# 3

# Games People Play (With Algorithms)

## THE DATING GAME

In 2013, a journalist named Amanda Lewis wrote an insightful article for *LA Weekly* about her experiences with a recently launched online dating app named Coffee Meets Bagel. One of the app's novelties was to apply the notion of economic scarcity to romantic matchmaking. Instead of encouraging users to indiscriminately spam potential dates with a barrage of online flirts, nudges, and winks, Coffee Meets Bagel limited users to a single, algorithmically proposed match or date each day, which they could accept or reject. Presumably the idea was to raise the value or demand for matches by artificially restricting the supply.

But Lewis went on to detail other "economic" side effects of the app that were perhaps less intentional, and less desirable—side effects that can be understood via game theory, the branch of economics that

deals with strategic interactions between groups of self-interested individuals. Coffee Meets Bagel invited users to specify racial, religious, and other preferences in their matches, which the algorithm would then try to obey in selecting their daily proposals.

Lewis described how after not specifying any racial preferences (or, more precisely, indicating that she was willing to be matched with people from any of the site's listed racial groups), she began to receive daily matches exclusively with Asian men. The problem was that if there were even a slight imbalance in the number of women who accept matches with Asian men, and the number of Asian men, there would be an oversupply of Asian men in the app's user population. And since the matching algorithm obeys users' stated preferences, the necessary consequence was that women who did not explicitly *exclude* Asian men from their preferences would be matched with them frequently.

Given the preferences selected by the rest of the user population, Lewis's "best response" (a game theory term)—that is, her only choice if she wanted to be matched with men from other races too—was to modify her stated preferences to say that she was *unwilling* to be matched with Asian men. She reluctantly did so, even though this was not what she originally wanted. Of course, this only exacerbates the original oversupply problem, creating a feedback loop that encourages other users to do the same.

It seemed as though Lewis had been cornered into choosing between two undesirable alternatives—cornered by the stated preferences of other users, and by an algorithm that blindly and myopically obeyed those preferences for each user individually, without regard for the macroscopic consequences. At least from Lewis's perspective, the system was trapped in what a game theorist might call a "bad equilibrium." If all of the users of the app could have simultaneously coordinated to change their preferences, they might all have been happier with their resulting set of matches—but each of them individually was helpless to escape this bad outcome. It's a bit like a run

on the banks in a financial crisis—even though it makes us all collectively worse off, it's still in your selfish interest to withdraw your money before it's too late.

## WHEN PEOPLE ARE THE PROBLEM

There are some important similarities and differences between the dilemma Lewis found herself in on Coffee Meets Bagel and the problems of fairness and privacy considered in earlier chapters. In all three settings, algorithms play a central role—algorithms acting on, and often building predictive models from, people's data. But in algorithmic violations of fairness and privacy, it seemed reasonable to view the algorithm as the "perpetrator" and people as the "victims," at least to a first approximation. As we saw, machine learning algorithms optimizing solely for predictive accuracy may discriminate against racial or gender groups, while algorithms computing aggregate statistics or models from behavioral or medical data may leak compromising information about specific individuals. But the people themselves were not conspirators in these violations of social norms—indeed, they may not even be aware that their data is contributing to a credit scoring or disease prediction model, and may not interact with those models at all. And since the problems we identified were largely algorithmic, we could propose algorithmic solutions that were better behaved.

The Coffee Meets Bagel conundrum is more nuanced. We might argue that Lewis is also a victim of sorts—she recounts feelings of guilt when the algorithm forces her to declare what feel like racist preferences, just in order to avoid being always matched with a homogeneous group. It seems unfair in a way not dissimilar to algorithmic discrimination. But the key difference is that we can no longer place the blame exclusively or even largely on the algorithm alone—the other users, and their competing preferences, are complicit in Lewis's dilemma. After all, it wasn't the algorithm's fault that there were too

many Asian men in the system relative to the population of women who reported a willingness to date them. The algorithm was simply trying to act as a mediator of sorts, attempting to satisfy each user's dating preferences in light of those of other users. We might even say that the algorithm was doing the most obvious and natural thing with the data it was given, and that the real problem was the data—the preferences themselves.

