/liveness. And now you also know why and how our Defining Liveness paper came into being. Again, thank you for this honor.

(January 1989), 147–167.

[8]Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *Journal of Computer Security* 18, 6 (September 2010), 1157–1210.

Edsger W. Dijkstra: The Man Behind the Prize

Bowen Alpern
Department of Computer Science
Lehman College, CUNY
New York, USA
bowen.alpern@lehman.cuny.edu



It would be understatement to say that news of this year's Dijkstra award came to me as a surprise. In fact, it put me into a rather profound funk, occasioning unsought reflection on my arguably misspent adulthood. It has been some time since I have done significant scientific research. So, I am quite humbled to be here.

I did not really know Dijkstra, but I certainly knew of him. With Fred and David Gries on the faculty, his influence on the Computer Science Department at Cornell was substantial when I was a graduate student. (I trust it still is.)

But I had actually encountered Dijkstra's shadow before coming to Cornell. He and I had overlapped briefly in the employ of Burroughs Corporation. He was their sole research fellow; I, part of a compiler implementation effort on a Cobol-like language for a check-processing machine.

Our paths did not cross, but I did stumble onto a stash of EWD reports. The EWD report is a unique literary form; roughly a cross between samizdat—the self-published manuscripts of Soviet dissidents—and contemporary blog posts, ranging in content from highly technical results to trip reports dripping with gossip. They are archived online at the University of Texas[1]. I commend them to your attention. I confess it was a quiet tirade about cheese that stayed with me over the years.

These past weeks, I have been rereading with some urgency these reports, his book (*A Discipline of Programming*), and his Turing Award lecture ("The Humble Programmer"). Mine was not the rigorous, depth-first close reading his work deserves, but a lazy, breadth-first perusal more congenial to my temperament.

Dijkstra felt it incumbent on the programmer to produce not only a running program but also a convincing argument that the program met its specification. He famously observed that no amount of testing could provide such an argument (EWD 361). The only thing that would satisfy was an "*elegant mathematical proof.*" As Fred indicated, my thesis tried to show how to adapt known techniques for proving correctness of sequential programs to prove properties of concurrent ones.

[1]http://www.cs.utexas.edu/ EWD/

But, Dijkstra wanted more from the programmer than facility with proof techniques: "*Besides a mathematical inclination, an exceptionally good mastery of [a natural language] is the most vital asset of a competent programmer.*" (EWD 498). "*A programmer must be able to express himself extremely well both in a natural language and in a formal system. The need for extreme mastery of a natural language is twofold. Firstly, …our natural language is so intimately tied with what we call understanding that we must be able to use it to express what we have understood.*" It is also "*an indispensable tool for thinking, in particular when new concepts have to be introduced. And this is what a programmer has to do all the time…in order to be able to find, to describe and to understand his own solution to [a] problem.*" (EWD 361).

As evidence of the crucial role that natural language plays in our field, it might be good here to remember some of the "*new concepts*" Dijkstra introduced (or helped to introduce) into the lexicon of "Computing Science": critical section, cooperating sequential processes, semaphores, deadly embrace, the Banker's algorithm (for deadlock avoidance), separation of concerns, predicate transformers, weakest preconditions, guarded commands, structured programming, self-stabilization, on-the-fly garbage collection, among others.

Notice that Dijkstra was very careful in his choice of metaphors. Sometimes they did not come easily. But he brought the same exacting discipline to his writing as he did to his programming. He wrote well!

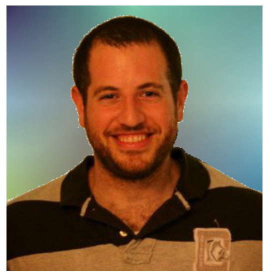
It occurs to me that this time I ultimately was reading his work as much for his style as for his content. The style mirrored the man: clear, concise, passionate, quirky, uncompromising, precise, scathing, brilliant, and wickedly funny. He had the audacity of his convictions and the integrity of Don Quixote. He abhorred the vague, the bloated, the ugly, and the sloppy. He did not suffer fools gladly.

The world needs more such voices.

Part of Dijkstra's genius lay in his ability to choreograph the dance between his imagery and his formalism. He demanded, for each of his metaphors, a precise definition that would render it amenable to rigorous formal manipulation. He delighted in the often unexpected insights that were offspring to this felicitous union of poetry and mathematics. I take deep satisfaction in receiving an award in his name for the work Fred and I did to provide such a definition for Leslie Lamport's liveness metaphor—a definition that illuminated the full mathematical grandeur of the dichotomy that Lamport had intuited between safety and liveness.

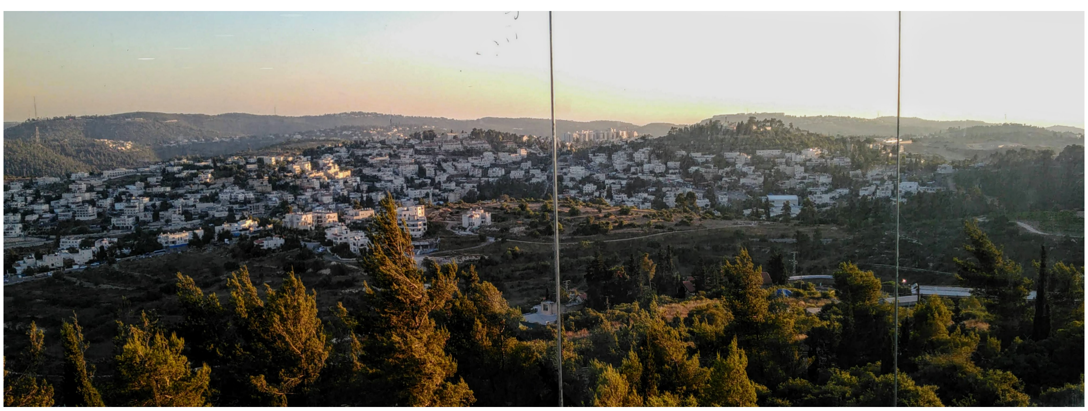
SIROCCO 2018 Review

Avery Miller
University of Manitoba
Winnipeg, Manitoba, Canada
avery.miller@umanitoba.ca



The International Colloquium on Structural Information and Communication Complexity (SIROCCO) is devoted to the study of decentralized systems consisting of multiple communicating entities, with a special focus on better understanding how knowledge and communication affect the feasibility and efficiency of solving tasks in such systems.

The conference turned 25 this year, and I am proud to say that I share its birthplace: Ottawa, Canada (unfortunately, 11-year-old Avery did not have a paper to submit to the first one). Another milestone is that this was the first SIROCCO to be held in Israel. The venue: a kibbutz called Ma'ale HaHamisha perched high up in the Judean Hills near Jerusalem.

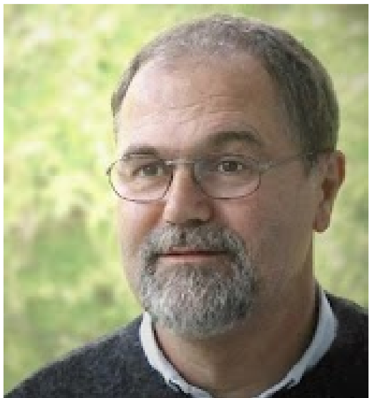


The sunset view in the photo above served as the backdrop for the opening reception, where, along with a buffet of Middle Eastern delicacies (the first of many), we were treated to two talks celebrating the SIROCCO Jubilee. The first was a historical retrospective, by Pierre Fraigniaud, that took us through a photographic chronology of the 24 previous SIROCCO venues and their participants. Although some of us have only recently started attending SIROCCO, it is clear that this is a close-knit and welcoming community that also happens to have fine taste in conference locales. The second presentation was a scientific retrospective, by David Peleg, that outlined several major lines of research whose humble origins can be traced back to a SIROCCO paper. This served as a reminder of the necessity of holding smaller conferences, where perhaps budding, incomplete, or "less popular" research ideas and results can be brought forward into the spotlight. The stage was now set and the bar was set high for SIROCCO #25, and it did not disappoint! This year's program featured four keynote speakers, two talks given by the winners of the SIROCCO Prize for Innovation in Distributed Computing and the Best Student Paper award, twenty-three presentations of full papers, and eight brief announcements. What follows is a summary of the program, and then a short description of our half-day excursion to the Old City of Jerusalem.

The Keynote Talks

David Peleg: "Realizability of Graph Specifications"
Given a graph G, one can associate a vector $\langle d_1, \ldots, d_n \rangle$ where each d_i represents the degree of vertex i. The dual problem, studied by Erdős and Gallai [6], is to characterize, for a given a vector $\langle d_1, \ldots, d_n \rangle$, when there is a graph that *realizes* this degree sequence. In this talk, the problem was generalized to any given type of graph profile that can be specified by an integer sequence. Two central questions are: characterize when a sequence is realizable, and, devise an algorithm that will produce a realizing graph for a given sequence. Several solved examples were given, however, the presentation asked more questions than it answered, practically inviting us to get started immediately on exploring these new avenues for research.

Seth Pettie: "The Distributed Lovász Local Lemma Problem"
The Lovász Local Lemma (LLL) is a common tool used in the probabilistic method to guarantee the existence of certain combinatorial objects. This can be used to show that good solutions to certain problems exist, but it is unsatisfying to designers of practical algorithms. This talk discusses the *Algorithmic Distributed LLL* problem: how to construct the combinatorial objects guaranteed by the LLL in a distributed way in the LOCAL model. The talk presented several interesting results: the problem demonstrates an exponential gap between the deterministic and random LOCAL models; randomized Distributed LLL is complete for sublogarithmic randomized time; and, the complexity of deterministic Distributed LLL is inextricably linked to deterministically computing network decompositions.



