The Top 10 List

1. *The second-price single-item auction.* Our first example of an "ideal" auction, which is dominant-strategy incentive compatible (DSIC), welfare maximizing, and computationally efficient (Theorem 2.4). Single-item auctions already show how small design changes, such as a first-price vs. a second-price payment rule, can have major ramifications for participant behavior.

2. *Myerson's lemma.* For single-parameter problems, DSIC mechanism design reduces to monotone allocation rule design (Theorem 3.7). Applications include ideal sponsored search auctions (Section 3.5), polynomial-time approximately optimal knapsack auctions (Theorem 4.2), and the reduction of expected revenue maximization with respect to a valuation distribution to expected virtual welfare maximization (Theorem 5.4).

3. *The Bulow-Klemperer theorem.* In a single-item auction, adding an extra bidder is as good as knowing the underlying distribution and running an optimal auction (Theorem 6.5). This result, along with the prophet inequality (Theorem 6.1), is an important clue that simple and prior-independent auctions can be almost as good as optimal ones.

4. *The VCG mechanism.* Charging participants their externalities yields a DSIC welfare-maximizing mechanism, even in very general settings (Theorem 7.3). The VCG mechanism is impractical in many real-world applications, including wireless spectrum auctions (Lecture 8), which motivates simpler and indirect auction formats like simultaneous ascending auctions (Section 8.3).

5. *Mechanism design without money.* Many of the most elegant and widely deployed mechanisms do not use payments. Exam-

Downloaded from https://www.cambridge.org/core. McGill University Libraries, on 08 Oct 2019 at 14:55:45, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316779309.022

ples include the Top Trading Cycle mechanism (Theorems 9.7 and 9.8), mechanisms for kidney exchange (Theorem 10.1), and the Gale-Shapley stable matching mechanism (Theorems 10.5, 10.7, and 10.8).

6. *Selfish routing.* Worst-case selfish routing networks are always simple, with Pigou-like networks maximizing the price of anarchy (POA) (Theorems 11.1 and 11.2). The POA of selfish routing is therefore large only when cost functions are highly nonlinear, corroborating empirical evidence that network over-provisioning leads to good network performance (Section 12.1).

7. *Robust POA Bounds.* All of the proofs of POA bounds in these lectures are smoothness arguments (Definition 14.2). As such, they apply to relatively permissive and tractable equilibrium concepts like coarse correlated equilibria (Theorem 14.4).

8. *Potential games.* In many classes of games, including routing, location, and network cost-sharing games, players are inadvertently striving to optimize a potential function. Every potential game has at least one pure Nash equilibrium (Theorem 13.7) and best-response dynamics always converges (Proposition 16.1). Potential functions are also useful for proving POA-type bounds (Theorems 15.1 and 15.3).

9. *No-regret algorithms.* No-regret algorithms exist, including simple ones with optimal regret bounds, like the multiplicative weights algorithm (Theorem 17.6). If each agent of a repeatedly played game uses a no-regret or no-swap-regret algorithm to choose her mixed strategies, then the time-averaged history of joint play converges to the sets of coarse correlated equilibria (Proposition 17.9) or correlated equilibria (Proposition 18.4), respectively. These two equilibrium concepts are computationally tractable, as are mixed Nash equilibria in two-player zero-sum games (Theorem 18.7).

10. *Complexity of equilibrium computation.* Computing a Nash equilibrium appears computationally intractable in general. $\mathcal{PLS}$-completeness (Section 19.2) and $\mathcal{PPAD}$-completeness (Section 20.3) are analogs of $\mathcal{NP}$-completeness tailored to provide evidence of intractability for pure and mixed equilibrium computation problems, respectively (Theorems 19.4 and 20.3).

Downloaded from https://www.cambridge.org/core. McGill University Libraries, on 08 Oct 2019 at 14:55:45, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316779309.022