We'll eventually see that despite the complicity of users, we shouldn't let algorithms off the hook quite so easily, and that in many settings in which user preferences are centrally involved, there are still algorithmic techniques that can avoid the bad equilibrium in which Lewis became trapped. In particular, sometimes there might be multiple equilibria, and an algorithm might be able to choose, or nudge its users toward, a better one. In the case of Coffee Meets Bagel, maybe *everyone*'s preferences were like Amanda's—wanting only a diversity of matches—and everyone felt trapped into entering preferences that weren't quite truthful. Maybe a different algorithm could have done better and incentivized everyone to enter their real preferences. And in other settings we might prefer an algorithm that doesn't encourage or implement any equilibrium at all, but instead finds a solution that makes the overall "social welfare" higher. But unlike the fairness and privacy chapters, to discuss these algorithmic alternatives, we need to put the users, and their preferences, on center stage. And this in turn leads us to the powerful concepts and tools of game theory.

## JUMP BALLS AND BOMBS

Many readers may have encountered a little game theory, owing in part to its generality and its ability to sometimes generate counterintuitive insights about everyday scenarios. Informally speaking, an equilibrium in game theory can be described as a situation in which all participants are acting in their own self-interest, given what everyone

else is doing. The key aspect of the definition—which we'll make a bit more precise when it's called for—is the notion of selfish, unilateral stability it embodies. It is assumed that each "player" in the system (such as a user of Coffee Meets Bagel) will behave selfishly (for example, by setting or changing her dating preferences) to advance her own goals, in response to similarly selfish behavior by others, and without regard to the consequences for other players or the global outcome.

An equilibrium is thus a kind of selfish standoff, in which all players are optimizing their own situation simultaneously, and no one can improve their situation by themselves. Technically speaking, the underlying mathematical notion of equilibrium we refer to here is known as a Nash equilibrium, named for the Nobel Prize–winning mathematician and economist John Forbes Nash, who proved that such equilibria always exist under very general conditions. We'll shortly have reason to consider non-equilibrium solutions to game-theoretic interactions, as well as alternative notions of equilibrium that are more cooperative.

When equilibrium is described as a selfish standoff, it's not particularly surprising that sometimes equilibrium can be undesirable to any particular individual in the system (like Amanda Lewis), or even to the entire population. In the words of the late economist Thomas Schelling (another Nobel Prize winner), who applied equilibrium analysis to things as varied as housing choices, traffic jams, sending holiday greeting cards, and choosing a seat in an auditorium, "The body of a hanged man is in equilibrium, but nobody is going to insist the man is all right."

While the competitive, selfish nature of our equilibrium notion might seem a bit cynical or depressing—everyone is simply out for themselves, and optimizing their choices and behavior in light of everyone else's greedy behavior—it can also provide valuable clues to why and how things can sometimes go wrong in settings in which there are conflicting preferences (like racial preferences in a dating

app). And it does not preclude cooperative behavior if there just so happens to be a solution in which cooperation is in everyone's self-interest. Closest to our own selfish interests, it turns out that sometimes game theory can not only describe what might go wrong at equilibrium but also can provide algorithmic prescriptions for making the outcome better.

For much of its long and storied history—depending on how one counts, the field dates to at least the 1930s—game theory trafficked primarily in the precise understanding of simple and highly stylized versions of real-world problems. These stylizations could often be described by small tables of numbers specifying the payoffs of just two players (and therefore their preferences, since it is assumed that a player will always prefer whatever offers their highest payoff, in light of the opponent's behavior). Classic examples include Rock-Paper-Scissors (useful in the real world as an alternative to jump balls in recreational basketball), where, for instance, choosing Rock yields payoff +1 against Scissors, which in turn receives payoff -1. The equilibrium turns out to be both players uniformly randomizing among their choices, playing Rock, Paper, and Scissors with probability 1/3 each. This is the only solution with the aforementioned unilateral stability property—if I uniformly randomize, your best response is to do so as well, and if you do anything else (such as playing Paper even slightly more often than the other two choices), I'll exploit that and punish you (by always playing Scissors). Some readers may be even more familiar with Prisoner's Dilemma, another simple game that has a disturbing equilibrium in which both players sabotage each other to their mutual harm, even though there is a cooperative non-equilibrium outcome in which they both benefit. As the story goes, two accomplices to a crime are captured and held in separate cells. They can either "cooperate" with their accomplice and admit to nothing or "defect" and admit to the crime and testify against their partner. If your partner defects and testifies against you, you get a long sentence. If your partner

cooperates, you get only a short one. And if you defect, the prosecutor offers to shave a little bit off the sentence you would have otherwise received. The problem is that if I cooperate, you can do even better by sabotaging me and defecting, and vice versa. When we both defect, we each experience close to the worst possible outcome. But since mutual cooperation is not unilaterally stable, we drag each other into the sabotaging abyss of equilibrium, hence the "dilemma".

