

# An Algorithmic Game Theory Primer\*

Tim Roughgarden<sup>†</sup>

June 21, 2008

## Abstract

We give a brief and biased survey of the past, present, and future of research on the interface of theoretical computer science and game theory.

## 1 Introduction

By the end of the 20th century, the widespread adoption of the Internet and the emergence of the Web had changed fundamentally society's relationship with computers. The primary role of a computer evolved from a stand-alone, well-understood machine for executing software to a conduit for global communication, content-dissemination, and commerce. Two consequences of this phase transition were inevitable: theoretical computer science would respond by formulating novel problems, goals, and design and analysis techniques relevant for Internet applications; and game theory, with its deep and beautiful study of interaction between competing or cooperating individuals, would play a crucial role. Research on the interface of theoretical computer science and game theory, an area now known as *algorithmic game theory (AGT)*, has exploded phenomenally over the past ten years.

The central research themes in AGT differ from those in classical microeconomics and game theory in important, albeit predictable, respects. Firstly in application areas: Internet-like networks and non-traditional auctions (such as digital goods and search auctions) motivate much of the work in AGT. Secondly in its quantitative engineering approach: AGT research typically models applications via concrete optimization problems and seeks optimal solutions, impossibility results, upper and lower bounds on feasible approximation guarantees, and so on. Finally, AGT usually adopts reasonable (e.g., polynomial-time) computational complexity as a binding constraint on the feasible behavior of system designers and participants. These themes, which have played only a peripheral role in traditional game theory, give AGT its distinct character and relevance.

AGT also connects to traditional theoretical computer science in remarkably diverse and rich ways. For example, recent work on auction design has been informed by techniques ranging from primal-dual algorithms to communication complexity; quantitative analyses of game-theoretic equilibria have drawn on tools familiar from approximation algorithm design, such as mathematical

---

\*Revised version of an invited survey for the 5th IFIP International Conference on Theoretical Computer Science, September 2008. Comments and corrections are welcome.

<sup>†</sup>Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Supported in part by NSF CAREER Award CCF-0448664, an ONR Young Investigator Award, and an Alfred P. Sloan Fellowship. Email: [tim@cs.stanford.edu](mailto:tim@cs.stanford.edu).

programming and potential function arguments; and studying the complexity of computing such equilibria has resurrected interest in previously obscure complexity classes that were originally motivated by local search and combinatorial topology problems.

The next three sections touch on the current dominant research trends in AGT, loosely following the organization of the first book in the field [119]; Section 5 highlights a number of prominent open questions. We discuss only (a subset of the) topics studied by “the STOC/FOCS community”; see [4, 74, 102, 150] for alternative perspectives on computer science and game theory.

## 2 Algorithmic Mechanism Design

*Algorithmic mechanism design (AMD)* studies optimization problems where the underlying data (such as a value of a good or a cost of performing a task) is *a priori unknown* to the algorithm designer, and must be implicitly or explicitly elicited from self-interested participants (e.g., via a bid). The high-level goal is to design a protocol, or “mechanism”, that interacts with participants so that *self-interested behavior yields a desirable outcome*.

There is a complex interaction between the way an algorithm employs elicited data and participant behavior. For example, in a “first-price” sealed-bid auction (where the winner pays its bid), bidders typically shade their bids below their maximum willingness to pay, by an amount that depends on knowledge or beliefs about the other bids. In the “second-price” or “Vickrey” variant [157], where the winner pays only the value of the second-highest bid, each participant may as well bid its true value for the good. (Do you see why?)

Nisan and Ronen [118] proposed the systematic study of what can and cannot be efficiently computed or approximated when the problem data is held by selfish agents, and also coined the term “algorithmic mechanism design”. (See [99, 128, 145] for related contemporaneous work in the AI literature.) Auction design is the most obvious motivation for this subfield, but there are many others. See [93] and [45] for overviews of two modern “killer applications” — keyword search auctions and spectrum auctions, respectively; and [117] and [119, Part IV] for a sampling of other applications in economics and computer science, respectively. The economic literature on mechanism design is quite mature (e.g., [81]), but AMD has contributed in several ways. We concentrate here on its emphasis on complexity bounds and worst-case approximation guarantees, but mention additional aspects of AMD at the end of the section.

The technical core of AMD is the following deep question:

(Q1) *to what extent is “incentive-compatible” efficient computation fundamentally less powerful than “classical” efficient computation?*

To translate question (Q1) into mathematics, reconsider the Vickrey auction for selling a single good. Each bidder  $i$  has a private (true) willingness-to-pay  $v_i$  and submits to the auctioneer a bid  $b_i$ . The auction comprises two algorithms: an *allocation algorithm*, which picks a winner, namely the highest bidder; and a *payment algorithm*, which uses the bids to charge payments, namely 0 for the losers and the second-highest bid for the winner. One easily checks that this auction is *truthful* in the following sense: for every bidder  $i$  and every set of bids by the other players, player  $i$  maximizes its “net value” — value for the good, if received, minus its payment, if any — by submitting its true private value:  $b_i = v_i$ . Moreover, no false bid is competitive with truthful bidding for all possible bids by the other players. Assuming all players bid truthfully (as

they should), the Vickrey auction solves the *social welfare maximization* problem, in the sense that the good is allocated to the participant with the highest value for it.

