Titles and Abstracts
Plenary Talks

Algorithms in Algebra and Combinatorics

Steve Linton, University of St Andrews

Abstract: This talk will explore the relationships between "pure" research in algebra and combinatorics, research aimed at developing or analyzing algorithms for problems in these areas, implementation of those algorithms into actual software systems and computational experiments applying those algorithms to actual problems. The talk will be built around the idea that a "virtuous circle" links all of these four types of research in both directions:

- theory underpins algorithms and analysis of algorithms raises new theoretical questions
- algorithms are implemented into systems, whose designs often throw up new algorithmic problems
- systems enable experiments and are validated by them
- experiments lead to theoretical insight.

This concept will be illustrated by a range of examples from different areas of discrete mathematics.

Computer systems for algebraic and combinatorial computation

Steve Linton, University of St Andrews

Abstract: n this talk I will survey, very briefly, some of the history of software in this area and the capabilities and interests of current offerings. After that I will explore some current trends and research questions in this area, in particular, the increasing need to harness parallel computers and multiple software systems to solve complex and the interesting system design and computer science questions that this is throwing up.

1. Monte-Carlo Methods,
2. Randomised Approximation Algorithms

Dominic Welsh, Oxford University

Abstract: In these two lectures I shall start by describing some very early Monte Carlo techniques and then go on to concentrate on the two problems of approximate counting and generating a sample approximating a probability distribution which is difficult to compute exactly. These two problems are closely related and a key tool in both is the use of finite Markov chains. After describing some heuristic algorithms I shall discuss the rigorous techniques which have resulted from the two landmark papers of Jerrum and Sinclair in 1989 on approximating the permanent and the partition function of the Ising model. Almost all the examples used will be basic counting/generation problems arising in graph theory, and I shall concentrate particularly on the use of coupling methods to achieve good approximations by Markov chain Monte Carlo.

A brief history of generative models for power law and lognormal distributions

Michael Mitzenmacher, Harvard University

Abstract: I recently became interested in a current debate over whether file size distributions are best modelled by a power law distribution or a lognormal distribution. In trying to learn enough about these distributions to settle the question, I found a rich and long history, spanning many fields. Indeed, several recently proposed models from the computer science community for things like page links in the World Wide Web have antecedents in work on power law and lognormal distributions from decades ago. Here, I briefly survey some of this history, focusing on underlying generative models that lead to these distributions. One finding is that lognormal and power law distributions connect quite naturally, and hence, it is not surprising that lognormal distributions have arisen as a possible alternative to power law distributions across many fields. Based on this history, I suggest future directions for power law research, particularly in computer science.

Hashing algorithms and data structures for network measurement and monitoring

Michael Mitzenmacher, Harvard University

Abstract: Many network flow monitoring and measuring tasks are now being done using hash-based data structures, including Bloom filters and their many variations. In this talk, we review the basics of these data structures, and discuss some of our own work in this space, including our work on approximate concurrent state machines, novel constructions of Bloom filters and counting Bloom filters, and the benefits of allowing moves in multiple-choice hash tables.

Analysis of High-Throughput Biological Data Part I: Scalable High Performance Algorithms and Implementations

Michael Langston, University of Tennessee

Abstract: This talk is focused on implementation issues for novel algorithmic methods based on the theory of fixed-parameter tractability. When combined with high performance computational platforms, these methods can be used to launch systematic attacks on key combinatorial problems of widespread significance. Efficient sequential techniques for problem reduction and highly parallel algorithms for exhaustive search will be discussed, as will the trade-offs between real and synthetic data. The importance of maintaining a balanced decomposition of the search space is often critical to achieving scalability.