Despite the simplicity of such games, they have occasionally been applied to rather serious and high-stakes problems. During the Cold War, researchers at the RAND Corporation (a long-standing think tank for political and strategic consulting) and elsewhere used game-theoretic models to try to understand the possible outcomes of US-Soviet nuclear war and détente—efforts that were memorably if darkly lampooned in the 1964 Stanley Kubrick film *Dr. Strangelove*, which ends with the Prisoner's Dilemma–like nuclear annihilation of the world. But the lasting influence and scope of game theory (which has also been widely applied to evolutionary biology and many other fields far from its origins) bears testament to the value of deeply understanding a "toy" version of a complex problem. By distilling strategic tensions down to tables of numbers with maybe only a few rows and columns, game theorists could solve exactly for the equilibrium and try to understand its ramifications for the real problem—which was usually considerably more complicated, messy, and imprecise.

As we shall see, the technological revolution of the last two decades has considerably expanded the scope and applicability of game-theoretic reasoning, while at the same time challenging the field to tackle problems of unprecedented scale and complexity—problems involving sophisticated algorithms operating on rich datasets generated by thousands, millions, or sometimes billions of users. Reducing such problems to simple models of the Rock-Paper-Scissors or Prisoner's Dilemma variety is entirely infeasible and would throw away too much salient detail to be even remotely useful. The matchmaking

equilibrium determined by the dating preferences of the users of Coffee Meets Bagel simply isn't something that can be computed by hand and understood with just a few numbers. It would itself require an algorithm to compute, which in an informal sense is exactly what the app provides.

To tackle such challenges, the new field of algorithmic game theory has emerged and developed rapidly. It blends ideas and methods from classic game theory and microeconomics with modern algorithm design, computational complexity, and machine learning, with the goal of developing efficient algorithmic solutions to complex strategic interactions between very large numbers of players. At a minimum, it aspires to broadly understand what might happen in systems like Coffee Meets Bagel. At its best, it is not only descriptive but also prescriptive—as in the fairness and privacy chapters, telling us how to design socially better algorithms, but now in settings in which the incentives and preferences of users, and how we will examine in act on them, must be taken into account. These are the topics we will examine in this chapter.

## THE COMMUTING GAME

To illustrate how the scale and power of modern technology have made algorithmic game theory relevant, let's consider an activity that many people engage in every day but may never have thought about as a "game" before: driving a car. Suppose you live in a busy metropolitan area with congested roads, and each day you must drive from your house in the suburbs to your workplace downtown. There is a complex network of freeways, highways, streets, thoroughfares, and back roads you must navigate, and the number of plausible routes you could take might be very large indeed. For instance, maybe the most straightforward route is to get on the freeway at the entrance nearest your home, get off at the exit nearest your workplace, and drive on the main surface streets before and after the freeway. But maybe one

segment of the freeway often has bumper-to-bumper traffic during your commute time, so sometimes it's better to get off earlier, take some back roads through a residential neighborhood, and rejoin the freeway later. And on any given day, transient conditions—a traffic accident, road construction or closures, a ball game—might render your usual route much slower than some other alternative.

If you think about it, on a moderately long commute in a busy city the number of distinct routes you might take or at least try over time could be in the dozens or even hundreds. Of course, these different routes might overlap to varying degrees—maybe many of them use the freeway, and if you live on a cul-de-sac, they will always start by getting off your street—but each route is a distinct path through your local network of roads. In game theory terminology, your "strategy space"—the possible actions you might choose—is much larger than in simple games like Rock-Paper-Scissors, where by definition you only have three actions available.

So you have a lot of choices; but what makes this a "game"? It's the fact that if you're like most commuters, your goal or objective is to minimize your driving time. But the driving time on each of your many possible routes depends not just on which one you choose but also on the choices of all the other commuters. How crowded each route is determines your driving time as much or more than the length of the roads, their traffic lights, speed limits, and other fixed aspects. The more drivers who choose a given road, the longer the driving time for all routes that use that road, making them less attractive to you. Similarly, the fewer drivers there are on a road, the more you might want to choose a route that uses it (as long as the other segments on the route aren't too busy).

The combination of your hundreds of possible routes with the choices made by the tens of thousands of other commuters presents you with a well-defined, if mind-boggling, optimization problem: pick the route with the lowest total driving time, given the choices of

all the other drivers. This is your "best response" in the commuting game. And it's not at all unreasonable for us to assume you will at least try to act selfishly and choose your best response (just as Amanda Lewis begrudgingly did on Coffee Meets Bagel). Who wants to spend more time commuting than they need to?