More generally, consider a feasible region  $\Omega$ ,  $n$  participants each with a real-valued private objective function  $t_i(\cdot)$  defined on  $\Omega$ , and a designer objective function  $f(t_1, \dots, t_n)$ . In the Vickrey auction,  $\Omega$  has one outcome per participant (indicating the winner),  $t_i(\omega)$  is  $v_i$  if  $i$  wins in  $\omega$  and 0 otherwise, and  $f$  is  $\sum_i t_i(\omega)$ . Classical optimization would ask: *given the  $t_i$ 's*, optimize the objective function  $f$  over  $\Omega$ . The AMD analog is only harder: simultaneously *determine the private  $t_i$ 's* and optimize the corresponding function  $f$  over  $\Omega$ . Sometimes the latter problem is no more difficult than the former (as with the Vickrey auction) — when is it strictly more difficult?

**Characterizations and the Limits of Approximation.** Question (Q1) is the subject of intense study by the AGT community. We confine our discussion here to mechanisms  $M$  that share the following properties with the Vickrey auction:  $M$  first asks each participant  $i$  for a “bid function”  $b_i(\cdot)$ , hopefully identical to the private objective function  $t_i(\cdot)$ ;  $M$  then invokes an allocation algorithm  $x(b_1, \dots, b_n)$  and a payment algorithm  $\pi(b_1, \dots, b_n)$  to determine an outcome  $\omega$  and payments  $p_1, \dots, p_n$ , respectively; and it is “incentive-compatible” in that truthful reporting always maximizes the resulting “utility”  $t_i(\omega) - p_i$  of a player, no matter what other players do. We call such mechanisms *simple*.<sup>1</sup> Assuming all players bid truthfully (as they should), the allocation algorithm of a simple mechanism faces the traditional version of the underlying optimization problem, with known  $t_i$ 's.

Call an allocation algorithm *implementable* if, for some cleverly chosen payment algorithm  $\pi$ , coupling  $x$  with  $\pi$  yields a (truthful) simple mechanism. For a single-good auction, if  $x$  is the “highest-bidder” allocation algorithm, then defining  $\pi$  as in the Vickrey auction shows that  $x$  is implementable. If  $x$  is the “second-highest bidder” allocation algorithm, then it is not implementable: *no* payment algorithm can be matched with  $x$  to yield a truthful mechanism. (This is not obvious but not hard to prove.) Thus some but not all algorithms are implementable. We can mathematically phrase the question (Q1) as follows: *are implementable algorithms less powerful than arbitrary algorithms for solving fundamental optimization problems?*

This question is interesting for both polynomial-time and computationally unbounded algorithms. In the latter scenario, both strong positive and strong negative results are known. For every mechanism design problem with a sum objective ( $\sum_i t_i(\omega)$ , and weighted variants), a far-reaching generalization of the Vickrey auction known as the “VCG mechanism” [40, 72, 157] proves that the optimal (not necessarily polynomial-time) allocation algorithm is implementable (see e.g. [117]). For problems with non-sum objectives, the optimal allocation algorithm is not generally implementable, even on problem instances of constant size [9, 118]. Thus “computationally easy” and “strategically easy” can, in some settings, mean radically different things.

Far less is known about polynomial-time implementability. Most intriguing are the many mechanism design problems that are derived from an *NP*-complete problem and for which the optimal allocation algorithm is implementable. For these, *any separation between implementable and non-implementable polynomial-time algorithms must be conditional on  $P \neq NP$ , and no such separation is known*. Any resolution of this issue would be conceptually and technically notable: either incentive-compatibility imposes no additional difficulty for a massive class of important mechanism design problems, or else there is a non-trivial way of amplifying (conditional) complexity-theoretic

---

<sup>1</sup>The usual term is “truthful, direct-revelation”. Our restriction to simple mechanisms is partially but not fully without loss of generality; see also Section 5. We also disallow collusion; see e.g. [71] for more on this issue.

approximation lower bounds using information-theoretic strategic requirements.

Understanding the reach of implementable algorithms generally involves two interrelated goals: characterization theorems and approximation bounds (see also [94]).

- (G1) Usefully characterize the implementable allocation algorithms  $x$  for the problem.
- (G2) Prove upper and lower bounds on the best-achievable approximation ratio of an implementable algorithm (subject to polynomial running time, if desired).

The second goal quantifies the limitations of implementable algorithms using a worst-case approximation measure. The first goal aims to reformulate the unwieldy definition of implementability into a form more amenable to (both upper and lower) approximation bounds. Both goals, and especially (G1), seem to grow more complex with the number of independent parameters required to describe the private information of a player.