Kurt Mehlhorn: "On Fair Division for Indivisible Goods"
Consider a collection of items: each item cannot be divided into smaller pieces, however there is a known number of copies of each item. The items must be allocated to a set of agents, who each have some personal opinions about how valuable each item is (with the condition that duplicate copies of an item are not as valuable). The goal is to maximize the fairness of the allocation, as measured by Nash social welfare. The problem is known to be NP-complete and APX-hard. This talk summarized some previous approaches based on Fisher markets that achieved constant-factor approximation algorithms in the case of additive valuations, and presented a new price-envy-free approach [5] that achieves a better approximation ratio under more general kinds of valuations.

Claire Mathieu: "College Admissions in Practice"
Students have preferences about which college they want to attend, and colleges have preferences about which students they want to admit. The *stable marriage* problem involves taking the rankings by all parties and partnering them up such that there is no pair (A, B) where A and B would both prefer to be matched together instead of their current partner. The Gale-Shapley algorithm is the standard solution, but in the case of college admissions in France, certain additional constraints were imposed. The talk discussed these additional constraints and how an adapted version of Gale-Shapley handled them. As many members of the audience had interest and experience in admission processes, a lively discussion ensued!

The 2018 SIROCCO Prize for Innovation in Distributed Computing

Zvi Lotker: "Taking Turing to the Theater"
Summarizing a story and identifying its critical moments: this is not an easy task, even for humans. It seems to require an understanding of the complete text, necessary background information, the narrative style, the character development, and more. In this talk, Zvi introduced us to his M algorithm, which is able to pinpoint critical events in scripts (e.g., for films or theater productions) and produce an executive summary consisting of a montage of these critical events. Surprisingly, the algorithm only uses structural elements of the text and how they relate to time, i.e., it does not actually "understand" a single word. He showed us a few motivating examples produced by his algorithm, demonstrated how to carry out the calculations ourselves, and encouraged us to do the same analysis on our favourite movies. Based on the discussion that arose, it seemed that many of us were ready to dive in and explore what the algorithm can achieve and how it can be applied.

The Best Student Paper Award

Tal Navon, David Peleg: "Mixed Fault Tolerance in Server Assignment: Combining Reinforcement and Backup"

In client-server systems where up to f servers might fail, one approach to fault tolerance is to introduce redundant servers. Another method is to "reinforce" some existing servers, at a high cost, but making them less likely to fail. This paper considers what can be achieved using a mixed strategy. The authors studied a variety of server placement tasks in graphs, including dominating set, k-centers, and uncapacited facility location. In each case, they showed how to augment known approximation algorithms that use only redundancy, creating good approximation algorithms that use both redundancy and reinforcement.

The Contributed Talks

Graph Algorithms

Simple and Local Independent Set Approximation

Ravi B. Boppana, Magnus M. Halldorsson, Dror Rawitz

This paper analyzes a simple 1-round distributed approximation algorithm for independent set, due to Boppana [1, 4]: each vertex v picks a random real number x_v from $[0,1]$, and joins the independent set if its random number is larger than that of its neighbours. The authors show that this algorithm gives a tight $(\Delta+1)/2$-approximation in unweighted graphs of maximum degree Δ, which is the best possible for 1-round distributed algorithms. For weighted graphs, they provide a simple modification that results in an asymptotic expected $0.529(\Delta+1)$-approximation, which improves on the $(\Delta+1)$ approximation factor of the original version.

Deterministic Distributed Ruling Sets of Line Graphs.

Fabian Kuhn, Yannic Maus, Simon Weidner

An (α,β)-*ruling set* of a graph G is a subset R of vertices such that the distance between any two vertices of R is at most α, and every vertex of G has distance at most β to a vertex of R. Such sets have been used as a powerful tool in distributed graph algorithms, e.g., for network decompositions, coloring, MIS, and shortest paths. This paper focuses on deterministic algorithms for computing ruling sets in the CONGEST model. Further, they define a (α,β)-*ruling edge set* as a subset F of edges that form an (α,β)-ruling set in the line graph of G, and show that these can be computed particularly efficiently.

A Distributed Algorithm for Finding Hamiltonian Cycles in Random Graphs in $O(\log n)$ Time.

Volker Turau

This paper considers distributed algorithms for finding a Hamiltonian cycle in a random graph $G(n,p)$ where $p \geq (\log n)^{3/2}/\sqrt{n}$. They provide a randomized algorithm that uses $O(\log n)$ rounds in the synchronous message passing model with messages of size $O(\log n)$ and $O(\log n)$ memory per node. The probability of success approaches 1 as n tends to infinity. The algorithm improves on the previous best known approach, which worked for $p = c\log n/n^{\delta}$ $(0 < \delta \leq 1)$ and used $\tilde{O}(n^{\delta})$ rounds.

A Deterministic Distributed 2-Approximation for Weighted Vertex Cover in $O(\log n \log \Delta/\log^2 \log \Delta)$ Rounds.

Ran Ben Basat, Guy Even, Ken-Ichi Kawarabayashi, Gregory Schwartzman

This paper considers the Minimum Weight Vertex Cover problem in the CONGEST model. The main result is a deterministic distributed $(2+\epsilon)$-approximation algorithm in which the number of rounds is bounded by $O(\frac{\log\Delta}{\log\log\Delta} + \frac{\log\epsilon^{-1}\log\Delta}{\log^2\log\Delta})$ when the nodes know Δ (although they also consider the case where Δ is unknown). Their algorithm builds upon the BCS algorithm [3], and improves the linear dependence on ϵ^{-1} to logarithmic. Moreover, their algorithm is optimal if $\epsilon^{-1} = (\log\Delta)^c$. Under the assumption that the maximum node weight is polynomial in the size of the network, their algorithm is a 2-approximation, and improves over the previous best known deterministic 2-approximation algorithm for the problem.

Online Service with Delay on a Line.
Marcin Bienkowski, Artur Kraska, Paweł Schmidt

Consider n equidistant points on a line. At various times, "requests" are made at the points, and a "server" can travel to the points to serve the requests. A helpful analogy was given in the talk: one can think of a plumber receiving calls from various houses to come fix a flooding toilet. The cost of serving a request is the distance traveled to the point, plus a "waiting" penalty that is a function of the arrival delay, e.g., the flooding got much worse because the plumber took a long time to arrive, which increases the time needed to fix it. The goal is to minimize the sum of the costs to serve all of the requests. In this paper, the authors present a deterministic $O(\log n)$-competitive algorithm for this problem. The high-level intuition is that, for segments of the line that are far away from the server, we should allow many requests to accumulate before incurring the high travel cost.

Fundamentals

Connectivity and Minimum Cut Approximation in the Broadcast Congested Clique.
Krzysztof Nowicki, Tomasz Jurdzinski

In the Broadcast Congested Clique model, there are n nodes that communicate in synchronous rounds, and, in each round, each node can transmit a single $O(\log n)$-bit message to all other nodes in the network. The nodes belong to some input graph (not necessarily a clique) and the goal is for the nodes to compute some function of this input graph. The authors present a deterministic algorithm that finds a maximal spanning forest using $O(\log n/ \log\log n)$ rounds. They also present a randomized $(1 + \epsilon)$-approximation algorithm that finds a minimum cut using $O(\log n)$ spanning forest computations (i.e., $O(\log^2 n/ \log\log n)$ total rounds). They also show how their min-cut algorithm can be used in a multi-pass semi-streaming model.

Communication Complexity in Vertex Partition Whiteboard Model.
Tomasz Jurdzinski, Krzysztof Lorys, Krzysztof Nowicki

This paper considers a k-player communication model where computation happens in synchronous rounds. At the end of each round, each player calculates a message of size b, and they all simultaneously send their messages to a central referee (a.k.a, whiteboard). At the start of any round after the first, the referee shares all messages with all players, who can use this information when computing what to send next. After r rounds, the players must stop sending messages, and the referee computes an output based on all of the messages it has received. The types of problems considered in this paper are questions about an input graph G. The vertices of G are partitioned in k sets, each player receives one of the sets as input, and the players run a protocol such that the referee can determine a property about G, e.g., is G connected? The authors show that, for any fixed constant b, there is an infinite strict hierarchy of problems for an increasing sequence of values for r (the number of rounds). Further, for deciding connectivity in two-regular input graphs G, they prove a matching upper and lower bound of $b \in \Theta(\log n)$ for the class of 1-round protocols, and show that no protocols exist with $r, b \in O(1)$ simultaneously.

Two Rounds Are Enough for Reconstructing Any Graph (Class) In The Congested Clique Model.
Pedro Montealegre, Sebastian Perez-Salazar, Ivan Rapaport, Ioan Todinca

In the congested clique model, communication proceeds in rounds: in each round, each node v can send a different $O(\log n)$-bit message to each node w. The authors consider the *reconstruction problem* for a graph class $\mathcal{G}$: a graph G with n vertices is provided to the nodes by assigning to each node v an n-bit vector indicating which vertices of G are neighbours of v. At termination, if $G \in \mathcal{G}$, all nodes v must output the entire graph G, and if $G \notin \mathcal{G}$, all nodes must reject. A weaker version of the problem promises that the input graph G belongs to $\mathcal{G}$, and nodes must output G. The paper gives a one-round private-coin randomized algorithm that solves the strong version w.h.p. when $\mathcal{G}$ is a hereditary class. For general graph classes, they provide a two-round private-coin randomized algorithm that solves the strong version w.h.p., a three-round deterministic algorithm that solves the strong version, and a two-round deterministic algorithm that solves the weak version. The main technique is to use “fingerprinting” and “error-correcting graphs” to essentially compress large amounts of graph information while still being able to distinguish which encoded objects represent graphs in $\mathcal{G}$. It follows that the feasibility of their technique depends only on the size of the graph class, i.e., the amount of information that needs to be compressed. One interesting open question is whether “recognition” (simply output 1 if $G \in \mathcal{G}$) is any easier than full reconstruction.