Note that while the complexity of this game is much greater than something like Rock-Paper-Scissors, the fundamental commonality is that the payoff or cost of any individual player's choice of action depends on the action choices of *all* the players. There are important differences as well. In Rock-Paper-Scissors the two players have the same payoff structure, whereas if you and I live and work in different places, our cost structures will differ (even though we still both want to minimize driving time for our own commute). And if you and I commute at different times of day, we aren't really even in the same round of the game. But these differences don't alter the fundamental view of commuting as just another (albeit very complex) game. And this means that as with Rock-Paper-Scissors and Coffee Meets Bagel, it makes sense—from both qualitative and algorithmic perspectives—to discuss its equilibrium, whether it is "good" or "bad," and whether there might be a better outcome.

## YOUR SELFISH WAZE

Commuting has been the game we have described ever since roads became sufficiently congested that the choices of other drivers affected your own. But for many years this formulation wasn't particularly relevant, because people really didn't have the ability to truly or even approximately optimize their route based on the current traffic. Commuting was a game, but people couldn't play it very well. This is where technology changed everything—and, as we shall see, not necessarily for the collective good.

The first challenge in playing the commuting game is informational. As older readers will recall, for decades you had to plan your daily commute by cobbling together radio and television traffic reports that were both incomplete (perhaps covering only major freeways, and providing little or no information about the vast majority of roads) and inaccurate (since the reports were only occasional, perhaps on the half hour, and therefore often stale). But even if one could magically always have perfect and current traffic data for every road, there is a second, algorithmic challenge, which is computing the fastest route between two points in a massive network of roadways, each annotated by its current driving time.

In a relatively short period, navigation apps such as Waze and Google Maps have effectively solved these problems. The algorithmic challenge was actually the easier one—there have long been fast, scalable algorithms for computing fastest routes (or "shortest paths," as they are called in computer science) from known traffic. A classical one is Dijkstra's algorithm, named for the Dutch computer scientist who described it in the late 1950s. Such algorithms in turn allowed the informational problem to be solved by crowdsourcing. Even though early navigation apps operated on traffic data not much better than in the pre-Internet days, they could still at least suggest plausible routes through a complex and perhaps unfamiliar city—a vast improvement over the era of dense and confusing fold-up maps in the glove compartment. And once users started adopting the apps and permitting (wittingly or not) their location data to be shared, the apps now had thousands of real-time traffic sensors right there on the roadways.

This crowdsourcing was the true game-changer. Whatever pride you might have had in your navigational wizardry in your home city, the utility of a tool that automatically optimized your driving time in response to real-time, highly accurate, and granular traffic data on virtually every roadway anywhere was just too alluring to decline. User populations grew to the hundreds of millions, further improving traffic data coverage and accuracy.

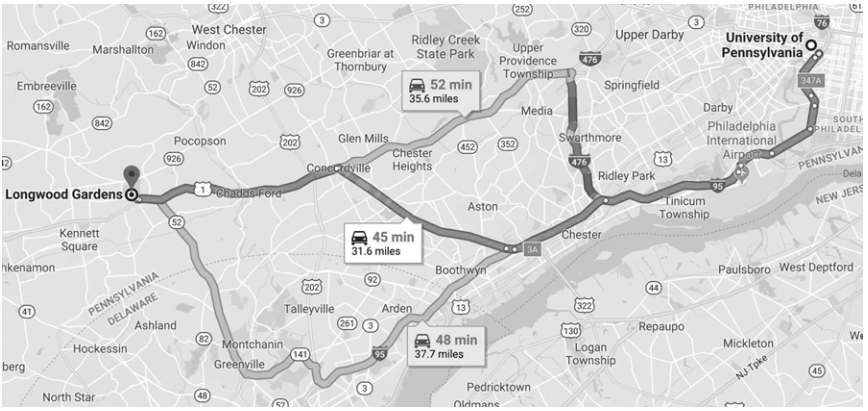


**Fig. 17.** Typical screenshot from Google Maps, showing just a few of the many hundreds or thousands of routes between two locations in the greater Philadelphia area. The suggested routes are ranked by lowest estimated driving time.

From our game-theoretic viewpoint, modern navigation apps finally allowed any player in the commuting game to compute her best response to all her "opponents" on the roads, anywhere and anytime. And there is little doubt that these apps are extraordinarily useful and efficient, and are doing the most obvious thing with the massive data at their disposal: looking out for the best interests of each individual user, finding their fastest route in light of the current traffic patterns.

