Research goals

The research area that most excites me is the *theoretical foundations of artificial intelligence*. This research involves three key steps. First, a *mathematical model* is identified that abstracts key features of several concrete problems (the more the merrier). Second the model is *analyzed theoretically*—algorithms are designed and theorems about them proved. Third and finally, the analysis of the model is *exploited* to better understand the concrete problems and vice versa, often by empirical study. I do all three steps from time to time, but the theoretical analysis step is my specialty. I try to keep the first and third steps in mind even when I do not do them myself. This causes me to highly value (a) *simplicity/elegance* (of models, algorithms, theorems and proofs) and (b) *proper modeling*.

Research experience

Many problems encountered in AI are NP-complete. One way to attack NP-complete problems is via *approximation algorithms*, which do not generally find the optimum solution but provably find a solution that is not much worse. An algorithm for a minimization problem is an α-approximation if its output has cost at most α times the cost of the optimum solution. An α-approximation for all $\alpha > 0$ is known as a *polynomial time approximation scheme* (PTAS). The majority of my research experience involves approximation algorithms for a variety of problems.

Correlation clustering. Clustering is an important tool for analyzing large data sets. Correlation clustering is a type of clustering that uses a very basic form of input data: indications that certain pairs of vertices (pairs of data items) belong in the same cluster, and certain other pairs belong in different clusters. Unfortunately the information is not necessarily consistent, possibly claiming for example that "cat" is similar to "dog" and "dog" is similar to "bog" but "cat" is not similar to "bog". We assume that we have information about every pair of objects. The goal is to find a clustering, that is, a partition of the vertices, that disagrees with as few pieces of information as possible. Correlation clustering is illustrated in Figure 1. This problem has applications in data mining and natural language processing.

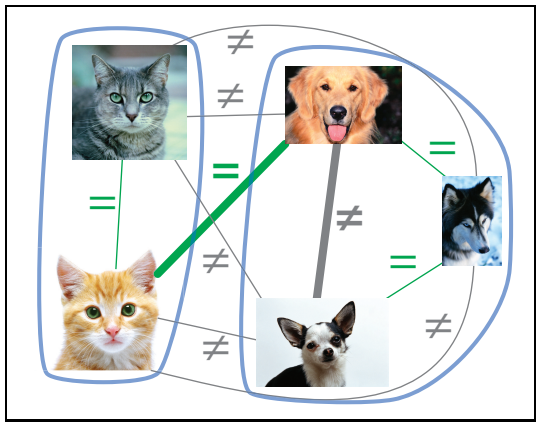

Figure 1: A sample correlation clustering problem and its optimal clustering. The optimal clustering (dogs and cats) has only two disagreements, shown in bold.

Charikar, Guruswami and Wirth showed that worst-case instances of correlation clustering can be fairly difficult—no PTAS exists unless P=NP. We therefore studied correlation clustering in a model which is a compromise between random and adversarial: We start from an arbitrary partition of the vertices into clusters. Then, for each pair of vertices, the similarity information is corrupted (made noisy) independently with probability p. Finally, an adversary generates the input by choosing similarity/dissimilarity information arbitrarily for each corrupted pair of vertices. **In this model Claire Mathieu and I [SODA '10] give a $(1 + \mathbf{O}(\mathbf{n}^{-1/6}))$-approximation.** We can also perfectly recover those planted clusters that are relatively large. **Micha Elsner and I [ILP-NLP '09] also compared a variety of correlation clustering algorithms experimentally.**

Feedback arc set tournament. Suppose you ran a chess tournament, everybody played everybody (a.k.a. round robin) and you wanted to use the results to rank everybody. Unless you were really lucky, the results would not be acyclic, so you could not just sort the players by who beat whom. A natural objective is to find a ranking that minimizes the number of upsets, where an upset is a pair in which the player ranked lower in the ranking beat the player ranked higher. Minimizing the number of upsets is called *feedback arc set problem* on tournaments (FAST). One application of this problem is the use of a binary

classifier to learn rankings. **Claire Mathieu and I [in submission; prelim. ver. STOC '07] give the first PTAS for this NP-hard problem.** A weighted generalization of our result yields the first PTAS for a compelling rank aggregation metric called Kemeny rank aggregation.

Fragility. Feedback arc set tournament is a specific problem in the general class of *min ranking constraint satisfaction problems* (CSPs). Such a problem consists of a constant arity k ($k = 2$ for FAST), a set of vertices V and a set of constraints on the vertices. Each constraint depends on the ordering of a set k of vertices and is satisfied by some of the possible orderings of those vertices. The objective is to find an ordering minimizing the number of unsatisfied constraints. In FAST each directed edge (game) corresponds to a constraint which is satisfied whenever the winner of the game is ordered before the loser. Another min ranking CSP is *betweenness*, which ranks objects given information of the form "v is between u and w," i.e. either $u < v < w$ or $w < v < u$.