Time-Bounded Influence Diffusion with Incentives.
Gennaro Cordasco, Luisa Gargano, Joseph Peters, Adele Rescigno, Ugo Vaccaro.

This paper studies “the spread of influence” in a social network. A network is modeled by a graph G, and each node v has an associated positive integer $t(v)$ called the *influence threshold*. Initially, some set $\mathcal{I}$ of nodes is considered *influenced*, and then we proceed in rounds: in round k, a node v joins $\mathcal{I}$ if at least $t(v)$ of its neighbours were in $\mathcal{I}$ before round k. The general problem is to find a small initial set so that eventually all nodes join $\mathcal{I}$. This paper extends the model and problem in two ways: first, it considers the time it takes for the entire network to be influenced, and it also adds the possibility of initially applying *incentives* that lower the threshold value $t(v)$ at some nodes (with the hope that this lowers the total influence time). Setting an incentive equal to $t(v)$ means that the node is initially influenced. The problem they consider takes the graph G, thresholds t, and time bound λ, and a solution must find a minimum choice of incentives that need to be applied so that all nodes in G are influenced within λ rounds. They provide a linear-time greedy algorithm for paths, and polynomial-time algorithms for trees and complete networks.

Brief Announcement: A Self-Stabilizing Algorithm for Maximal Matching in Link-Register Model.
George Manoussakis, Johanne Cohen, Laurence Pilard, Devan Sohier

Consider a graph G of n processes. For each edge $\{u, v\}$, there is a shared register r_{uv} which only node v can read and to which only node u can write, and similarly there is also a shared register r_{vu}. At each computation step, each node can perform a single read operation or a single write operation. In this model, the paper presents a distributed self-stabilizing algorithm that solves maximal matching under fair distributed adversaries, i.e., those that guarantees that every process that wants to take steps is eventually scheduled. The algorithm stabilizes in $O(m\Delta)$ steps, where m is the number of graph edges and Δ is the maximum degree. As an intermediate step, they also solve maximal matching under unfair distributed adversaries but under the additional assumption that each node u can also read r_{uv}.

Brief Announcement: Message-Efficient Self-stabilizing Transformer Using Snap-Stabilizing Quiescence Detection.
Anas Durand, Shay Kutten

A *diffusing computation* is an algorithm where a unique *initiator* spontaneously sends a message exactly once (to one or more of its neighbours) and all other processes can start sending messages (as often as they would like) after receiving at least one message. Examples include performing a one-to-all broadcast, a BFS, a DFS, etc. Such a computation reaches *quiescence* when no messages are in the communication links and a local indicator of stability holds at every process. Examples of quiescence are termination and deadlock. The ability to detect quiescence can be very useful, e.g., to reset a deadlock, to start another operation after termination, or as a building block in building a *transformer* that can convert non-self-stabilizing algorithms into self-stabilizing ones. This paper proposes a self-stabilizing quiescence detection algorithm for diffusing computations, and uses it to implement a message-efficient self-stabilizing transformer.

Brief Announcement: Constant-Space Self-Stabilizing Token Distribution in Trees.
Yuichi Sudo, Ajoy K. Datta, Lawrence Larmore, Toshimitsu Masuzawa

Consider a rooted n-node tree, a given positive integer k, and a known node capacity $\ell \geq k$. The goal is to distribute tokens so that each node has exactly k tokens, starting from any configuration in which some nodes might already have some tokens. The number of tokens in the initial configuration might not be exactly nk, so the designated root node is allowed to push out tokens or pull new tokens in from an external source. The tokens are passed between nodes using link registers: for each edge $\{u, v\}$, there is a register r_{uv} that both u, v can read and to which only u can write, and similarly there is a register r_{vu}. The paper proposes three algorithms for this task, and evaluates them by analyzing their time complexity, space complexity, and total number of token moves.

Mobile Agents

Explorable Families of Graphs.
Andrzej Pelc

An agent is located at a node v of a finite simple connected undirected graph G. It has no *a priori* information about G. The nodes of G are not labeled, however the agent can distinguish between the incident edges of a node as they are labeled with distinct port numbers. In one step, the agent chooses a port to exit the node via an edge, it arrives at the other endpoint of the edge, and learns the port number from which it entered. The agent's goal is to eventually visit all nodes of G using a deterministic algorithm. It is known that this task is impossible in arbitrary graphs, so this paper studies which graph families are *explorable*. More specifically, if the agent is promised that G is from a given graph family $\mathcal{F}$, is there an algorithm that will solve exploration? The paper gives an exact characterization of which graph families $\mathcal{F}$ are explorable, and in the positive case, constructs a deterministic exploration algorithm that works specifically for $\mathcal{F}$. The author also considers the possibility of a universal algorithm that takes an arbitrary explorable family $\mathcal{F}$ as input and can explore every graph in $\mathcal{F}$. The answer depends on how $\mathcal{F}$ is provided: if there is an oracle that can answer every yes/no question about $\mathcal{F}$, then there is a universal algorithm for exploration. However, if only given an enumerator that can provide any requested graph from $\mathcal{F}$, there is no universal algorithm.

Space-efficient Uniform Deployment of Mobile Agents in Asynchronous Unidirectional Ring.
Masahiro Shibata, Hirotsugu Kakugawa, Toshimitsu Masuzawa

Consider k asynchronous mobile agents located at arbitrary nodes of an unlabeled n-node unidirectional ring network. The value of k is known by the agents, and an agent can only observe the node at which it is currently located. Agents cannot directly communicate, but they each have a token that they can place at a node they are visiting and cannot pick it up again. This paper considers the problem of uniform deployment: at termination, the distance between every pair of two consecutive agents is the same. They give memory-efficient algorithms for solving the problem under two different assumptions: *no multiplicity detection*, which means that an agent cannot detect when it is at the same node as another agent, and *weak multiplicity detection*, which means that an agent knows if it is at a node containing at least one more agent, but not the exact number of agents.

Priority Evacuation from a Disk Using Mobile Robots.
Ryan Killick, Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, Sunil Shende

This paper introduces a *priority evacuation* problem where $n+1$ agents (n servants and one 'queen') are located on a disk and must find a hidden exit on the perimeter of the disk through which the queen must escape. All agents initially start at the center of the disk. The exit is found as soon as an agent occupies the exit's position, and then this agent instaneously broadcasts its coordinates to all others. The goal is to minimize the time it takes for the queen to reach the exit (the other n agents are expendable and can be sacrificed to whatever the queen is escaping from). For the case $n \geq 4$, the authors propose an algorithm that guarantees that the queen escapes in time less than $2 + 4(\sqrt{2}-1)\frac{\pi}{n}$, and they prove that the queen cannot be evacuated in time less than

$2 + \frac{\pi}{n} + \frac{2}{n^2}$ (they conjecture that their upper bound is in fact optimal). Many in the audience at the talk felt that the proposed search strategy was quite nice, especially when presented beautifully as an animation in Ryan's talk slides!

Gathering in the Plane of Location-Aware Robots in the Presence of Spies.
Ryan Killick, Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Oscar Morales Ponce

Consider a set of $n > 2$ mobile robots in the Cartesian plane, where at most $F \leq n-2$ of the robots are Byzantine and indistinguishable from the non-faulty robots. The robots have GPS devices so that they accurately know their position at all times. There is a central authority to which all robots can report their position. The goal is for the central authority to send trajectory instructions to the robots so that all of the non-faulty robots eventually meet at the same point. The Byzantine robots are trying to delay this from happening, either by falsely reporting their location, failing to report, or failing to follow given trajectory instructions. The performance of an algorithm is measured by the competitive ratio of the algorithm's gathering time versus the time that could be achieved if the central authority knew which robots were Byzantine. The authors provide efficient algorithms in the case where the central authority knows an upper bound on the number of Byzantine robots. In the case of at most one faulty robot, they give an optimal algorithm, and in the case where the proportion of Byzantine robots is less than one half or one third, they give algorithms with small constant competitive ratios. For an arbitrary number of faulty robots, they give algorithms whose competitive ratio is bounded above by $\min\{32\sqrt{2}, F + 2\}$.

Symmetric Rendezvous With Advice: How to Rendezvous in a Disk.
Konstantinos Georgiou, Jay Griffiths, Yuval Yakubov

In the Symmetric Rendezvous Problem in a Disk, two agents are initially located at points in the plane, the initial distance d between them is known, and they both know the location of the origin O. They both move at the same speed, can move in any direction, and both run the same randomized and synchronized algorithm. They cannot see each other, but can detect when they are located at the exact same point. The goal is to design the algorithm as to minimize the expected time that elapses before they first meet. The paper also considers the worst-case performance of algorithms, called *energy*, which is defined as the amount of time that elapses such that rendezvous occurs with probability 1. The main contribution is to show how known (sub-optimal) algorithms for the model where O is not known by the agents can be improved by providing knowledge of O. Additionally, they show that this additional knowledge of O can be used to guarantee a finite upper bound on energy.