## THE MAXWELL SOLUTION

But there is another perspective worth considering, which is that because the apps are computing best responses for every player individually, they are driving the collective behavior toward the kind of competitive equilibrium we have discussed in Coffee Meets Bagel, Prisoner's Dilemma, and Rock-Paper-Scissors—the apps enable, and thus encourage, selfish behavior on everyone's part. And with Coffee Meets Bagel and Prisoner's Dilemma, we already have seen cases in which the resulting competitive equilibrium may not be something that any particular individual is happy about. Surely anyone with even moderate experience with city driving has encountered situations in

which individually selfish behavior by everyone seems to make everyone worse off—for instance, in the jockeying and slithering that occur when merging down to a few lanes of traffic at the entrance to the Lincoln Tunnel in New York City.

What might be an alternative to individually selfish, collectively competitive driving? Surely no one believes we'd all be better off (at least in terms of driving time) if we rolled back the calendar and returned to the era of spotty traffic reports and folding maps. But now that we do have large-scale systems and apps with the ability to aggregate granular traffic data, compute and suggest routes to drivers, it might be worth considering making recommendations other than the obvious, selfish ones.

Let's consider a conceptually simple thought experiment. Imagine a new navigation app—we'll name it Maxwell, for reasons that will become clear later—that behaves similarly to Google Maps and Waze, at least at a high level. Like those apps, Maxwell gathers GPS and other location data from its users to create a detailed and up-to-date traffic map, and then for any user at any moment, it computes and suggests a driving route based on origin, destination, and the traffic. But Maxwell is going to use a very different algorithm to compute suggested routes—an algorithm with a different goal, and one that will lead to a different and better collective outcome than the competitive equilibrium.

Instead of always suggesting the selfish or best response route to each user in isolation, Maxwell gathers the planned origin and destination of every user in the system and uses them to compute a coordinated solution that is known in game theory as the maximum social welfare solution (hence the app's name, Maxwell). In the commuting game, the maximum social welfare solution is the one that minimizes the *average* driving time across the entire population, instead of trying to minimize the driving time of each user *individually* in response to the current traffic. By minimizing average driving time, Maxwell is maximizing the time people have to do other things, which is presumably a good thing.

It might seem like there shouldn't be any difference between these two solutions, but there is. A stylized but concrete example will be helpful here. Imagine that there is a large population of $N$ drivers in a city, and all of them want to simultaneously travel from location A to location B. There are only two possible routes from A to B; let's call them the slow route and the fast route.

The slow route passes many schools, hospitals, libraries, restaurants, shops, and other places that generate a great deal of pedestrian traffic. It is littered with stop signs, crosswalks, speed bumps, and police making sure that all laws are obeyed. Because of this, it really doesn't matter how many drivers take the slow route. The real bottleneck is all the stop signs, crosswalks, speed bumps, and police. In other words, we are going to assume that the time it takes to travel from A to B on the slow route is independent of the number of drivers on it. To make things concrete, let's suppose that travel time is exactly one hour.

The fast route, on the other hand, is a freeway without speed limits or police, but it has limited capacity. If you're the only one driving on it, it can be very fast—almost instantaneous—to get from A to B. But the more drivers who take the fast route, the less fast it becomes. Specifically, let's assume that if $M$ out of the $N$ drivers take the fast route, the travel time for each of them is $M/N$ hours. Since $M$ is a whole number less than or equal to $N$, this means that the time it takes to travel the fast route is between $1/N$ (if only one driver takes

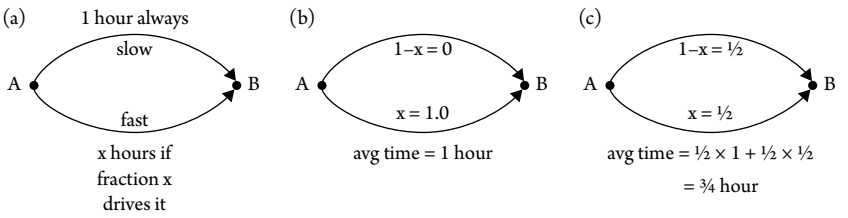


**Fig. 18.** Illustration of simple two-route navigation problem, with a fixed-driving-time slow route and a traffic-dependent fast route (a); equilibrium or Waze solution (b); Maxwell solution (c).

it), which is nearly zero if $N$ is large, and $N/N$, which is one hour if everyone takes it. So in the worst case, the fast route isn't any faster than the slow route, but it depends on $M$. From your perspective as a driver, you'd like all the other $N - 1$ drivers to take the slow route, taking exactly an hour each, and for you to take the fast route and virtually teleport to the destination. Of course, none of the other drivers like your solution.