A constraint S is called *fragile* if changing the relative order of a single vertex in S with respect to the rest of S makes it unsatisfied whenever S was satisfied by the original order. A min ranking CSP is *fully dense* if it contains a constraint for every set of k vertices. It is easy to see that feedback arc set tournament is a fragile fully-dense min ranking CSP. **Marek Karpinski and I [in submission] generalized our FAST result by showing that every fragile fully-dense min ranking CSP has a PTAS.** In particular we give the first PTAS for fully-dense betweenness.

Marek Karpinski and I also gave analogous results for ordinary (non-ranking) CSPs. A min CSP consists of a constant arity k (often 2), a set of variables V which take values from a constant-sized domain D (the clusters), and a set of constraints on the variables. Each constraint depends on a set of k of the variables and is *satisfied* by some of the possible configurations of those variables. The class of min CSP problems is quite broad, including for example the unique games problem and the problem of finding a coloring of an undirected graph with two colors while minimizing the number of bichromatic edges. A constraint is *fragile* if changing the value assigned to a single variable changes all the satisfied constraints it was in previously to unsatisfied. We say that an instance is *everywhere-dense* if each variable participates in $\Omega(n^{k-1})$ constraints. **Marek Karpinski and I [STOC '09] found a PTAS for all fragile everywhere-dense min CSPs.** A variant of our techniques gives a more efficient PTAS for correlation clustering with a fixed number of clusters.

Additive error. Our approximation algorithms for the fragile problems discussed above call *additive error algorithms* as a subroutine. These algorithms take a min CSP with arity k as input and produce an assignment with cost at most ϵn^k more than optimal. Several such algorithms were previously known, but previous algorithms were a bit complicated, especially for $k > 2$. **Claire Mathieu and I [SODA '08] used martingales to show that a simple and elegant greedy algorithm has low additive error.**

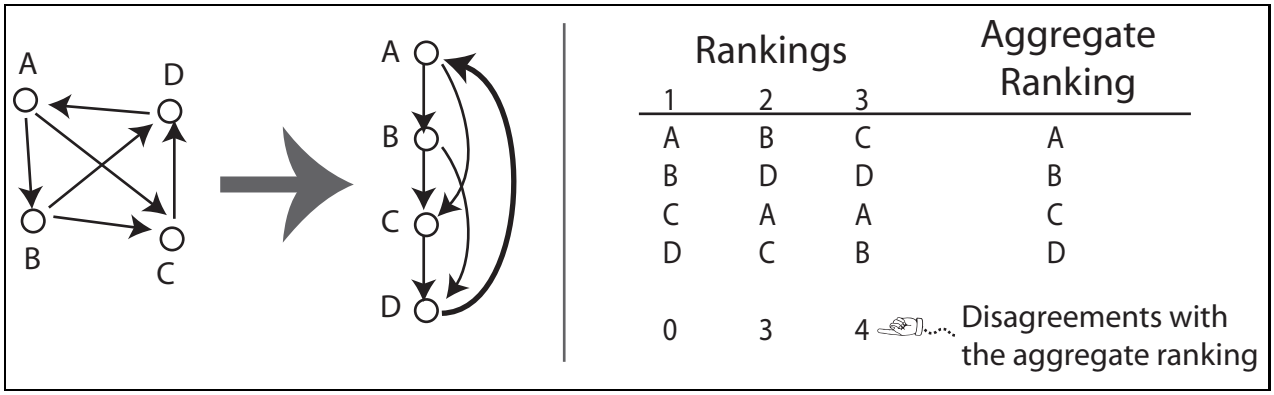

Figure 2: Feedback arc set tournament (left) and Kemeny rank aggregation (right) problems.

Online learning. An online decision problem (ODP) consists of a series of rounds, during each of which an agent chooses one of n pure actions and receives a reward corresponding to its choice. For example the agent may be playing a matrix game such as rock-paper-scissors or chicken repeatedly. The agent's objective is to maximize its cumulative rewards. It can work towards this goal by abiding by an online learning algorithm, which prescribes a possibly mixed action (i.e., a probability distribution over the set of pure actions) to play each round, based on past actions and their corresponding rewards. No-internal-regret (NIR) learners converge to the set of correlated equilibria in repeated matrix games. However, standard NIR learning algorithms involve a fixed point calculation during each round of learning, which is time-consuming when the number of pure actions available to the player is large. **Amy Greenwald, Zheng Li and I [COLT '08] discovered a natural NIR learner that only requires a matrix-vector multiplication, improving the runtime.**

Other Results. See also my **single-authored paper on parallel computation of strongly connected components [SPAA '08]** and my paper on **online correlation clustering with Claire Mathieu and Ocan Sankur [STACS '10].**

Research plans

I expect to do research in learning theory, theoretical AI, and approximation algorithms in the next few years. A confidential description of several concrete research plans is available upon request.