Versions of (G2) pervade modern algorithmic research: for a given “constrained computational model”, where the constraint can be either computational (as for polynomial-time approximation algorithms) or information-theoretic (as for online algorithms), quantify its limitations for optimization and approximation. Goal (G1) reflects the additional difficulty in AMD that even the “computational model” (of implementable algorithms) induced by strategic constraints is poorly understood — for example, determining whether or not a given algorithm is online is intuitively far easier than checking if one is implementable.

**Single-Parameter Mechanism Design.** This two-step approach is vividly illustrated by the important special case of *single-parameter problems*, where goal (G1) has been completely resolved. A mechanism design problem is *single-parameter* if all outcomes are real  $n$ -vectors and participants’ private objective functions have the form  $t_i(\omega) = v_i\omega_i$  for a private real number  $v_i$  (the “single parameter”);  $\omega_i$  and  $v_i$  can be thought of as the quantity received and the value-per-unit of a good, respectively. (A single-item auction is the special case in which each  $\omega$  is a standard basis vector.) An algorithm for a single-parameter problem is *monotone* if a greater value begets a greater allocation: increasing the value of a  $v_i$  (keeping the other  $v_j$ ’s fixed) can only increase the corresponding value of the computed  $\omega_i$ . For example, the “highest bidder” allocation algorithm for a single-good auction is monotone, while the “second-highest bidder” allocation algorithm is not. More generally, monotonicity characterizes implementability for single-parameter problems.

**Proposition 1 (E.g. [113])** *An algorithm for a single-parameter mechanism design problem is implementable if and only if it is monotone.*

Proposition 1 should be viewed as a useful solution to the first goal (G1), and it reduces implementable algorithm design to monotone algorithm design. An analogous characterization applies to randomized algorithms, where the monotonicity and truthfulness conditions concern expected allocations and expected participant utilities, respectively [9].

Archer and Tardos [9] were the first to systematically study approximation in single-parameter mechanism design problems. Among other contributions, they identified a natural candidate problem for a conditional separation between implementable and non-implementable polynomial-time approximation algorithms: minimizing the makespan of parallel related machines with private machine speeds. (In a scheduling context, each player is a machine with a private speed  $s_i = -1/v_i$ , allocations describe the sum of job processing times assigned to each machine, and monotonicity

dictates that declaring a slower speed can only decrease the amount of work received.) The problem admits an (exponential-time) implementable optimal algorithm, but all classical polynomial-time approximation algorithms for it (e.g., the PTASes in [55, 79]) are not monotone and hence not implementable. Beginning with [9], numerous constant-factor monotone approximation algorithms were devised (see [92] for references). Very recently, Dhangwatnotai et al. [48] proved that, allowing randomization, monotone polynomial-time algorithms are competitive with arbitrary polynomial-time algorithms for makespan minimization.

**Theorem 2 ([48])** *There is a monotone randomized PTAS, and a corresponding truthful in expectation mechanism, for makespan minimization on parallel related machines.*

Whether or not there is a conditional separation between implementable and arbitrary polynomial-time algorithms remains open. In light of Theorem 2, the most likely candidate problems for obtaining such a separation are multi-parameter; we discuss these next.

**Multi-Parameter Mechanism Design.** Many important mechanism design problems are not single-parameter. *Combinatorial auctions*, in which each participant aims to acquire a heterogeneous set of goods and has unrelated values for different sets, are a practical and basic example. (See [27, 46] for much more on the topic.) Multi-parameter mechanism design is complex and our current understanding of goals (G1) and (G2) is fairly primitive for most problems of interest. Because of its importance and bounty of open questions, the subject has been a hotbed of activity over the past few years; we briefly indicate the primary research threads next.

New characterizations of implementable algorithms are useful (and possibly essential) for understanding their approximation capabilities, and are interesting in their own right. Rochet’s Theorem [133] is a classical characterization of implementable algorithms in terms of a certain shortest-path condition known as *cycle monotonicity* (see [159]) that is general but difficult to use to prove upper or lower approximation bounds (see [97] for an exception). Archer and Kleinberg [7] give a promising reformulation of Rochet’s Theorem that could lend itself to new approximation bounds. Saks and Yu [144] show that in the common special case where the  $t_i$ ’s are drawn from convex sets, implementability is equivalent to a simpler 2-cycle condition known as *weak monotonicity*; see also [7, 111] for new extensions and alternative proofs, and [115] for a recent analog in discrete domains.