Now let's analyze the consequences of selfish behavior, of the kind enabled and even encouraged by existing navigation apps. If we think about it, such apps will recommend the fast route to the entire population of $N$ drivers. This is because if the app recommended the slow route to even a small number of drivers—say, five—then these five drivers all experience the fixed one hour of slow route travel time, but any one of them would have been slightly better off taking the fast route, where the travel time will be $(N - 5)/N = 1–5/N$ hours—just a shade less than an hour. So the competitive equilibrium that results from selfish routes is where everyone takes the fast route, which then becomes no faster than the slow route, and everyone's driving time is exactly one hour. Note that in this equilibrium, each individual driver is actually indifferent to which route is taken—the driving time for both is an hour—but if even one driver is on the slow route, the drivers on the fast route are strictly better off.

What is Maxwell going to do in the same situation? It is going to pick half of the drivers—let's say a random half—and suggest that they drive the slow route, and suggest the fast route to the other half. Before discussing why anyone would follow the suggestion to drive the slow route, let's analyze the average driving time in this alternative solution. Obviously the $N/2$ drivers taking the slow route will, as always, experience a driving time of one hour. The $N/2$ drivers taking the fast route will experience a driving time of only $(N/2)/N = 1/2$ hour each. So the average driving time across the entire population is $(1/2 \times 1) + (1/2 \times 1/2) = 3/4$ of an hour, or only forty-five minutes. It turns out this is the split of the population into the slow and fast route that minimizes the

average driving time. (For readers who both took and remember some calculus, if we let $x$ denote the fraction of the population on the fast route, the average driving time is simply $1 - x + x^2$, which is minimized at $x = 1/2$ and yields the 3/4 hour average.)

In other words, by suggesting routes with a different goal—one with an explicit concern for the collective benefit rather than individual self-interest—we can reduce the overall driving time significantly, by 25 percent in this case. And we can do so without making anyone worse off than they would have been in the competitive equilibrium. So there is a better alternative to the competitive equilibrium in our toy example, and the gains may generally be even greater in complex networks of roads in the real world.[1] (In 2018 a team of researchers from UC Berkeley presented empirical evidence that navigation apps indeed cause increased congestion and delays on side streets.) The question is whether and how we can actually realize this savings of collective driving time "in the wild."

## MAXWELL'S EQUATIONS

The first challenge in implementing Maxwell is algorithmic. While it was a simple calculus exercise to find the socially optimal solution in our toy two-route example, how will Maxwell do it when confronted with colossal networks of real roads and freeways, and thousands or more drivers, all with different origins and destinations? At least the selfish routes suggested by Google Maps and Waze can be computed quickly on large-scale networks, using Dijkstra's algorithm.

Fortunately, it turns out that there are also fast, practical algorithms for computing the global solution that minimizes collective average

[1] A distinct but related side effect of selfish behavior in commuting is known as Braess's paradox, in which adding capacity to a network of roadways actually *increases* congestion (or closing roads decreases congestion), and which has been reported to have occurred in large cities such as Seoul, Stuttgart, and New York City. Such phenomena cannot occur under the Maxwell solution.

driving time in large networks, especially if the driving times on each road are a linear (i.e., proportional) function of the number or fraction of drivers on them (like the roads in our earlier example, or more realistic ones such as a road that hypothetically takes $1/4 + 2x$ hours to travel if a fraction $x$ of the population drives on it). This proportional model actually seems like a reasonable one for real traffic, and we can easily envision deriving such models from the voluminous empirical data that services such as Waze already routinely collect, which provides samples of the driving times at different levels of traffic. And for such roads, the average driving time is then just a quadratic function (e.g., if a fraction $x$ of drivers takes a $1/4 + 2x$ road, then the contribution to the overall average driving time from just this road will be $(1/4 + 2x)x = 1/4x + 2x^2$).

Even though Maxwell must solve a very high-dimensional problem—finding the exact fractions of drivers taking every road in the network, in a way that is consistent with everyone's origins and destinations and is socially optimal—it is a problem of a well-studied and well-understood type that has very practical algorithms. It is an instance of what are known as convex minimization problems, which can be solved by so-called gradient descent methods; this is just algorithm-speak for "walk downhill in the steepest direction to quickly get to the lowest point in the valley." In our context, this simply means that we start with an arbitrary assignment of driving routes and make incremental improvements to it until the collective driving time is minimized.

