

Decision problems – those in which the goal is to return an answer of “YES” or “NO” – are at the heart of the theory of computational complexity, and remain the most studied problems in the theoretical algorithms community. However, in many real-world applications it is not enough just to decide whether the problem under consideration admits a solution: we often want to find all solutions, or at least count (either exactly or approximately) their total number. It is clear that finding or counting all solutions is at least as computationally difficult as deciding whether there exists a single solution, and indeed in many cases it is strictly harder even to count approximately the number of solutions than it is to decide whether there exists at least one (assuming P is not equal NP). In this talk I will discuss a restricted family of problems, in which we are interested in solutions of a given size: for example, solutions could be k -vertex subgraphs in a large host graph G that have some specific property (e.g. being connected), or size- k solutions to a database query. In this setting, although exact counting is strictly harder than decision (assuming standard assumptions in parameterised complexity), the methods typically used to separate approximate counting from decision break down. Indeed, I will describe two randomised approaches that, subject to some additional assumptions, allow efficient decision algorithms for problems of this form to be transformed into efficient algorithms to find or count all solutions. This includes joint work with John Lapinskas (Bristol) and Holger Dell (Frankfurt).