

Designing Algorithms and the Fairness Criteria They Should Satisfy

Vincent Conitzer

Algorithms are increasingly used to determine allocations of scarce, high-value resources. For example, *spectrum auctions*, which are used by governments to allocate radio spectrum, require algorithms to determine which combinations of bids can and should be accepted. *Kidney exchanges* allow patients that require a kidney transplant and have a willing but medically incompatible donor to trade their donors, and some of these exchanges now use algorithms to determine who matches with whom. These are very different application domains—for one, in the former transfers of money play an essential role, but in the latter they are illegal. Other applications have yet different features, so each comes with its own requirements.

In spite of the lack of a single, universal solution, there are several clear benefits of allocating resources algorithmically. In both examples above, there is a combinatorial explosion in the space of possible alternatives, and computers are much better able to search through these spaces. If the algorithm is clearly specified beforehand, this can also improve trust in the process as a whole. On the other hand, the process of designing the algorithm often brings into sharp focus that the objective is not clear. Should a government running a spectrum auction focus on the spectrum being allocated efficiently, or on bringing in revenue? In a kidney exchange, should we simply maximize the number of transplants, or give some priority to certain patients, for example ones who will be hard to match in the future due to blood type?

Without answers to these questions, it is hard to design the algorithm; even if we avoid explicitly answering them, any choice of algorithm implicitly corresponds to a decision about how much to prioritize each aspect. On top of that, different objectives often require algorithms that are quite different in nature, even in the same application domain—and some objectives will not allow a sufficiently efficient algorithm. As a result, having a clean division of labor between computer scientists who design the algorithms, and others (policy makers, economists, doctors, ethicists) who determine the objective(s) to optimize is generally not feasible. They need to talk with each other.

If there is one objective that is often both essential and difficult to make precise, it is that of *fairness*. Optimizing some straightforward criterion often results in outcomes that people intuitively perceive to be unfair. To make matters worse, it is often difficult for them to put their finger on precisely *what* makes these outcomes unfair. They may be able to verbalize it to some

extent, but it will generally fall short of a precise mathematical criterion.

The paper that follows focuses on the specific problem of *rent division*, where we have n people renting an apartment together that costs B to rent. There are n bedrooms to assign among them, and the rooms are not all the same, so it may make sense to split the rent unequally. However, the renters do not agree on how much each room is worth. Given how much each person values each room, how should the rent be split? This is the problem that the authors set out to solve. They are motivated in part by the website Spliddit, which was developed by some of the authors to make tools for various kinds of fair division problem, including rent division, broadly available. While the rent division problem is important in its own right, in my mind the bigger contribution of the paper is as a great case study in how to address the types of problem discussed earlier. What does it mean for a rent division to be fair, and is there an efficient algorithm for computing such rent divisions?

What makes the paper stand out is the variety of techniques applied to arrive at a solution. The authors are guided by theory, including the Second Welfare Theorem from mathematical economics. But they also do a user study, with Spliddit users as the subjects. And of course, they design an efficient algorithm for the problem, which is now deployed on Spliddit.

Algorithms make increasingly many decisions about allocations of resources to people, as well as other decisions affecting people's lives—for example, machine learning classifiers determining whether someone is released on bail. The type of multidisciplinary approach in the following paper—combining techniques from economic theory, the behavioral sciences, and computer science—is essential for steering these developments in the most beneficial direction.