What if the driving times on the roads are not proportional to traffic but are more complex functions? For example, consider a hypothetical road whose driving time is $x/2$ for $x < 0.1$ but is $10x + 2$ for $x >= 0.1$. So the time it takes to drive this road takes a sudden, discontinuous jump once 10 percent or more of the population takes it. For more complex roads such as these, we do not know of fast algorithms that are always guaranteed to find the socially optimal solution, but we do know of good techniques that work well in practice. And in these

more complex cases, the improvement of the socially optimal solution over the selfish equilibrium can be much greater than in the proportional-road setting. Thus at least the algorithmic challenges in implementing Maxwell seem surmountable.

## CHEATING ON MAXWELL

But as is often the case in settings in which human preferences and game theory are involved, the biggest challenge Maxwell would face in the real world has less to do with good algorithms and more to do with incentives. Specifically, why would anyone ever follow the advice of an app that sometimes doesn't suggest the route that would be fastest for him at that given moment? Consider any particular driver assigned to the slow route in the earlier example—he could always "defect" to the fast route and reduce his driving time, so why wouldn't he? And if everyone did this, they all revert to the competitive equilibrium of existing apps.

If we think about it for a moment, it seems possible that even current navigation apps such as Google Maps and Waze could also be susceptible to various kinds of cheating or manipulation. For example, I could lie to Waze about my intended origin and destination, in an effort to influence the routes it recommends to other users in a way that favors me—creating false traffic that causes the Waze solution to route other drivers away from my true intended route. Manipulation of this variety apparently occurred in residential Los Angeles neighborhoods frustrated by the amount of Waze-generated traffic, as reported by the *Wall Street Journal* in 2015:

> Some people try to beat Waze at its own game by sending misinformation about traffic jams and accidents so it will steer commuters elsewhere. Others log in and leave their devices in

their cars, hoping Waze will interpret that as a traffic standstill and suggest alternate routes.

But the incentive problems that Maxwell faces are arguably even worse, because they are not simply about drivers lying to the app; rather, the problem is drivers disregarding its recommendations entirely when they are not best responses.

There are a couple of reasonable replies to this concern. The first is that we may eventually (perhaps even soon) arrive at an era of mostly or even entirely self-driving cars, in which case the Maxwell solution could simply be implemented by centralized fiat. Public transportation systems are generally already designed and coordinated for collective, not individual, optimality. If you want to fly commercially from Ithaca, New York, to the island town of Lipari in Italy, you can't simply direct American Airlines to take a nonstop route along the great circle between the two locations—instead you'll have multiple flight legs and layovers, all for the sake of macroscopic efficiency at the expense of your own time and convenience. In a similar vein, it would be natural for a massive network of self-driving cars to be coordinated so as to implement navigation schemes that optimize for collective average driving time (and perhaps other considerations, such as fuel efficiency) rather than individual self-interest.

But even before the self-driving cars arrive en masse, we can imagine other ways Maxwell might be effectively deployed. One is that if, as in our two-route example above, Maxwell randomly chooses the drivers who are given nonselfish routes, users might have a stronger incentives to use the app, since over time the assignment of nonselfish routes will balance out across users, and then each individual user would enjoy lower average driving time. So while you might have an incentive to disregard Maxwell's recommendation of a slower route on any given trip (which you might well discover by using Google Maps to see your selfish best-response route), you know that over

time you will benefit from following Maxwell's suggestions (as long as others do as well). We might call this phenomenon cooperation through averaging, which is also known to occur when human subjects play repeated rounds of Prisoner's Dilemma. But perhaps there is a better and more general solution to these incentive concerns.

## COOPERATION THROUGH CORRELATION

Let's review where we are. Maxwell may have a better collective solution, but it is vulnerable to defection, and even the selfish navigation apps may be prone to manipulation. Both approaches have good algorithms, but the concern is that their goals can be compromised by human nature.

It turns out that sometimes these concerns can be overcome by considering yet a third notion of solution in games (our first being the selfish equilibrium, and the second being the best social welfare but non-equilibrium Maxwell solution). This third notion is known as a correlated equilibrium, and it too can be illustrated by a simple situation involving driving. Imagine an intersection of two very busy roads, one of which has a yield sign and the other of which does not. Then not only the law but also the selfish equilibrium is for drivers on the yielding road to always wait for an opening before continuing, and for drivers on the through road to speed along. Given what the drivers on the other road are doing, everyone is following their best response. But drivers on the yielding road suffer all the waiting time, which might feel unfair to them.

In this example a correlated equilibrium could be implemented with a traffic signal, which now allows drivers to follow strategies that depend on the signal, such as "If the light shows green to me, I will speed through, and if it shows red to me, I will wait." If everyone follows this strategy, they are all best-responding, but now the waiting time is split between the two roads—a fairer outcome not possible with

only yield signs. The traffic signal is thus a coordination (or correlating) device that allows cooperation to become an equilibrium.

