Writeup on research interests and some results

Srinivasan Kannan (alias) Ka.Shrinivaasan (alias) Shrinivas Kannan
Ph: 9789346927, 9003082186, 9791165980
Krishna iResearch Open Source Products Profiles:
http://sourceforge.net/users/ka_shrinivaasan,
Personal website(research): https://sites.google.com/site/kuja27/
ZODIAC DATASOFT: https://github.com/shrinivaasanka/ZodiacDatasoft
emails: ka.shrinivaasan@gmail.com, shrinivas.kannan@gmail.com,
kashrinivaasan@live.com

My research interests are:
- Hash Functions and Datastructures (bloom filters and other hashing algorithms)
- Satisfiability and Majority Voting (majority circuits and SAT)
- Graph algorithms (Expander graphs, Random graphs, Dynamic graphs, Graph spectra)
- Parallel RAM algorithms for sorting and merging sorted lists
- Circuit lowerbounds
- Probabilistic, Approximate, Correct Learning and Learning boolean functions from a dataset
- Computational Number Theory and Geometry
- Application of Graph algorithms to machine learning problems viz., Document Comparison and Ranking without Link graphs, Summarization, etc.,
- Non-statistical Graphical inference models
- Program (Software) Analysis, application of graph reachability and machine learning to Program analysis and debugging

During past 2 years and more I am privately researching on some complexity related problems (atleast as directions for research) though not officially as a PhD student. I have uploaded my research drafts in my website: https://sites.google.com/site/kuja27/. This mail is about few probable results in these drafts. These are my ongoing private research only and might have errors:

1. A Computational Geometric Algorithm for Factorization – Discrete Hyperbolic Factorization

Computational geometric factorization algorithm that uses
hyperbola to factorize in parallel time using Parallel RAM NC circuits:

Link to the above parallel factorization draft PDF (with lot of handwritten illustrations also) is at:
http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegeFactorization_PRAM_TileMergeAndSearch_And_Stirling_Upperbound.pdf/download

and Parallel RAM implementation handwritten notes with illustrations are at(requires a tile_id due to shuffling after parallel merge and sort of the tiles):
http://sourceforge.net/p/asfer/code/HEAD/tree/ImplementationDesignNotesForDiscreteHyperbolicFactorizationInPRAM.jpg.

TeX version of the draft is at:
http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_PRAM_TileMergeAndSearch_And_Stirling_Upperbound.tex/download

A recent version with Parallel RAM to NC reduction and Input size references is at:
http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_PRAM_TileMergeAndSearch_And_Stirling_Upperbound_updateddraft.pdf/download

http://sourceforge.net/projects/acadpdrafts/files/DiscreteHyperbolicPolylogarithmicSieveForIntegerFactorization_PRAM_TileMergeAndSearch_And_Stirling_Upperbound_updateddraft.tex/download

Above first discretizes or tesselates a hyperbolic curve (xy=<some integer>) and uses Binary Search, Interpolation Search and All Nearest Smaller Values parallel RAM algorithm (http://en.wikipedia.org/wiki/All_nearest_smaller_values#CITEREFBerkmanSchieberVishkin1993) on a sorted list of integers which are points on merged discretized hyperbolic tiles to find the factors in O(logN*logN) polylog time using polynomial processors in PRAM model and thus in NC.

The Sequential version of the above factorization algorithm has been implemented already and is working
(at: http://sourceforge.net/p/asfer/code/HEAD/tree/cpp-src/miscellaneous/DiscreteHyperbolicFactorizationUpperbound.cpp).

Sample result logs for
sequential implementation are at: http://sourceforge.net/p/asfer/code/HEAD/tree/cpp-src/miscellaneous/DiscreteHyperbolicFactorization_FactorsOf_4189998_Screenshot_201311-08_103903.png.