Analysis of High-Throughput Biological Data Part II: Computational Bottlenecks and Novel Applications

Michael Langston, University of Tennessee

Abstract: This talk is aimed at applications of novel algorithmic technologies to high-throughput computational biology. The analysis of microarray data serves as a prime example. Using mRNA samples obtained from recombinant inbred strains of mice, we can now solve immense instances of clique and related problems to derive putatively co-regulated genesets and other items of interest. The depth of genetic analysis we can perform is vastly enhanced by combining these results with the knowledge of cis-regulatory elements, microRNA binding sites, ontological classifications, literature review, and causal structures imposed with quantitative trait locus mapping. Techniques for dealing with noisy data are important concerns. Long-term goals include the elucidation of genetic variation effects and the discovery of biological network structures.

Computing isomorphisms and automorphisms of discrete structures

Brendan McKay, Australian National University

Abstract: Discrete stuctures consist of some sets and some relations between them. Examples include graphs (sets of vertices and edges with an incidence relation), matrices (sets of rows, columns and entry values), and all manner of other structures. Two structures in the same class are called isomorphic if there is a bijection between the sets that respects the relations. Isomorphic structures are thus essentially the same apart from the names of the elements of the sets comprising them. An automorphism is an isomorphism from a structure to itself. An example is a symmetry of a geometric figure. The composition of two automorphisms is also an automorphism, so under some finiteness condition the set of all automorphisms form a group under composition. In this talk we address the practical question of determining whether two structures are isomorphic and of computing the automorphism group of a structure.

Constructive enumeration of discrete structures without isomorphs

Brendan McKay, Australian National University

Abstract: We survey the problem of generating classes of combinatorial objects without isomorphic objects appearing. The methods considered will include orderly generation and the method of canonical augmentation, as well as more brute-force methods. Examples will be given from graph theory and latin squares.

Contributed Talks

The coolest way to generate balanced parenthesis and k-ary Dyck words

Aaron Williams, University of Victoria, Canada

Abstract: Balanced parenthesis are strings with an equal number of ('s and)'s together with the property that no prefix contains more)'s than ('s. The balanced parentheses of length $2n$ are counted by the nth Catalan number and are in bijective correspondence to objects as varied as binary trees and triangulations of convex polygons. In this talk I will present a novel algorithm for generating exhaustive listings of balanced parenthesis. The algorithm is simple enough to be taught to undergraduate students in a single lecture. It also has optimal efficiency in the sense that each successive string is produced in $O(1)$ time while using only $O(n)$ bits of total storage. The result is based on a subtle variation of the cool-lex method for generating combinations. Further modifications result in a generalized algorithm for k-ary Dyck words and linear extensions of B-posets. This is joint work with Frank Ruskey.

An explicit universal cycle for the $(n-1)$-permutations of an n-set

Frank Ruskey, University of Victoria, Canada

Abstract: We show how to construct an *explicit* Hamilton cycle in the directed Cayley graph $\overrightarrow{\mathrm{Cay}}(\{\sigma_n, \sigma_{n-1}\} : \mathbb{S}_n)$, where $\sigma_k = (1\ 2\ \cdots\ k)$. The existence of such cycles was shown by Jackson (Discrete Mathematics, **149** (1996) 123–129) but the proof only shows that a certain directed graph is Eulerian, and Knuth (Volume 4 Fascicle 2, Generating All Tuples and Permutations (2005)) asks for an explicit construction. We show that a simple recursion describes our Hamilton cycle and that the cycle can be generated by an iterative algorithm that uses $O(n)$ space. Moreover, the algorithm produces each successive edge of the cycle in constant time; such algorithms are said to be *loopless*.

Optimizing Diversity with Ecological Constraints

Beata Faller, University of Canterbury

Abstract: Phylogenetic diversity is a measure for describing how much of an evolutionary tree is spanned by a subset of species. Given a taxon set and a corresponding edge-weighted evolutionary tree, a central question in conservation biology is how to choose a fixed number of taxa with maximum diversity. This problem can be solved by the greedy algorithm. However, if we consider a more realistic problem, where we also have an acyclic digraph describing dependencies between taxa, we are only allowed to choose a viable taxon set of given size and the problem becomes more complicated. This talk will discuss the complexity of this latter problem. This is joint work with Magnus Bordewich and Charles Semple.