Can cooperation via coordination help solve Maxwell's incentive problems? The answer is yes—at least in principle. Very recent research has shown how it is possible to design a variant of Maxwell's algorithm—let's call it Maxwell 2.0—that quickly computes a correlated equilibrium, and enjoys three rather strong and appealing incentive properties. First, it is in the best interest of any driver to actually use Maxwell 2.0: nobody has an incentive to opt out and use another app instead (unlike Maxwell 1.0). Second, it is in the best interest of any driver to honestly input his true origin and destination: one cannot beneficially manipulate the solution found by Maxwell 2.0 by lying to it. (This is a property known as "truthfulness" in game theory.) Third and finally, it is in the best interest of any driver to actually follow the route recommended by Maxwell 2.0 after he sees it. So all drivers want to use Maxwell 2.0, and to use it honestly—both in what they input to it and in following its output.

How does Maxwell 2.0 achieve these apparently magical properties? Looking back to Chapter 1, it does so by applying differential privacy to the computation of the recommended routes in a correlated equilibrium. Recall that differential privacy promises that the data of any individual user cannot influence the resulting computation by very much. In this case a user's data consists of both the origin and destination he reports to Maxwell 2.0 and the traffic data his GPS locations contribute. The computation in question is the assignment of driving routes in a correlated equilibrium. Since a single driver's data has little influence, it means manipulations like lying about where you want to go or leaving the app on in your parked car won't benefit you or change what others do. And since a correlated equilibrium is being computed, your best response is to follow the suggested route.

Note that there was no explicit privacy goal here. Rather, the incentive properties we desired were a by-product of privacy. But at a

high level, it makes sense: if others can't learn anything about what you have entered into Maxwell 2.0 or what it told you to do, then you can't beneficially manipulate your inputs to change the behavior of others. That techniques developed for one purpose (like privacy) turn out to have applications elsewhere (like incentivizing truthfulness) is a common theme in algorithms. In fact, differential privacy has many other non-privacy-related applications—we will see another one in Chapter 4.

## GAMES EVERYWHERE

Even though Maxwell is only a hypothetical app (at least for now), we spent some time on the commuting game because it crisply illustrates a number of more general themes, and does so in a situation with which many of us have daily experience. These themes include:

- Individual preferences (e.g., where you want to drive from and to) that may be in conflict with those of others (e.g., traffic).
- The notion of a competitive or "selfish" equilibrium, and the observation that convenient modern technology (e.g., Waze) might drive us toward this equilibrium.
- The observation that there might be socially better outcomes that can be found with fast algorithms (e.g., Maxwell) that also enjoy good incentive properties (e.g., Maxwell 2.0).
- The lesson that when an app is mediating or coordinating the preferences of its users (as opposed to simply using their data for some other purpose, such as building a predictive model), the algorithm design must specifically take into consideration how users might react to its recommendations—including trying to manipulate, defect, or cheat.

In the rest of this chapter, we'll see that these same ideas apply in a wide variety of other modern, technology-mediated interactions, from routine activities such as shopping and reading news to more specialized

situations such as assigning graduating medical students to hospital residencies, and even to kidney transplants. In some cases we'll see that the algorithms involved might be pushing us toward a bad equilibrium, and in other cases we'll see they are doing social good. But in all of them, the design of the algorithm and the preferences and desires of the users are inextricably intertwined.

## SHOPPING WITH 300 MILLION FRIENDS

Like driving, shopping is another activity that many of us engage in daily and that has been made more social and game-theoretic by technology. Before the consumer Internet, shopping—whether for groceries, plane tickets, or a new car—was largely a solitary activity. You went to physical, local stores, and your decisions were based on your own experience and research (and perhaps the advertising you were exposed to). For bigger purchases such as a car or a television, there might be publications such as *Consumer Reports*. But for most things, you were more or less on your own. As in the commuting game, you had shopping preferences—admittedly more complex, multifaceted, and harder to articulate than simply wanting to drive from point A to point B. But there were very few tools to help you optimize your decisions. It was the shopping equivalent of the era of spotty traffic reports and fold-up maps.

As readers will have experienced, all of this changed with the explosive growth of online shopping. Once we began researching and purchasing virtually everything imaginable on the web, we provided retailers such as Amazon extremely fine-grained data on our interests, tastes, and desires. And as we have discussed in previous chapters, machine learning could then take this data and build detailed predictive models that generalized from the products and services we already did like to the ones we would like if only we were made aware of them. The technical term in the computer science community for this general technology is *collaborative filtering* (which was widely used in the Netflix competition,