Brief Announcement: Distributed Counting along Lossy Paths without Feedback.
Vitalii Demianiuk, Sergey Gorinsky, Sergey Nikolenko, Kirill Kogan

In an asynchronous packet network, a set f of packets originating at a source switch S is destined for a destination switch D. For accounting and management purposes, it is important to be able to calculate $|f|$. This paper proposes a distributed counter that maintains some bits at both S and D. This is motivated by the fact that, as the number of parallel flows and as the network traffic rate increases, a single network switch does not have the memory or processing capacity to keep up. One complication that the authors include in their model (and deal with in their solution) is the realistic assumption that packets might be reordered or lost along the way from S to D.

Coordination

On the Strongest Message Adversary for Consensus in Directed Dynamic Networks.
Ulrich Schmid, Manfred Schwarz

> Consider a synchronous distributed directed message-passing system consisting of an unknown number n of processes that never fail. A *message adversary* controls the ability to communicate: in each round, the adversary chooses which communication arcs are present in the system, and thus has the ability to "suppress" communication from p to q in a given round by removing arc (p, q). A run of the system is an infinite sequence of communication graphs $G^1, G^2, \ldots$ where G^i specifies the communication edges available in round i. A message adversary is defined by which such graph sequences it may generate. If given too much power, a message adversary can certainly prevent consensus from happening (e.g., just make all graphs in the sequence empty), and when very restricted (e.g., each element of the sequence must be the complete graph), consensus is always possible. This paper characterizes *strongest message adversaries* for consensus, which turn out to be those that include all possible infinite sequences of identical star graphs.

A Characterization of t-Resilient Solvable Colorless Tasks in Anonymous Shared-Memory Model.
Carole Delporte-Gallet, Hugues Fauconnier, Sergio Rajsbaum, Nayuta Yanagisawa

> This paper considers the asynchronous shared memory model where n anonymous processes communicate through multi-writer/multi-reader registers. They provide a characterization of which colorless tasks are solvable when at most t processes may crash: a colorless task is t-resilient solvable anonymously if and only if it is t-resilient solvable non-anonymously. To prove the result, they first design an anonymous non-blocking implementation of an atomic weak set object with n registers, which they use to build a wait-free implementation of a safe agreement object for an arbitrary set V of values. Then, they describe two ways to prove the characterization: one way is through a new anonymous implementation of the BG-simulation, and the other way is to solve k-set agreement and then apply a topological argument.

Crash-tolerant Consensus in Directed Graph Revisited.
Ashish Choudhury, Gayathri Garimella, Arpita Patra, Divya Ravi, Pratik Sarkar

> This paper considers distributed consensus in directed graphs in the synchronous message-passing model where up to f nodes might crash at any time. Previous work [8] presented necessary and sufficient conditions for the existence of consensus protocols in this model. This paper modifies the min-max-based protocols from [8] in order to improve the round and communication complexity. The authors also provide new lower bounds for the class of min-max-based consensus protocols in this model.

Balanced allocations and global clock in population protocols: An accurate analysis.
Yves Mocquard, Bruno Sericola, Emmanuelle Anceaume

Consider a set of n bins, and consider the following random process: at each discrete step, randomly choose two bins, and place a ball in the least filled bin of the two. This is known as the two-choice paradigm for online load balancing, which has previously been shown [2] to achieve a drastically more balanced load compared to simply placing the ball in one randomly chosen bin. At any time t, let $Gap(t)$ denote the difference between the most loaded and least loaded of the n bins. This paper aims to derive good asymptotic approximations of $Gap(t)$ for large values of n. They show that, for all $t \geq 0, n \geq 2$ and $\sigma > 0$, the value of $Gap(t)$ is less than $a(1+\sigma)\ln(n)+b$ with probability greater than $1 - 1/n^{\sigma}$ where the constants a and b are optimized and given explicitly. Their approach is to analyze a population protocol that implements the process: at each time step, the two interacting agents represent the two chosen bins, and the agent with the smaller local counter increments its local counter. An additional feature of this protocol is that, by accurately estimating the maximal gap between any two counters, other protocols can use this to implement a global clock.

On Knowledge and Communication Complexity in Distributed Systems.
Daniel Pfleger, Ulrich Schmid

This paper explores the possibility of proving bounds on the communication complexity of a problem $\mathcal{P}$ by using the epistemic knowledge that must necessarily be attained by the processes in order to solve $\mathcal{P}$. Epistemic logic can be used to formally reason about the knowledge and beliefs of the processes, and in particular is useful for defining which states are indistinguishable from one another. In addition, *Action Models* can be used to describe the possible communication events that may occur at certain times in an algorithm's execution. A synchronous algorithm can then be viewed as an alternating sequence between "knowledge models" and "action models", where the current knowledge model defines the next action model which defines the next knowledge model, and so on. An action model at time t can be partitioned from the point of view of a particular process p: the possible actions that might occur at time t that are indistinguishable to process p belong to the same class in the partition. The authors argue that there is a strong connection between the communication complexity (i.e., the number of bits received by p) and the number of sets into which the action model is partitioned with respect to process p's point of view. Essentially, the number of bits p receives is directly related to the number of different situations that process p can distinguish between. The authors apply this idea to prove two (already known) lower bounds: one for a distributed function computation in the 2-player communication model, and one for consensus in directed dynamic networks controlled by a message adversary.

Brief Announcement: Make&Activate-Before-Break: Policy Preserving Seamless Routes Replacement in SDN.
Yefim Dinitz, Shlomi Dolev, Daniel Khankin

This paper considers the problem of seamless route replacement in SDNs: a centralized algorithm updates the routing rules at the network switches in order to change the set of routes, but must be done live in a seamless way, i.e., that doesn't cause problems for packets already in transit. Changing several routes at a time can get complicated, as there can be various interdependencies between sub-routes. The first contribution of this work proposes a solution to an extended version of the problem that additionally requires that network policies are maintained while routes are being updated. There are two cases based on whether or not policies are allowed to be migrated to different nodes to make the route updates easier to perform. The second contribution proposed by this paper is the implementation of a *Route Readiness Verifier* that can check when the "update" commands sent by the central controller to the network switches are actually executed. This would enable the implementation of the *Make&Activate-Before-Break* approach to route replacements, e.g., wait until the new route is ready before using it. This would close a significant restrictive gap in the OpenFlow standard.

Networks

Biased Clocks: A Novel Approach to Improve the Ability to Perform Predicate Detection with O(1) Clocks.
Vidhya Tekken Valapil, Sandeep Kulkarni

This paper introduces a version of hybrid logical clocks that adds bias B to the timestamps of received messages. In particular, each sent message contains the sender's local timestamp for the send event, and when a process j calculates the timestamp of a receive event, it takes the maximum of $\{j$'s physical clock value, $1+$ j's previous logical clock value, and $B+$ the timestamp included in the received message$\}$. Previous hybrid logical clocks [7] had the same implementation but with $B = 1$. The effect of adding bias is that it introduces more opportunities to detect global states consisting of concurrent events, since now two events are considered concurrent if their timestamp values differ by less than B (instead of requiring equality). The authors conduct experiments to show that taking $B > 1$ is more effective than the previous version of hybrid logical clocks that sets $B = 1$.

Formalizing Compute-Aggregate Problems in Cloud Computing.
Pavel Chuprikov, Alex Davydow, Kirill Kogan, Sergey Nikolenko, Alexander Sirotkin

Consider an undirected connected network G such that each edge is labeled with the cost of moving one unit of information across it. Each node of the network starts with a certain number of units of data, and the goal is to aggregate the data and supply the answer to a target node t. In a simple scenario where all aggregation must happen at node t, the goal is to find the best paths along which to send the data to minimize the total cost. However, more realistically, aggregation can occur at any node in the network, so there is an additional function μ that, given two pieces of data, tells us the size after aggregating the two pieces. This paper considers the Compute-Aggregate Minimization (CAM) problem: given an undirected connected graph G, an edge-cost function c, a target vertex t, a set of initial data chunks C with sizes and initial locations, and an aggregation size function μ, find an aggregation plan such that the total cost of moving data across edges is minimized. The paper also considers two special cases: when G is a tree (TCAM) and when aggregation can only happen at nodes that were initially assigned chunks (CCAM). They give polynomial-time approximation algorithms for general μ, although the approximation factor cannot be constant unless $P = NP$. They also give approximation algorithms and hardness results for aggregations size functions with restricted ranges, and also for specific examples of μ such as min, max, and $+$.

Broadcast with Energy-Exchanging Mobile Agents Distributed on a Tree.
Jurek Czyzowicz, Krzysztof Diks, Jean Moussi, Wojciech Rytter

Consider an rooted edge-weighted tree with mobile agents arbitrarily deployed at some of the nodes. Each agent possesses some initial amount of energy. The agents can move through the tree from node to node, however each move incurs a cost that gets deducted from their energy tank. When two nodes meet at a node, they may transfer any amount of energy that they currently possess. The problem considered in this paper assumes that a token is initially located at the root of the tree, and this token can be carried by any agent and exchanged between agents. The goal is to determine whether or not there is a schedule of agent movements and energy transfers such that the token eventually visits every node of the network. The authors solve the problem by providing a full-knowledge centralized algorithm that controls the movements of the agents. Their approach also solves a more general optimization problem: if the agents can permanently deposit unused energy at the root, determine the maximum amount that can be deposited while ensuring that the token visits all nodes.