But what kinds of algorithms meet these technical conditions? The answer depends on the “richness” of the domain in which the private information (the  $t_i$ ’s) lie — richer domains possess more potentially profitable false declarations, making the space of implementable algorithms more highly constrained. For the extreme case of “unrestricted domains”, where  $\Omega$  is an abstract outcome set and the  $t_i$ ’s are arbitrary real-valued functions on  $\Omega$ , *Robert’s Theorem* [132] states that there are almost no implementable algorithms: only the VCG-like “affine maximizers”, all minor variants on the algorithm that always chooses the outcome maximizing  $\sum_i t_i(\omega)$ , possibly over an a priori restricted subset of  $\Omega$ . (Restricting to a subset of outcomes can be useful for obtaining polynomial running time.) This should be viewed as a negative result, since these affine maximizers have limited polynomial-time approximation capabilities in most important problems (see e.g. [50]). However, applications usually involve more structured domains. This point motivates an important research agenda, still in its embryonic stages, to identify the types of domains for which Robert’s Theorem holds (see [127] for a surprising new example) and characterize the additional implementable algorithms for domains in which Robert’s Theorem breaks down (see [22, 95] and [38, 53] for partial

but highly non-trivial results on combinatorial auctions and machine scheduling, respectively).

The design and analysis of good truthful multi-parameter mechanisms has proceeded apace despite our limited understanding of implementability. Much of this research has coalesced around welfare maximization in combinatorial auctions (see [27]), where  $\Omega$  is the ordered partitions  $(S_1, \dots, S_n)$  of a set of  $m$  goods among the  $n$  players, the private information  $t_i$  describes player  $i$ 's valuation (willingness to pay)  $v_i(S)$  for each of the  $2^m$  possible subsets  $S$  of goods, and the optimization problem is to choose an allocation maximizing  $\sum_i v_i(S_i)$ .<sup>2</sup> While the aforementioned VCG mechanism truthfully solves this optimization problem in exponential time, its polynomial-time approximability varies with the degree of structure imposed on valuations. General valuations exhibit both “complements”, where goods are useful only when purchased in tandem (as with a pair of tennis shoes), and “substitutes”, where goods are redundant (as with a pair of tennis rackets). Early research focused on 1- and 2-parameter valuations with complements but no substitutes and obtained best-possible approximate guarantees (assuming  $P \neq NP$ ) [99, 114]. Some of these have been extended to general valuations [26, 52, 96]. Unfortunately, with complements, the underlying welfare maximization problem includes the Maximum Independent Set problem as a special case and reasonable approximation guarantees are possible only under strong additional assumptions (as in [8, 19]).

Recent work has focused on classes of valuations with substitutes but no complements, including subadditive valuations (satisfying  $v(S \cup T) \leq v(S) + v(T)$  for all  $S, T$ ) and submodular valuations (satisfying the stronger condition that  $v(S \cup \{j\}) - v(S) \leq v(T \cup \{j\}) - v(T)$  for all  $T \subseteq S$  and  $j \notin S$ ). Here, constant-factor approximation guarantees appear possible, though challenging to obtain. Beginning with [51, 98, 121], a number of papers gave increasingly good upper and lower bounds for polynomial-time approximation of welfare maximization with complement-free valuations by *non-implementable* algorithms. See [61] and [162] for two recent gems on the algorithmic side and [109] for the latest developments in (information-theoretic) lower bounds. Remarkably, no constant-factor *implementable* algorithm is known for any such problem (see [49, 51, 52] for the state-of-the-art). Separating the power of implementable and non-implementable algorithms for welfare maximization in combinatorial auctions remains a premier research challenge. See [127] for a very recent communication complexity-based separation for a more general welfare-maximization problem, a significant research breakthrough.

**Further Aspects of AMD.** This section focused on the design of computationally efficient truthful mechanisms with provable approximation guarantees for three reasons: it comprises a large portion of AMD research; there remain numerous deep open questions on the topic; and appreciating its motivating questions and key results requires minimal economics background. We emphasize that AMD has several other thriving aspects, including: revenue-maximization with worst-case guarantees, and related algorithmic pricing problems (surveyed in [76]); revenue guarantees and cost-sharing mechanism design, with strong connections to the primal-dual method (see [82, 83, 106]); online mechanism design, in which participants arrive and depart over time (surveyed in [129]); and new models and goals for Internet-suitable mechanism design, such as distributed mechanisms (see [62, 63]), mechanisms restricted to use little or no payments [77, 101, 107, 148], and novel notions of incentive-compatibility (e.g. [101]).

---

<sup>2</sup>Valuations are usually modeled as a “black box” that supports various queries, or via a compact representation of size polynomial in  $m$ . An “efficient algorithm” in this context has running time polynomial in both  $n$  and  $m$ .

### 3 Quantifying Inefficiency and the Price of Anarchy

The truthful mechanisms studied in Section 2 are strategically degenerate in that the best course of action of a player (i.e., truth-telling) does not depend on the actions taken by the others. This was possible because a designer was assumed to have tremendous control over the game being played. Strategic games that occur “in the wild” are rarely so well behaved. Even in a design context, when the designer cannot directly dictate the allocation of resources (such as traffic rates or routing paths in a large network), dependencies between different players’ optimal courses of action are generally unavoidable, and these dependencies usually preclude exact optimization of standard objective functions. This harsh reality motivates adopting an *equilibrium concept* — a rigorous proposal for the expected outcome(s) of a game with self-interested participants — and an *approximation measure* that quantifies the inefficiency of a game’s equilibria, to address the following basic question:

(Q2) *when, and in what senses, are game-theoretic equilibria guaranteed to approximately optimize natural objective functions?*

Such a guarantee implies that the benefit of imposing additional control over the system is small, and is particularly reassuring when implementing an optimal solution is infeasible (as in a typical Internet application).