(Older PDF and TeX versions of the above are in the same url -
https://sites.google.com/site/kuja27)

2. Majority Voting and Pseudorandom Choice

This set of drafts is about even more significant problem of containment of P in NP. Infact this has origins to a probability series I was working in 2006 about relative merits of Majority Voting and Pseudorandom choice (i.e Randomly chosen or majority voted which is better). I did my Master's thesis partly using it in 2010 followed by publication in TAC 2010. They are available at:

2.1 http://arxiv.org/abs/1006.4458 (Initial sections mention about this series) (2010)

2.2 http://www.nist.gov/tac/publications/2010/participant.papers/CMI_IIT.proceedings.pdf (Initial sections mention about this series) (2010-11)

2.3 https://sites.google.com/site/kuja27/DocumentSummarization_using_SpectralGraphTheory_RGOGraph_2014.pdf?attredirects=0&d=1 (2014)

This series and probability of good choice defined in: https://sites.google.com/site/kuja27/CircuitForComputingErrorProbabilityOfMajorityVoting_2014.pdf?attredirects=0&d=1

The computation of series is at: http://sourceforge.net/p/asfer/code/HEAD/tree/cpp-src/miscellaneous/pgood_batch_sum.cpp and requires hypergeometric functions in general case. But it converges in special cases of p=1 and p=0.5 (i.e Probability of good choice for each individual voter is uniformly 1 or 0.5) to 1 and 0.5 both sides pseudorandom choice and majority voting.

From 2012 onwards, I thought of extending the above graduate thesis into a PhD level dissertation problem and wrote a set of drafts during past 2 years which again surprisingly gave me a result P=NP if there is a perfect judgement i.e With zeroerror P=NP (and) P!=NP is undecidable if existence of Perfect Voter with zeroerror is undecidable by constructing a circuit for the above Majority Voting and Pseudorandom Choice series. Because of the enormity of this, I am still verifying it. The series convergence is also illustrated in http://sourceforge.net/p/asfer/code/HEAD/tree/cpp-src/miscellaneous/MajorityVotingErrorProbabilityConvergence.JPG

I also felt that it has some connections to Arrow's Theorem - A Proof of Generalization of Arrow's theorem to infinite number of candidates (if decidable) implies P !=NP due to the above series. Point 10 in http://sourceforge.net/p/asfer/code/HEAD/tree/AstroInferDesign.txt describes this.

Following are the set of drafts and handwritten notes on the above (these are all connected in someway to the above series):

2.4 - Equating Majority Voting and Pseudorandom Choice
https://sites.google.com/site/kuja27/LowerBoundsForMajorityVotingPseudorandomChoice.pdf?attredirects=0

2.5 - Above Series Derivation
https://sites.google.com/site/kuja27/CircuitForComputingErrorProbabilityOfMajorityVoting_2014.pdf?attredirects=0&d=1

2.6 - Axiom of Choice and Majority Voting
https://sites.google.com/site/kuja27/IndepthAnalysisOfVariantOfMajorityVotingwithZFAOC_2014.pdf?attredirects=0

2.7 - A PRG for Pseudorandom Choice using Chaos Theory Attractors
https://sites.google.com/site/kuja27/ChaoticPRG.pdf?attredirects=0

2.8 - Interview Algorithm and Interactive Proof
https://sites.google.com/site/kuja27/InterviewAlgorithmInPSPACE.pdf?attredirects=0

2.9 - Equivalence of Integer Partitions and Hash Functions
https://sites.google.com/site/kuja27/IntegerPartitionAndHashFunctions_2014.pdf?attredirects=0&d=1. (This also gives an alternative proof of NP completeness of multipartisan democracy using reduction from restricted partition problem Money Changing Problem and Schur theorem for partitions)

2.10 - A gadget for randomized space filling to simulate some natural phenomena and LP for it - https://sites.google.com/site/kuja27/Analysis%20of%20a%20Randomized%20Space%20Filling%20Algorithm%20and%20its%20Linear%20Program%20Formulation.pdf?attredirects=0