Brief Announcement: Fast Approximate Counting and Leader Election in Populations.
Othon Michail, Paul Spirakis, Michail Theofilatos

Consider a population of n distributed and anonymous agents. In every discrete time step, a uniform random scheduler chooses two agents that will "interact", which means that a transition function is applied to their current pair of states to determine their new states. The time complexity of solving a task is the number of interactions until stabilization divided by n (called parallel time). This paper solves two problems: they provide an algorithm that, w.h.p., computes an upper bound on n that is at most n^a for some constant $a > 1$, assuming that a unique leader is already chosen; also, they provide an algorithm that solves leader election w.h.p. assuming that the nodes know an upper bound $n^b, b > 1$ on n. The first algorithm stabilizes in $\Theta(\log n)$ parallel time and all nodes except the leader use a constant number of states (the leader uses $\Theta(\log^2 n)$ states). The second algorithm has a smooth time-space tradeoff: on one end is $O(\log^2 n)$ parallel time and $O(\log n)$ states, and on the other end is $O(\log n)$ parallel time and $O(n)$ states.

Brief Announcement: One-Max Constant-Probability Networks: Results and Future Work.
Mark Korenblit

This paper summarizes new tree-like models for randomized network growth/generation that extend from the well-known Barabási-Albert random graph model for scale-free networks. In the *one-max constrant-probability models*, a newly-added vertex can be connected to at most one old vertex, any connection event is realized with the same probability p, and the probability that it will be connected to a particular vertex i depends on the rank of i's degree in relation to all other vertex degrees. This models real-life choices that depend on relative status rather than an absolute characteristic. The *constant-probability ordered model* keeps the vertices sorted in decreasing order of their degrees, although leads to many isolated vertices. This is addressed in the *constant-probability ordered non-0 model* which prioritizes connections to isolated vertices over newly-added vertices of degree 1. The paper suggests several directions for future research about network topology evolution based on extensions of these tree-like models.

Brief Announcement: Reaching Distributed Equilibrium with Limited ID Space.
Dor Bank, Moshe Sulamy, Eyal Waserman

Consider the synchronous message-passing model with n agents such that the network topology is 2-vertex-connected. The agents are running an algorithm $\mathcal{A}$, and each agent has a utility function defined over all possible algorithm inputs. All utility functions satisfy the property that an agent never prefers an outcome in which the algorithm fails over one in which it terminates correctly (this distinguishes the model from ones that include Byzantine faults). An algorithm is in *equilibrium* if no agent can unilaterally increase its utility by deviating from the algorithm. The value of n is not known by the agents, which means that an agent may try to increase its utility by duplicating itself (i.e., simulating imaginary agents and acting on their behalf). However, there is a known restricted range of agent ID's $\{1, \ldots, L\}$, so there is a risk that a duplicate agent ID is discovered and the algorithm fails. So, if the agents are initially given a large enough lower bound t such that $t \leq n \leq L$, their incentive to cheat will go down and the algorithm will be in equilibrium. This paper provides a method for calculating the minimal such value t and they apply the method to Leader Election and Knowledge Sharing algorithms.

Excursion to the Old City

We had three scheduled events on our excursion to the Old City. The first was a guided tour of the city and the tunnels under the Western Wall. Along with the standard stops at places like the Church of the Holy Sepulchre, we were led up a staircase to a rooftop that gave a beautiful view of the Temple Mount, the Mount of Olives, and the rest of the Old City vista. We then visited the Western Wall. The tunnels under the wall were interesting, but not interesting enough to see twice, which we did anyway because we needed to retrace our steps all the way back to the entrance (the exit at the end was blocked). This delayed our arrival at the second scheduled stop: dinner.

Eucalyptus Restaurant is located just outside of the walls of the Old City. We were treated to many small courses of traditional fare, including some perennial favourites of mine: lamb siniya, anything with eggplant, and makloubeh, which consists of rice, vegetables and meat prepared in a large pot and flipped over quickly and courageously before serving. The conference awards were presented after dinner, but the earlier delay meant that we needed to rush out to get to our third stop: the multimedia Tower of David Night Spectacular that would have no issue starting without us.

Thankfully we made it with time to spare, I suspect because whoever designed our schedule anticipated that we would be delayed. It was a beautiful show: the story of Jerusalem starting from biblical times, but purely through music and projected animations on the old walls of the Citadel's courtyard. But after a half-day of talks and a 7+ hour excursion, all of us were definitely ready to call it a night.

And on that fitting note, I'll conclude my report on SIROCCO 2018. I hope to see you at next year's conference in L'Aquila!

References

[1] N. Alon and J. H. Spencer. The probabilistic method. John Wiley & Sons, fourth edition, 2016.

[2] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations (extended abstract). In Proceedings of the ACM Symposium on Theory of Computing (STOC), 1994.

[3] Reuven Bar-Yehuda, Keren Censor-Hillel, and Gregory Schwartzman. A distributed $(2+\epsilon)$-approximation for vertex cover in $o(\log \Delta/\epsilon \log\log \Delta)$ rounds. J. ACM, 64(3):23:123:11, 2017.

[4] R. B. Boppana. Personal communication to Joel Spencer, 1987.

[5] Yun Kuen Cheung, Bhaskar Chaudhuri, Jugal Garg, Naveen Garg, Martin Hoefer, Kurt Mehlhorn. On Fair Division of Indivisible Items. CoRR abs/1805.06232, 2018.

[6] Paul Erdős and Tibor Gallai. Graphs with prescribed degrees of vertices [hungarian]. Matematikai Lapok, 11:264274, 1960.

[7] Sandeep S Kulkarni, Murat Demirbas, Deepak Madappa, Bharadwaj Avva, and Marcelo Leone. Logical physical clocks. In International Conference on Principles of Distributed Systems, pages 1732. Springer, 2014.

[8] L. Tseng and N. H. Vaidya. Fault-Tolerant Consensus in Directed Graphs. In PODC, pages 451460. ACM, 2015.

PODC 2018 Review

Naama Ben-David
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
nbendavi@cs.cmu.edu

The 2018 ACM Symposium on the Principles of Distributed Computing (PODC 2018) was held on July 23-27, at Royal Holloway, University of London, in Egham, UK (see Figure 1). This venue achieved a nice balance between two opposing desirable features of a conference location: being remote enough to encourage conference participants to attend most of the talks, and being central enough to allow for enjoyable tourism. Royal Holloway is only a 15 minute drive from Windsor Castle (see Figure 2), and about an hour away from central London. This allowed conference attendees to enjoy London before and after the conference, but also created a lively atmosphere in the lecture hall and during breaks, as most conference attendees remained in the area.

Figure 1: Royal Holloway, University of London

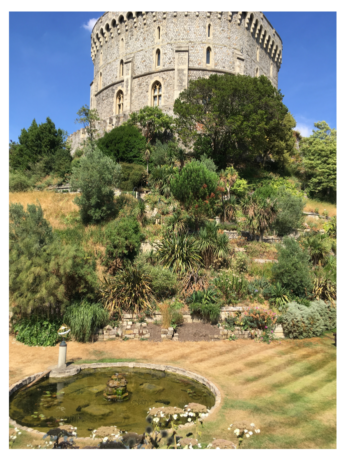


Figure 2: Windsor Castle

This year, PODC received 163 regular submissions, and 13 brief announcement submissions. Of these, 41 were accepted as full papers, and 18 were accepted as brief announcements. There were 144 attendees, with an additional 67 people who only attended the workshops. Keynote talks were given by Rob Peglar, Graham Cormode, and Keren Censor-Hillel. The technical content covered a wide range of topics, including shared memory algorithms, distributed graph algorithms, blockchains, persistent memory, and population protocols. Unfortunately, I cannot cover all presented papers in this review. Instead, I give an overview of the topics and events of each day, and discuss some talks in more detail.

PODC began with the *Jennifer Jubilee*– a celebration of Jennifer Welch's 60th birthday. Nancy Lynch, Nitin Vaidya, Shlomi Dolev, and Hagit Attiya gave talks remembering Jennifer throughout her career, and explaining her many contributions to our field. Happy Birthday, Jennifer!

Day 1

The first session of the conference featured papers about persistent memories. Upcoming memory technology will provide persistent, non-volatile main memory, meaning that upon a power failure, data in the cache will be lost, but data in main memory will remain unharmed. This introduces a lot of potential in designing applications that are resistent to system crashes. The first talk of the conference, a keynote talk given by Rob Peglar, addressed these new technologies. He gave an overview of persistent memory trends in the past, and explained why he believes that non-volatile memories (NVM) will be very important in the near future.

We then heard about recent research on this topic. Ohad Ben-Baruch presented his paper on Nesting-Safe Recoverable Linearizability, or NRL. In summary, this is a new correctness criterion for objects designed to enable recovery in NVM systems, where execution histories include crash events, upon which all data in the cache is lost. NRL ensures that an operation's return value is stored in a persistent manner, so that the execution can be recovered and continued after a crash. In the same session, Adam Alon described a deterministic algorithm for abortable mutual exclusion that has only sublogarithmic remote memory reference (RMR) complexity. Abortable mutual exclusion allows processes to abandon their attempt to grab a lock. A common approach to mutual exclusion that ensures starvation freedom is to have a queue of processes waiting for the lock, and simply pass the lock to the next process in line. However, simply abandoning an attempt can cause problems when determining which process should grab the lock next, as there can be many 'holes' in the queue. Adam presented a solution to this problem that involves constructing a tree on top of the queue to facilitate locating the next process in line.