We only address question (Q2) for the most popular modeling choices (the price of anarchy of Nash equilibria) and the most well-studied application area (routing games). The end of the section provides pointers to some of the many other results in the area.

**Routing with Congestion.** General tight bounds on the inefficiency of equilibria were first proved in a model of “selfish routing” [141]. The model is originally from [20, 163] and is thoroughly discussed in [136]; the price of anarchy was originally suggested in [91] for a scheduling model, results on which are surveyed in [158].

Consider a directed multicommodity flow network — a directed graph with fixed flow rates between given source-sink vertex pairs — in which selfish users choose paths to minimize individual cost. Edge costs are *congestion-dependent*, with  $c_e(f_e)$  denoting the per-unit cost incurred by flow on edge  $e$  when there are  $f_e$  units of such flow. In an *equilibrium*, each selfish user with source  $s_i$  and sink  $t_i$  chooses an  $s_i$ - $t_i$  path  $P$  to minimize its cost  $c_P(f) = \sum_{e \in P} c_e(f_e)$ , given the routing selections of the other users. Such games are strategically non-trivial in that the routing decision of one user can alter the optimal path for another.

To keep things simple, assume that each selfish user controls a negligible fraction of the overall traffic, and that all edge cost functions are continuous and non-decreasing. Equilibrium flows are then, by definition, those on which all flow is routed on shortest paths, given the congestion. That is, a path carries flow only if it minimizes  $c_P(f)$  over all paths  $P$  with the same source and sink — if a path violated this condition, some of its selfish users would switch to a cheaper one. In every network, there is at least one equilibrium flow, and all such flows have equal total cost [20].

For example, in a “Pigou-like network” (named after [130]),  $r$  units of selfish users decide between parallel edges  $e_1$  and  $e_2$  connecting a source  $s$  to a sink  $t$ . Suppose the second edge has some cost function  $c_2(\cdot)$ , and the first edge has a constant cost function  $c_1$  everywhere equal to  $c_2(r)$ . Such networks are strategically trivial, just like the simple mechanisms of Section 2: the second edge’s cost is never larger than that of the first, even when it is fully congested. For this

reason, routing all flow on the second edge is an equilibrium. This equilibrium is generally not a minimum-cost flow. For example, if  $r = 1$  and  $c_2(x) = x$ , the equilibrium flow has cost 1, while splitting the traffic equally between the two edges yields an (optimal) flow with cost  $3/4$ . The latter flow is not an equilibrium because of a “congestion externality”: a selfish network user routed on the first edge would switch to the second edge, indifferent to the fact that this switch (slightly) increases the cost incurred by a large portion of the population.

The *price of anarchy (POA)* of a selfish routing network is the ratio of the costs of an equilibrium and an optimal flow —  $4/3$  in the example above. The closer the POA is to 1, the lesser the consequences of selfish behavior. Simple exploration of Pigou-like networks suggests that the POA is governed by the “degree of nonlinearity” of the cost function  $c_2$ ; in particular, the POA can be arbitrarily large in Pigou-like networks with unrestricted cost functions. A key result formalizes and extends this intuition to *arbitrary* multicommodity networks: among all multicommodity networks with cost functions lying in a set  $\mathcal{C}$  (e.g., bounded-degree polynomials with nonnegative coefficients), the largest-possible POA is already achieved in Pigou-like networks [135]. Conceptually, *complex topologies do not amplify the worst-case POA*. Technically, this reduction permits the easy calculation of tight bounds on the worst-case POA for most interesting sets  $\mathcal{C}$  of cost functions. For example, the POA of every multicommodity selfish routing network with affine cost functions (of the form  $c_e(f_e) = a_e f_e + b_e$  for  $a_e, b_e \geq 0$ ) is at most  $4/3$ , with a matching lower bound provided by the simple example above. See [138, 139] for recent surveys detailing these and related results.

While there is no explicit design aspect to these POA bounds, they justify nicely a common rule of thumb used in real-life network design and management: *overprovisioning networks with extra capacity ensures good performance*. This postulate was first formalized mathematically and proved in [141]. Here we provide a conceptually similar but technically different result, which is a special case of the POA bounds in [135] (see also [136, §3.6]). Suppose every edge  $e$  of a network has a *capacity*  $u_e$  and a corresponding cost function  $c_e(f_e) = 1/(u_e - f_e)$ . (If  $f_e \geq u_e$ , we interpret the cost as infinite.) This is the standard M/M/1 queueing delay function with service rate  $u_e$ , a common model in the networking literature (e.g. [21]). We say the network is  $\beta$ -*overprovisioned* for  $\beta \in (0, 1)$  if, at equilibrium, at least a  $\beta$  fraction of each edge’s capacity remains unused. The following tight bound on the POA holds for such networks.