2.11 – Arrow's Theorem , Circuit for Democracy and P versus NP
https://sites.google.com/site/kuja27/CircuitsForDemocracyAndPseudorandomChoice_and_PVsNP.pdf?attredirects=0&d=1

Handwritten notes:

2.12 - Philosophical analysis of Democracy circuit and Pseudorandom choice
https://sites.google.com/site/kuja27/PhilosophicalAnalysisOfDemocracyCircuitAndPRGChoice_20140326.pdf?attredirects=0

2.13 – Handwritten notes on Schur Theorem, Restricted Partitions and Hash Functions, NP-Completeness of MultiPartisan Majority Voting
https://sites.google.com/site/kuja27/SchurTheoremMCPAndDistinctPartitions_20140417.pdf?attredirects=0

2.14 - An experimental theory of Convex Hull of the logical implication graph and Perfect Voter problem -
https://sites.google.com/site/kuja27/ImplicationRandomGraphConvexHullsAndPerfectVoterProblem_20140111.pdf?attredirects=0&d=1

2.15 - Experimental Notes on Logical Implication Graphs:
An experimental logical implication graph model for P and NP http://sourceforge.net/projects/acadpdrafts/files/ImplicationGraphsPGoodEquationAndPNotEqualToNPQuestion_

excerpts.pdf/download

Basic idea is to construct a Majority with SAT circuit as inputs and each SAT input is a voter who needs to be satisfied. Thus Majority voting or Democracy is NPComplete, but the Pseudorandom generator is in P or NC. In the above series if both LHS(Pseudorandom choice) and RHS(Majority voting) are 100% (i.e zero error) as shown in above convergence, then there is a polytime algorithm (pseudorandom choice) to the NP problem of Majority voting which is a counterexample. One-wayfunctions exist when the series does not converge and do not exist when series converges to 100%.

Thus if P != NP then there cannot be a perfection. Moreover, finding even a single perfect voter who never makes an error itself looks to be intractable. Infact this circuit requires all voter SATs to be perfect for the series to converge to 100% thereby placing a tighter restriction than mere intractability. Thus above is not against what is widely believed (i.e P is properly contained in NP). Rather above gives 3 possibilities – when P can equal NP, when it cannot and whether question itself is decidable for infinite case.

(Older PDF and TeX versions of the above are in the same url - https://sites.google.com/site/kuja27)

3.Circuit for Complement Functions – special case for Riemann Zeta Function

A non-uniform boolean circuit and arithmetic circuit for complement function (special case for Riemann Zeta Function for prime distribution) using Fourier Analysis and Euler product and some conjectures based on the property of this non-uniform circuit graph family. The draft writeup is at point 24 of http://sourceforge.net/p/asfer/code/HEAD/tree/AstroInferDesign.txt.
This has some importance due to the connection that can be conjectured between the zeros of Riemann Zeta Function (if Riemann Hypothesis is true) and the patterns in the circuit graphs for this non-uniform family. Also it has connections to Ihara Zeta Function and Ramanujan Graphs. Riemann Zeta Function can be written in terms of Ihara identity of Ihara Zeta Function by assuming infinite set of prime-regular graphs. Solving the product of Characteristic polynomials of the prime regular graphs for the zeroes by equating them to eigenvalue (<= degree) gives some information about the Riemann Zeta Function zeros. Illustrations for this are at : https://sites.google.com/site/kuja27/RamanujanGraphsRiemannZetaFunctionAndIharaZetaFunction.pdf?attredirects=0&d=1

This is an extension of a miniproject course I was doing with Meena Mahajan (IMSc) in 2011 (http://arxiv.org/abs/1106.4102)

My detailed CV is at: https://sites.google.com/site/kuja27/CV_of_SrinivasanKannan_alias_KaShrinivaasan_alias_ShrinivasKannan.pdf?attredirects=0&d=1. Open Source Product codebases above are my own.