The next session was about the theory of shared memory. I presented a paper entitled Passing Messages while Sharing Memory. The paper introduces the Message-and-Memory model – an asynchronous computational model in which processes can both pass messages and share memory, and may fail by crashing. In the paper, we show that this model allows solving consensus while tolerating more failures than in a message passing system, and that leader election can be solved with less synchrony requirements. David Yu Cheng Chan presented a paper about the classification of objects according to their set-agreement power. A classic result by Herlihy classifies objects according to their ability to solve consensus. However, the heirarchy produced is not precise, since some objects are placed at the same level of the heirarchy even though they cannot implement each other in all settings. David showed that, while set-agreement is a more refined way of classifying objects, it still cannot perfectly separate objects of different power.

The later sessions of the first day were about wireless networks and graph algorithms. Yi-Jun Chang presented a paper entitled The Energy Complexity of Broadcast. In the broadcast problem, one source node wants to send a message to all other nodes in a distributed graph. It takes 'energy' when a node has to either send a message in a round, or listen to see if it receives incoming messages. We want to minimze the energy cost, while keeping the number of rounds low. Yi-Jun showed how to broadcast with $O(\log n)$ energy. Ellis Hershkowitz presented a paper about round- and message- optimal distributed algorithms. Round complexity and message complexity are both common measures for evaluating a distributed graph algorithm. However, usually, an algorithm is optimized for one of the measures, and does not achieve optimality in the other measure. Furthermore, in some cases, the two goals seem at odds. In his talk, Ellis showed how to design algorithms that are optimal in both round and message complexity in the congest model. The techniques in this paper apply to many distributed graph problems, like MST, min cut, and single-source shortest paths.

The first day ended with the business meeting. Jennifer Welch was unanimously chosen as steering committee chair. There were lengthy discussions about two possible changes to future PODCs: parallel tracks, and double-blind reviewing. The motivation for parallel tracks is that in this way, we can grow PODC's size, number of accepted papers, and length of talks. Furthermore, there already seems to be a split in the community, roughly along the lines of synchronous vs asynchronous computation. It already feels like the program committee is effectively split, and people skip or attend sessions according to this divide in topics. However, people opposing to this suggestion felt that creating parallel tracks and thereby solidifying the divide would have a negative effect on our community, and could possibly weaken PODC's reputation. For now, the decision is not to pursue parallel tracks.

Double-blinding the review process came up because some people worry that single-blind reviewing benefits established members of the community, while making it harder for new people to join. However, some people were concerned that when submitting a paper for double-blind review, promoting the paper or putting it on ArXiv would be discouraged. Other people also felt that the identity of the author can be important when reviewing a paper, since verifying correctness is so hard to do.

Next year's PODC will be held in Toronto. Faith Ellen will be the program committee chair.

Day 2

The second day was opened with a keynote talk by Graham Cormode about Data Summarization and Distributed Computation. He showed that sketching and summarization can be useful for distributed computing tasks. The idea is that we want to capture the essence of data in as little space as possible, to allow for quick communication. Graham talked about how to continuously monitor data that is distributed across a network to compute a function of this data at a single coordinator. He gave examples inspired by machine learning.

The regular program then began with a talk by Nikola Konstantinov, who presented his paper about the convergence of stochastic gradient descent in asynchronous shared memory. The idea in shared memory stochastic gradient descent (SGD) is to give a mini-batch of data to each process, have each process calculate a new gradient from their data, and use fetch&add to update the global direction with their gradient. In practice, this works relatively well. However, in theory, it seems like the adversary might be able to prevent convergence by deciding who goes first, and arbitrarily delaying some process's updates. The paper shows that under some assumptions, the failure probability can be bounded. Ultimately, the adversary can slow down the computation, but not prevent convergence.

The next session focused on routing and leader election. Vijaya Ramachandran showed how to compute distributed exact weighted all pairs shortest path (APSP) in a deterministic manner in $\tilde{O}(n^{3/2})$ rounds. This problem has received a lot of attention recently, and this is the first deterministic algorithm to beat $O(n^2)$. The idea is based on the concept of an *h-hop shortest path*: a weighted shortest path that uses at most h edges. An h-hop single-source shortest path is computed from each vertex, and then a designated set of nodes, called a *blocker* set, is used to find shortcuts to improve the distances calculated.

Assaf Yifrach then talked about fair leader election for rational agents in rings and networks. The goal is to elect a leader in a network, despite possible preferences of each node. The difficulty is that nodes might prefer themselves to be the leader, or some other specific node, and could try to bias the decision in their favor. To overcome this, the agents each randomly pick a leader ℓ_i (with id from 1 to n), and send this to everyone. The final leader elected will be the process with id equal to $\sum_i \ell_i \mod n$. To prevent agents from using asynchrony to their advantage, messages are buffered.

After lunch, there was a session about blockchains and security. Maurice Herlihy talked about Atomic Cross-Chain Swaps. The problem is to design a protocol that allows parties that do not trust each other to swap resources. We can think of a directed graph of trades. If this graph is strongly connected, then there is an atomic cross-chain swap protocol for the graph. Vertices generate a secret key, and hash it to use as a lock.

In the last session of the day, Guy Goren presented his best student paper, entitled Silence. (See Figure 5.) The idea in this paper is that in synchronous networks, not sending a message in some round can convey information. This can be used to improve message complexity of algorithms. If processes may crash, silence conveys a little less information, since we don't know whether a process purposefully didn't send anything, or they actually crashed. However, even this little bit of information can still be useful. Guy showed how to use this insight for atomic commit: everyone has to commit or abort, and must commit only if all inputs were 1. To achieve message optimal atomic commit, silence must be used. The paper presents an atomic commit algorithm that is message optimal and time optimal, using only 3 rounds in the common case.

The second day ended with the PODC banquet. At the banquet, the Dijkstra Award was presented to Bowen Alpern and Fred B. Schneider for their paper, Defining Liveness. See Figures 3 and 4. This paper formally defined the concepts of safety and liveness, creating a foundation for many papers in the PODC community. Congratulations to Bowen and Fred!

Day 3

The final day of PODC started with a keynote talk by Keren Censor-Hillel. She talked about the CONGEST model, and barriers that arise from congestion in such networks. Interestingly, many problems in the CONGEST model require many rounds to compute exactly, but settling for approximate solutions can greatly speed things up. For example, computing the diameter of a graph requires $\Omega(n/B)$ rounds, where B is the capacity of each edge in a round, but to get a 2-approximation, we only need $O(D)$ where D is the diameter. We can pay a little bit more and do slightly better – to get a 3/2-approximation, we need $O(D+\sqrt{n})$. However, if we want to do better than that, there is suddenly a jump in price; a $(3/2-\epsilon)$-approximation requires $\Omega(n/(B\cdot\text{poly}\log n))$. So, we see that there is a sharp threshold for trading off efficiency and accuracy. This phenomenon repeats in other problems, like minimum vertex cover and subgraph detection.

Figure 3: Bowen Alpern and Ulrich Schmid

Figure 4: Ulrich Schmid and Fred Schneider

The rest of the morning session also focused on the CONGEST model. François le Gall talked about a variant of this model – the quantum CONGEST model. He showed that certain problems, including diameter and eccentricity, can get speedup in the quantum model over the classic one. He presented an algorithm that calculates the eccentricities of vertices in $O(\sqrt{nD})$ rounds in the quantum CONGEST model, where D is the diameter of the graph. In the classic model, there is a known linear lower bound for this problem.

The second session of the day dealt with concurrency. Dan Alistarh talked about relaxed priority schedulers, and how to use them to parallelize sequential tasks. Relaxed priority queues are concurrent data structures that experience redued contention when compared to their classic counterparts, but may return a value that is off from the minimum when a deleteMin operation is executed on them. This paper focuses on the use of relaxed priority queues as a substitute for classic priority queues as primitives in other algorithms. Dan showed that if the task for which they are used has uniformly random priorities, the benefit of faster accesses to the queue can more than make up for the extra work necessary to correct possible mistakes that the queue makes. An example was shown on calculating MIS. In the same session, the co-best student paper, An Asynchronous Computability Theorem for Fair Adversaries, was presented by Thibault Rieutord. (See Figure 5.) He gave a topological characterization of fair adversaries, which include commonly studied classes like wait-freedom and t-resilience. This paper thus generalized previously known characterizations of task computability.

After lunch, the best paper was presented by Uri Goldenberg. (See Figure 6.) This paper shows how to achieve $\Delta + 1$ coloring using a locally iterative algorithm; we start with a bad, but feasible, coloring, and iteratively improve it. Uri presented their algorithm, and showed that it can achieve a $\Delta + 1$ coloring in only $O(\Delta + \log^* n)$ rounds. These ideas also apply to other distributed graph problems.

The final session of the conference was about graph and population protocols. Goran Zuzic showed that minor-excluded network families admit fast distributed algorithms. This paper focuses

Figure 5: Idit Keidar with best student papers award winners Thibault Rieutord, Yuan He, and Guy Goren.

Figure 6: Idit Keidar with best paper award winners Michael Elkin and Uri Goldenberg.

on two problems in the CONGEST model: MST and min cut. The main result is that on a family of minor-free graphs, these two problems can be solved in $O(D^2)$ rounds, whereas on general graphs, a lower bound of $\tilde{\Omega}(\sqrt{n})$ is known.