**Theorem 3 (Consequence of [135])** *The POA of a  $\beta$ -overprovisioned network is at most*

$$\frac{1}{2} \left( 1 + \frac{1}{\sqrt{\beta}} \right).$$

Thus even 10% extra capacity reduces the worst-case price of anarchy of selfish routing to roughly 2.

**Designing for Good Equilibria.** In the same spirit as mechanism design and our prescriptive interpretation of Theorem 3, inefficiency measures such as the POA can inform how to *design* systems with good equilibria. Two variants of this idea have been explored in a number of different models: improving the POA of a given game (see [139] for a survey of selfish routing examples), and designing a *family* of games to minimize the worst-case POA. We focus on the latter idea, first proposed in [37], where a number open issues remain. See [84, 143] for surveys of other work on this important topic.

We focus on the network cost-allocation model of [32], which was motivated by the network formation games of [6] (see [137, 154] for relevant surveys). As in a selfish routing network, each

player selects a path in a multicommodity network to minimize its incurred cost. We now assume that each player controls a single (non-negligible) unit of flow and uses a single path to route it. The other key difference between the two models is the cost structure. If  $f_e$  units of flow use an edge  $e$  of a selfish routing network, this creates total cost  $f_e \cdot c_e(f_e)$  which is distributed evenly among the edges' users, for a per-unit cost of  $c_e(f_e)$ . In a network cost-allocation game, each edge  $e$  has a fixed price  $p_e$  for being used by one or more players — the cost of installing infrastructure or leasing a large fixed amount of bandwidth, say — that must be somehow distributed among the edges' users. The average per-player cost of an edge is thus *decreasing* with the number of users, giving players an incentive to cooperate via shared paths. The cost of an outcome (a choice of path per player) is the sum of the prices of the used edges, with the optimal outcome corresponding to the minimum-cost way of connecting all of the players' source-sink pairs, a Steiner connectivity problem. An obvious question is: *how should we allocate costs to players to minimize the worst-case equilibrium efficiency loss over all networks?* This cost-allocation design decision does not affect the underlying optimization problem, but it fundamentally determines the players' incentives, and hence the Nash equilibria, in the resulting path selection game.

For example, *Shapley cost-sharing* dictates sharing each edge cost equally among its users. So if  $k$  players choose paths  $P_1, \dots, P_k$ , the cost incurred by the  $i$ th player is  $\sum_{e \in P_i} (p_e / f_e)$ , where  $f_e$  is the number of players choosing a path including  $e$ . At a (pure-strategy) *Nash equilibrium*, no player can switch paths to strictly decrease its cost, given the paths selected by the other players. Shapley cost-sharing leads to at least one equilibrium in every induced network cost-allocation game [6], and generally to multiple equilibria. For example, in a network of parallel links, all with costs strictly between 1 and  $k$ , every link corresponds to a different Nash equilibrium: if all players use a link with price  $p$ , each player pays only  $p/k < 1$ , and a unilateral deviation to a different link would cost more than this. The POA is traditionally defined by the worst equilibrium [91], and this example yields a linear lower bound for the worst-case POA of Shapley cost-sharing (and there is an easy matching upper bound). Can we do better?

The answer is different for undirected and directed networks. An alternative to Shapley cost-sharing is *ordered cost-sharing*, a simple priority scheme: order the players arbitrarily, with the first user of an edge (according to this order) paying its full cost. Up to tie-breaking, there is a unique Nash equilibrium under ordered cost-sharing: the first player chooses a shortest path between its source and sink, the second player chooses a shortest path given the edges already paid for by the first player, and so on. Indeed, the equilibria induced by ordered cost-sharing are in one-to-one correspondence with the possible outputs of well-studied greedy online algorithms for Steiner connectivity problems [13, 80]. This correspondence implies that, in undirected networks, *ordered cost-sharing has exponentially better worst-case POA than Shapley cost-sharing*. There is also a matching lower bound.

**Theorem 4 ([32])** *In undirected cost-allocation games, ordered cost-sharing minimizes the worst-case POA (up to constant factors).*

The proof of Theorem 4 is highly non-trivial, and hinges on a complete classification of the cost-sharing methods that are guaranteed to induce at least one Nash equilibrium in every network. These turn out to be precisely the “concatenations” of weighted Shapley values (in the sense of [87]); Shapley cost-sharing is the special case of uniform weights and no concatenation, while ordered cost-sharing is the  $k$ -fold concatenation of the trivial one-player Shapley value. As proved in [32], no method of this type can outperform ordered cost-sharing by more than a constant factor.