Limit probabilities of random Boolean expression values

Alexey Yashunsky, Moscow State University

Abstract: Random Boolean expressions are obtained by substituting independently at random the constants 1 and 0 with probabilities p and $1-p$, accordingly, for variables of a random read-once formula, containing symbols of Boolean operations from a given set. We study the limit probability of expressions with value 1, as expression complexity grows infinitely. This limit - the "probability function" of variable p - is expressed explicitly and investigated in detail. In particular, a Weierstrass-like theorem is proved for approximation of continuous functions by probability functions.

Making networks RealTime is NP-complete

Tanja Gernhard, University of Munich

Abstract: Binary networks are used to represent phylogenies with recombination events. Recombination events are assumed to happen instantaneously, a network fulfilling this time constraint is called RealTime. Reconstructed networks which do not fulfill the RealTime condition can be made RealTime by adding additional taxa – species which we did not sample or which are extinct. We show that determining the minimal number of taxa to add such that the network becomes RealTime is NP-complete. Further we discuss approximation algorithms for that problem.

Algorithms for achieving consensus

Arkadii Slinko, University of Auckland

Abstract: In this talk we are trying to investigate the following question: can automated agents be programmed to use the consensus decision making? We suggest a protocol which leads to an NP-complete problem but for which the

corresponding parameterized language is kernalizable and hence fixed parameter tractable.

FPT algorithms for path-transversals and cycle-transversals problems in graphs

Sylvain Guillemot, INRIA, France

Abstract: We consider problems on graphs of the following form: given a graph, remove p edges/vertices to achieve some property. The first kind of problems are *separation problems* on graphs, where we aim at separating distinguished vertices in a graph. The second kind of problems are *feedback set problems* on group-labelled graphs, where we aim at breaking nonnull cycles in a graph. We obtain new FPT algorithms for these different problems. A building stone for our algorithms is a general $O^*(4^p)$ algorithm for a class of problems aiming at breaking a set of paths in a graph, provided that the set of paths has a special property called *homogeneity.*

Certifying non-representability of matroids over finite fields

Geoff Whittle, Victoria University

Abstract: Matroids axiomatise the combinatorial properties of a collection of vectors over a field. If a matroid is isomorphic to one obtained as a set of vectors from a given field $\mathbb{F}$, then we say that M is $\mathbb{F}$-*representable.* Using an oracle model of complexity it is provably exponential to decide if a matroid is $\mathbb{F}$-representable for any field $\mathbb{F}$. On the other hand, if $\mathbb{F}$ is finite it turns out that it can be shown that a matroid is *not* $\mathbb{F}$-representable using a polynomial number of calls to the oracle. This is joint work with Jim Geelen, Waterloo, and Bert Gerards, CWI Amsterdam.

Recovering Reticulation: An Application of a FPT Algorithm to a Real-World Problem

Charles Semple, University of Canterbury

Abstract: Reticulate (non-tree-like) evolution is a fundamental process in the evolution of certain groups of species. This process results in species being a composite of DNA regions derived from different ancestors. Consequently, conflicting signals in a data set may not be the result of sampling or modelling errors, but due to the fact that reticulation has played a role in the evolutionary history of the species under consideration. Such species include certain birds and plants. Assuming that our initial data set is correct, a fundamental problem for biologists is to compute the minimum number of reticulation events that

explains this set. This smallest number sets a lower bound on the number of such events and provides an indication of the extent that reticulation has had on the evolutionary history of a collection of present-day species. In this talk, we describe a fixed-parameter algorithm for solving this problem for when the initial set consists of two (evolutionary) trees. Considering such a restricted input may seem rather special, but there are several reasons for this. Firstly, the problem is NP-hard even when the initial set consists of two such trees. Secondly, we are interested in finding a general solution rather than one that is restricted in some way. Lastly, the problem for when the initial data set consists of binary sequences can be interpreted as a sequence of two-tree problems.