Concluding Remarks

This year, as always, PODC included many exciting results, interesting conversations, and great company. There was a feeling that interest in distributed computing is growing, both from the theoretical community, and from the practical side. The days before and after PODC were filled with workshops and tutorials, ranging in topics from blockchains, to biologically-inspired distributed algorithms, to distributed machine learning, and bridging the gap between theory and practice in the field. All this serves to further highlight new directions in which the community is progressing, and I looking forward to hearing more developments next year in Toronto.

DISC 2018 Review

Aditya Biradavolu
Department of Computer Science and Engineering
Texas A&M University
College Station, TX, USA
aditya95913@tamu.edu

Saptaparni Kumar
Department of Computer Science and Engineering
Texas A&M University
College Station, TX, USA
saptaparni@tamu.edu

The *32nd International Symposium on Distributed Computing, DISC 2018* was held in New Orleans, USA, from October 15-19, 2018. The conference took place at the Hampton Inn & Suites New Orleans-Convention Center and was organized by Costas Busch of Louisiana State University. The conference included two workshops and three days of main conference this year.

1 Day 1

The 7th workshop on Advances in Distributed Graph Algorithms (ADGA) was held on Monday, October 15. There were six talks at this year's ADGA workshop. The first talk *The Unreasonable Effectiveness of Decay-Based Broadcasting in Radio Networks* was by Calvin Newport. Calvin gave a nice glimpse into the backstory and impact of the surprisingly simple randomized distributed algorithm BGI [1] for broadcasting a single message to all processes on a multihop shared channel in radio networks. This solution solved the problem faster (with high probability) than the best known centralized scheduling algorithm at the time and was later proved to be optimal among all possible distributed algorithms. Calvin goes on to describe their recent efforts to explore the behavior of this basic "decay" strategy in more realistic models which proves that the BGI algorithm retains strong guarantees even in very noisy and unpredictable radio networks. The next four talks were on varied topics ranging from graph sketching and streaming to optimality of single source shortest paths and spanning tree algorithms. The speakers were Andrew McGregor, Valerie King, Seth Pettie and Sebastian Forster.

The last talk in this workshop *Recent Advances in Population Protocols* was by Rati Gelashvili. Rati spoke about the computability and complexity characteristics of recent developments in population protocols. From a theoretical perspective, a population protocol is probably the simplest distributed model and yet, perhaps surprisingly, solutions to classical distributed tasks are still possible. Moreover, these solutions often rely on neat algorithmic ideas for design and interesting combinatorial techniques for analysis. Rati also discussed some open problems and directions in this area.

2 Day 2

The first day of the conference started with a keynote talk by Tom Goldstein titled *Challenges for Machine Learning on Distributed Platforms.* Tom explained the problems and challenges that arise when scaling deep neural nets over large systems, highlighting reasons why naive distributed training methods fail. He then discussed recent algorithmic innovations that have overcome these limitations, including big batch training for tightly coupled clusters, and variance reduction strategies to reduce communication in high latency settings.

The first session of the day was on distributed storage and was chaired by Ulrich Schmid. The first talk was on *Multi-Shot Distributed Transaction Commit* [2] by Gregory Chockler and Alexei Gotsman and this paper was awarded the Co-Best Paper award. Alexei motivated the paper by highlighting how the atomic-commit problem is too restrictive to capture the complexities of modern transactional data stores, where commit protocols are integrated with concurrency control, and their executions for different transactions are interdependent. As an alterntive, they introduce the Transaction Certification Service (TCS), a new formal problem that captures safety guarantees of multi-shot transaction commit protocols with integrated concurrency control. TCS is parameterized by a certification function that can be instantiated to support common isolation levels, such as serializability and snapshot isolation. Alexei also described a crash-resilient protocol for implementing TCS which achieves a better time complexity than mainstream approaches that layer two-phase commit on top of Paxos-style replication.

The next paper in this session was *Integrated Bounds for Disintegrated Storage* [3] by Alon Berger, Idit Keidar and Alexander Spiegelman. Alon pointed out a somewhat surprising similarity between non-authenticated Byzantine storage, coded storage, and certain emulations of shared registers from smaller ones. A common characteristic in all of these is the inability of reads to safely return a value obtained in a single atomic access to shared storage. The authors show that if readers are invisible, then the storage cost of such systems is inherently exponential in the size of written values; otherwise, it is at least linear in the number of readers. Their bounds are asymptotically tight to known algorithms, and thus justify their high costs. There were two more full talks and two brief announcements in this session.

The second session was on shared memory systems and it was chaired by Idit Keidar. The first paper of this session was *Allocate-On-Use Space Complexity of Shared-Memory Algorithms* [4] by James Aspnes, Bernhard Haeupler, Alexander Tong and Philipp Woelfel. James spoke about the consequences of adopting a per-execution approach to space complexity, where an object only counts toward the space complexity of an execution if it is used in that execution as opposed to worst case complexity. This helped show that many known randomized algorithms for fundamental problems in shared-memory systems have expected space complexity much lower than the worst-case lower bounds. It was also shown that many algorithms that are adaptive in time complexity can also be made adaptive in space complexity. James demonstrated this with the problem of mutual exclusion where the algorithm illustrates an apparent trade-off between low expected space complexity and low expected RMR complexity. There were two more papers and a brief announcement in this session.

The last two sessions for the day were on graph algorithms, chaired by Calvin Newport and

Nicola Santoro respectively. The third talk in the third session was *Redundancy in Distributed Proofs* [5] by Laurent Feuilloley, Pierre Fraigniaud, Juho Hirvonen, Ami Paz and Mor Perry. Distributed proofs are mechanisms enabling the nodes of a network to collectively and efficiently check the correctness of Boolean predicates on the structure of the network or on data structures distributed over the nodes. The authors consider well known mechanisms consisting of two components: a prover that assigns a certificate to each node, and a distributed algorithm called verifier that is in charge of verifying the distributed proof formed by the collection of all certificates. The authors show that many network predicates have distributed proofs offering a high level of redundancy, explicitly or implicitly. They use this property of distributed proofs to establish perfect tradeoffs between the size of the certificate stored at every node, and the number of rounds of the verification protocol. There were two more full papers and two brief announcements in the third session of the day.

The last session of the day had four papers. The third talk in this session was on *Distributed MST and Broadcast with Fewer Messages, and Faster Gossiping* [6] by Mohsen Ghaffari and Fabian Kuhn. Fabian spoke about their distributed minimum spanning tree algorithm with near-optimal round complexity of $\widetilde{O}(D + \sqrt{n})$ and message complexity $\widetilde{O}(\min\{n^{3/2}, m\})$. Their's is the first algorithm with sub-linear message complexity and near-optimal round complexity and it improves over other recent algorithms, which have the same round complexity but message complexity $\widetilde{O}(m)$. This method also gives the first broadcast algorithm with $o(n)$ time complexity when $D = o(n)$ with $o(m)$ messages. In addition to this Fabian explained how their method leads to an $\widetilde{O}\sqrt{nD})$-round GOSSIP algorithm with bounded-size messages which is the first such algorithm with a sublinear round complexity.

3 Day 3

The first session of this day was chaired by Yoram Moses, and it kicked off with a keynote talk by Sandor P. Fekete titled *Autonomous Vehicles: From Individual Navigation to Challenges of Distributed Swarms*. He talked about how even though many technological advances are being made in autonomous vehicles, there are still many challenges to be dealt with in some complex scenarios. He talked about self-organized structures in general, and their emergent behavior, i.e., actions of small individual agents in the system leading to a global phenomenon (for example, hurricanes or traffic jams). He then talked about the challenges in dealing with these self-organized structures particularly from the point of view of individual autonomous vehicles. He proposed some interesting game theoretic mechanisms to model these large scale issues, and reach the equilibrium between selfishness vs. cooperation. There were two other presentations in this session in the area of multi-agent systems.

The next session was chaired by Jennifer Welch, and it started with a presentation by Valerie King of the paper titled *Broadcast and Minimum Spanning Tree with $o(m)$ Messages in the Asynchronous CONGEST Model* [7]. This paper was selected as a co-best paper. The authors consider the problem of finding a minimum spanning tree in a distributed network with efficient message communication. This problem was motivated by mentioning about how researchers have been working on it for decades, and that the best message complexity until this paper was only $\Omega(n)$ in the asynchronous KT1 CONGEST model. Their paper was the first in this model to give an algorithm which uses $o(m)$ bits of communication to find a spanning tree. The algorithm is randomized and builds the minimum spanning tree with high probability. The authors use an interesting technique for building a spanning tree in phases, by using a pre-selected leader as the root to grow a tree. There were two other full presentations in this session in the area of wireless networks, followed by three brief announcements before the lunch break.

The third session was chaired by Gokarna Sharma, and all the four presentations in the session were on leader election. It started off with a presentation by Fabien Dufoulon on *Beeping a Deterministic Time-Optimal Leader Election* [8]. Then Shaked Rafaeli presented a paper on *The Role of A-priori Information in Networks of Rational Agents* [9]. The authors consider game-theoretic distributed computing, where all the agents in the network are rational, i.e., they may deviate from the algorithm only if they deem it as profitable. They assume the synchronous model and the worst possible utility function i.e., agents cheat even if slightly profitable. Shaked mentioned that in this setting, it is traditionally assumed that each rational agent know the total number of agents in the system, n. This assumption is unrealistic, particularly when large-scale distributed systems like blockchains, social networks, etc., are considered. The authors consider a few standard distributed problems and aim to know how the knowledge of n affects the equilibria of those distributed algorithms and proceed to show that it is in fact impossible to achieve equilibrium when all rational agents are considered, without prior knowledge of n. They show this using a Sybil attack, i.e., agents duplicating themselves to increase their utility.