In directed networks, it is easy to show that all cost-sharing methods, including ordered ones, have linear worst-case POA and thus do not improve upon Shapley cost-sharing. We can refine this comparison by analyzing the ratio of the cost of a best (instead of the worst) Nash equilibrium and that of an optimal solution, a quantity known as the price of stability (POS). The worst-case POS of Shapley cost-sharing in directed networks is precisely the  $k$ th Harmonic number  $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k} \approx \ln k$  [6]. A consequence of the classification in [32] discussed above is that no other cost-sharing method has superior worst-case POS (or POA).

**Theorem 5 ([32])** *In directed cost-allocation games, Shapley cost-sharing minimizes the worst-case POS and POA.*

**Further Aspects of Quantifying Inefficiency.** We have barely scratched the surface of recent work on equilibrium efficiency analyses. See [90, 138] for a partial overview of other models of routing games — following [134, 142], much of this work is couched in the more abstract language of “congestion games”. See [18, 73, 84, 111, 123, 139, 154, 158] for efficiency analyses in some other application domains. Currently active research topics that apply approximation measures to game-theoretic concepts include: efficiency analyses of equilibrium concepts other than best-case and worst-case Nash equilibria [5, 11, 30, 31, 36, 39, 54]; efficiency guarantees in models that allow altruistic and/or malicious participants, rather than only self-interested ones [16, 33, 88]; and novel uses of worst-case approximation to quantify different phenomena in games [15, 60, 66, 78, 149].

As already noted, one important emerging trend in the area is problems of designing games with efficient equilibria. A second is to prove POA-type bounds under increasingly weak assumptions on the rationality of participants. Recall that in Section 2, our only assumption was that participants will make use of a “foolproof” strategy (one that dominates all others), should one be available. This section implicitly assumed that selfish participants can reach a Nash equilibrium of a game without such foolproof strategies, presumably through repeated experimentation. This much stronger assumption has been addressed in two different ways in the recent literature. The first is to formally justify it by positing natural experimentation strategies and proving that they quickly reach a (possibly approximate) equilibrium; see [14, 23, 35, 57, 65] for a sampling of such results. The second is to prove POA-like guarantees on system performance that apply even when such experimentation strategies fail to converge to an equilibrium. Remarkably, such bounds are possible in interesting classes of games, including the selfish routing networks discussed in this section. See [70, 110] and [24] for two different formalizations of this approach.

## 4 Complexity of Equilibrium Computation

Equilibrium concepts such as the Nash equilibrium obviously play a starring role in game theory and microeconomics. If nothing else, a notion of equilibrium describes outcomes that, once reached, persist under some model of individual behavior. In engineering applications we generally demand a stronger interpretation of an equilibrium, as a credible *prediction* of the long-run state of the system. But none of the standard equilibrium notions or the corresponding proofs of existence suggest how to arrive at an equilibrium with a reasonable amount of effort. The Pavlovian response of any theoretical computer scientist would be to pose the following queries.

(Q3) *When can the participants of a game quickly converge to an equilibrium? More modestly, when can a centralized algorithm quickly compute an equilibrium?*

These questions are important for two reasons. Algorithms for equilibrium computation can be useful practically, for example in game-playing (e.g. [69]) and for multi-agent reasoning (see [151] for an introduction). Second, resolving the complexity of computing an equilibrium concept has economic implications: a polynomial-time algorithm for computing an equilibrium is a crucial step toward establishing its credibility, while an intractability result casts doubt on its predictive power (a type of critique dating back at least 50 years [131]).

There has been a frenzy of recent work on these questions, for many different fundamental equilibrium concepts. Perhaps the most celebrated results in the area concern the *PPAD*-completeness of computing mixed-strategy Nash equilibria in finite games with two or more players [34, 47]. To briefly convey the spirit of the area with a minimum of technical fuss, we instead discuss the complexity of converging to and computing pure-strategy Nash equilibria in variants of the routing games studied in Section 3. The end of the section mentions the key differences between the two settings, and also provides references on results on other central equilibrium computation problems (such as computing market and correlated equilibria).

**Pure Equilibria in Network Congestion Games.** Recall the selfish routing networks of Section 3. The *atomic* variant is similar to the cost allocation games of that section, in that each of  $k$  players controls a non-negligible fraction of the overall traffic (say one unit each) and routes it on a single path. Each edge cost function  $c_e : \{1, 2, \dots, k\} \rightarrow \mathcal{R}^+$ , describing the per-player cost along an edge as a function of its number of users, is non-decreasing. Similarly to cost allocation games, an outcome  $(P_1, \dots, P_k)$  is a (pure-strategy) Nash equilibrium (PNE) if each player simultaneously chooses a *best response*: a path with minimum-possible cost  $\sum_e c_e(f_e)$ , given the paths chosen by the other players.

*Best-response dynamics (BRD)* is a simple model of experimentation by players over time: while the current outcome is not a PNE, choose an arbitrary player that is not using a best response, and update its path to a best response. The update of one player usually changes the best responses of the others; for this reason, BRD fails to converge in many games. In an atomic selfish routing network, however, every iteration of BRD strictly decreases the potential function  $\Phi(P_1, \dots, P_k) = \sum_{e \in E} \sum_{i=1}^{f_e} c_e(i)$ , and thus BRD is guaranteed to terminate, necessarily at a PNE [112, 134]. Does convergence require polynomial or exponential time? Can we compute a PNE of such a game by other means in polynomial time?