The final session of the day was chaired by Robert Gmyr, and the presentations were on randomized network algorithms. One of these was on *A Population Protocol for Exact Majority with* $O(\log^{5/3} n)$ *Stabilization Time and* $\Theta(\log n)$ *States* [10], presented by Robert Elsässer. He first talked about population protocols in general, wherein each node of the graph has an initial state and all nodes communicate and update their states pairwise according to a deterministic transition function. The goal is to minimize the number of states needed per node as well as the stabilization time. He then presented a fast population protocol for the exact majority problem with $O(\log^{5/3} n)$ parallel stabilization time in high probability and $\Theta(\log n)$ states.

After a day with intense talks, everyone at the conference was looking forward to the banquet. The co-best paper awards were awarded to Gregory Chockler, Alexey Gotsman, Ali Mashreghi and Valerie King. The Doctoral Dissertation Award was awarded to Dr. Rati Gelashvili for his PhD. dissertation titled *On the Complexity of Synchronization*, supervised by Professor Nir Shavit at the Massachusetts Institute of Technology. A delicious meal was served at the banquet, which was enriched by a room full of ideas and discussions. To top that off, Costas had arranged for everyone at the conference to go on a "Ghost Tour" on foot at the "haunted" French Quarter of New Orleans. (See Figures 1 and 2.) It was a very unique experience for both the tour guides and the scientists on foot. Our tour guide was very impressed when Moti asked her about the technical difference between a spirit and a ghoul. We learned a lot about the history of New Orleans and that there is not much "French" left in the French Quarter and that most of the architecture was Spanish. Even though we were not really fortunate enough to experience anything paranormal, it was overall an extremely entertaining experience.

4 Day 4

The first session of this day was chaired by Costas Busch, and it started off with a keynote talk by Michael Mendler titled *Logical Analysis of Distributed Systems: The Importance of Being Constructive*. He talked about how in distributed systems, the abstraction of synchronization is in a way paradoxical and that achieving global consistency from local communications is impossible

Figure 1: Back of St. Louis Cathedral

Figure 2: French Quarter scene.

without synchronization. In this point of view, distributed algorithms only reduce one synchronization problem to another. To justify the logic of this abstraction, he studied this at the circuit level. He further talked about the concepts of constructive logic, intuitionistic semantics, modality of propagation delay and inertiality. This talk was followed by three other full presentations in the area of distributed agreement, and one brief announcement.

After the lunch break, the second session was chaired by Gregory Chockler, and it had four presentations, again in the area of distributed agreement. One of the presentations was by Calvin Newport, on *Fault-Tolerant Consensus with an Abstract MAC Layer* [11]. Calvin started by mentioning about the abstract MAC layer model. Most models of distributed systems just abstract the physical layer of networks, whereas this model abstracts the MAC layer as well. He then talked about the advantages of using this model and the difference between Abstract MAC and the asynchronous model. The authors looked at the binary consensus problem; the motivation for considering consensus in wireless settings is for data-center type scenarios. Calvin mentioned previous work on deterministic consensus in this model, which was shown to be impossible just like in the asynchronous model. In this paper, they gave two randomized algorithms for consensus in this model which work with any number of failures, require no knowledge of network size and terminate with high probability. This on the other hand is not possible in asynchronous model even if no faults are assumed.

The final two sessions were chaired by Peter Robinson and Robert Elsässer respectively, and the presentations were on the CONGEST Model. The third session started off with a presentation by Christian Konrad about the problem of detecting cliques in the classic distributed CONGEST network model, with limited bandwidth on the communication links [12]. The authors gave tight lower bounds for communication rounds to detect K_l cliques for $l \geq 4$, in the two party communication framework, using a reduction to the set disjointness problem. This was followed by a presentation by Merav Parter on *Congested Clique Algorithms for Graph Spanners* [13]. Merav briefly talked about the advantages of using the congested clique model in global problems, and its difference when compared to CONGEST and LOCAL models. The authors studied graph spanner constructions on the congested clique model for an n-vertex graph and gave the following three results. Randomized and deterministic constructions of a $(2k-1)$-spanner with $\tilde{O}\left(n^{1+1/k}\right)$ edges in improved number of rounds than previous algorithms was presented. This improvement is achieved by a new derandomization theorem for hitting sets. They also showed a deterministic construction of a $O(k)$ spanner with $O\left(kn^{1+1/k}\right)$ edges in $O(\log k)$ rounds. This was achieved by using a technique based on dividing the graph into sparse and dense edges and constructing spanners for each of these separately. There were five other presentations following this on CONGEST model before the end of the final day of the conference.

5 Day 5

The 2nd Workshop on Storage, Control, Networking in Dynamic Systems (SCNDS) was conducted on Friday, October 19. It had four invited talks, one tutorial and a few paper presentations. It started off with a keynote by Antonio Fernandez Anta titled *Putting Distributed Ledgers Together*. He started by motivating the need for a blockchain formalization and the possible use of distributed computing knowledge to achieve this, and quoted Herlihy's Keynote at PODC '17, regarding the same. Antonio and coauthors contributed to this formalization by defining a concurrent object called a Distributed Ledger Object (DLO) as a building block of blockchains, which stores a totally ordered sequence of records. It supports ***Append(r)*** and ***Get()*** operations. He mentioned about the inevitability of interconnection of blockchains, and the problems that arise as a result of it. He talked about atomic swap, atomic appends problem and about the competitive utility model, and gave a possible solution to the appends problem by using a Smart DLO. He also explained the scope of the future work, wherein the power of using Smart DLOs can be explored, and how the atomic appends problem can be studied under different utility models.

This was followed by a talk given by Nitin H. Vaidya, titled *Distributed Fault-Tolerant Optimization* In this talked Nitin talked about security and privacy for distributed optimization and learning. He first described the following simple convex optimization problem: $argmin\, h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$. This problem can have various different applications, for example each $h_i(x)$ can be the cost of a robot i to go to a location x, and the total cost of rendezvous has to be minimized. This could also relate to learning in that the data is distributed across different agents during training. He considered the case where agents are faulty or adversarial. In this case the original problem wouldnt be meaningful since it includes the cost of faulty agents, and we would want to optimize over non-faulty agents. So a technique is proposed wherein we allow unequal weights and modify the problem to $argmin\, h(x) = \sum_{i=1}^{n} \alpha_i h_i(x)$. The weight vector α must filter out the bad behavior. He also talked about the issue of privacy and mentioned how variants of differential privacy can be applied here. For example instead of communicating the cost function to neighbors directly we could add some amount of balanced noise (that is a function of the value being sent) to it, that cancels over the network.

Following this, there were some paper presentations and two other talks, one on *Resilient Distributed State Estimation of Dynamical Systems* by Shreyas Sundaram, and the other on *Securing Distributed Machine Learning in High Dimensions* by Lili Su. The workshop concluded after the tutorial by Kishori M. Konwar titled *Erasure Coding in Consistent Object Stores: Advantages, Challenges, Theory and Practice.*

References

[1] Reuven Bar-Yehuda, Oded Goldreich, Alon Itai: *On the Time-Complexity of Broadcast in Radio Networks: An Exponential Gap Between Determinism and Randomization.* PODC 1987: 98-108

[2] Gregory V. Chockler, Alexey Gotsman: *Multi-Shot Distributed Transaction Commit.* DISC 2018: 14:1-14:18

[3] Alon Berger, Idit Keidar, Alexander Spiegelman: *Integrated Bounds for Disintegrated Storage.* DISC 2018: 11:1-11:18

[4] James Aspnes, Bernhard Haeupler, Alexander Tong, Philipp Woelfel: *Allocate-On-Use Space Complexity of Shared-Memory Algorithms.* DISC 2018: 8:1-8:17

[5] Laurent Feuilloley, Pierre Fraigniaud, Juho Hirvonen, Ami Paz, Mor Perry: Redundancy in Distributed Proofs. DISC 2018: 24:1-24:18

[6] Mohsen Ghaffari, Fabian Kuhn: *Distributed MST and Broadcast with Fewer Messages, and Faster Gossiping.* DISC 2018: 30:1-30:12

[7] Ali Mashreghi, Valerie King: *Broadcast and Minimum Spanning Tree with $o(m)$ Messages in the Asynchronous CONGEST Model*. DISC 2018: 37:1-37:17

[8] Fabien Dufoulon, Janna Burman, Joffroy Beauquier: *Beeping a Deterministic Time-Optimal Leader Election*. DISC 2018: 20:1-20:17

[9] Yehuda Afek, Shaked Rafaeli, Moshe Sulamy: *The Role of A-priori Information in Networks of Rational Agents*. DISC 2018: 5:1-5:18

[10] Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, Tomasz Radzik: *A Population Protocol for Exact Majority with $O(log^{5/3}n)$ Stabilization Time and $\Theta(logn)$ States*. DISC 2018: 10:1-10:18

[11] Calvin Newport, Peter Robinson: *Fault-Tolerant Consensus with an Abstract MAC Layer*. DISC 2018: 38:1-38:20

[12] Artur Czumaj, Christian Konrad: *Detecting Cliques in CONGEST Networks*. DISC 2018: 16:1-16:15

[13] Merav Parter, Eylon Yogev: *Congested Clique Algorithms for Graph Spanners*. DISC 2018: 40:1-40:18