**TFNP and PLS.** The problem of computing a PNE of an atomic selfish routing game is a member of *TFNP* (“total functional *NP*”), the intriguing class of search problems for which all instances have a short and efficiently verifiable witness [105]. Intuitively, all well-formed instances have a solution (in our case, a PNE); the only issue is finding one in polynomial time.

Assume for the moment that the problem lies outside *P*; how would we amass evidence for this fact? We can’t expect to prove that a *TFNP* problem is *NP*-hard in a meaningful sense; a short argument shows that such a reduction would imply  $NP = coNP$  [105]. We also can’t expect to show that it is *TFNP*-complete, since *TFNP* is a “semantic class” — informally, there is no apparent way to efficiently check membership in *TFNP*, given (say) a Turing machine description of an *NP* search problem — and thus unlikely to contain complete problems (see [85, 152]). Our best option is therefore to define a “syntactic” subclass of *TFNP* that contains as many problems as possible (including computing PNE), while admitting complete problems.

We follow [140] in motivating the appropriate subclass. View the definition of *NP* (existence

of short witnesses and an efficient verifier) as the minimal constraint ensuring that a problem is solvable by brute-force search, enumerating all possible witnesses, using polynomial time per iteration. Computing a PNE of an atomic selfish routing game appears to be easier than  $NP$ -hard problems because there is a *guided search* algorithm (namely BRD) that is guaranteed to find a legitimate witness. What are the minimal ingredients that guarantee that a problem admits an analogous guided search procedure? This question was answered twenty years ago in the context of local search algorithms, by the definition of the class  $PLS$  (for “polynomial local search”) [86]. A  $PLS$  problem is described by three polynomial-time algorithms: one to accept an instance and output an initial candidate solution; one to evaluate the objective function value of a candidate solution; and one that either verifies local optimality (for some local neighborhood) or else returns a neighboring solution with strictly better objective function value. To “solve” a  $PLS$  problem means to compute a local optimum. For example, computing a PNE of an atomic selfish routing game can be cast as a  $PLS$  problem by adopting the potential function as an objective, and defining two outcomes to be neighbors if all but one player choose the same path in both. Local minima then correspond to the PNE of the game.

Every  $PLS$  problem can be solved in exponential time via local search, although this is not always the fastest method. For example, in *single-commodity* atomic selfish routing games, where all players have the same source and sink, a PNE can be computed in polynomial time using minimum-cost flow [59] despite the fact that BRD (i.e., local search) can require an exponential number of iterations [1].

If  $P = PLS$ , then given only an abstract description of a  $PLS$  problem in terms of the three algorithms above, there is a generic, problem-independent way of finding a “shortcut” to a locally optimal solution, exponentially faster than rote traversal of the path suggested by the guided search algorithm. For both this conceptual reason and its inclusion of many well-known and apparently difficult problems, it is generally believed that  $P \neq PLS$ .  $PLS$ -hardness should therefore be viewed as strong evidence that a  $TfNP$  search problem is not solvable in polynomial time.

$PLS$  can be phrased as a syntactic class and it therefore admits a generic complete problem [86]. The analog of Cook’s Theorem — a reduction from the generic complete problem to a concrete one — states that a particular local search problem for Boolean circuits called “Circuit Flip” is  $PLS$ -complete [86]. Circuit Flip has been used to establish the  $PLS$ -completeness of many other problems (e.g. [147, 164]). Computing a PNE of a (multicommodity) atomic selfish routing network is also hard in this sense.

**Theorem 6 ([59])** *The problem of computing a PNE of an atomic selfish routing game is  $PLS$ -complete.*

See also [1] for an alternative proof, and [1, 2, 59, 153] for further  $PLS$ -completeness results for computing PNE.

The reductions in  $PLS$ -completeness results such as Theorem 6 nearly always give unconditional exponential lower bounds on the worst-case running time of the generic local search algorithm (BRD, in the present context). Even if  $P = PLS$ , the following corollary holds.

**Corollary 7 ([59])** *There is a constant  $c > 0$  such that for arbitrarily large  $n$ , there is an  $n$ -player atomic selfish routing network and an initial outcome from which BRD requires  $2^{cn}$  iterations to converge to a PNE, no matter how players are selected in each step of BRD.*

**Mixed-Strategy Nash Equilibria and *PPAD*.** A *mixed strategy* is a probability distribution over the pure strategies of a player. In a *mixed-strategy Nash equilibrium (MNE)*, every player simultaneously chooses a mixed strategy maximizing its expected utility, given those chosen by the others. Mixed strategies are necessary to establish the existence of Nash equilibria in arbitrary finite games [116], but they are not without conceptual controversy (see e.g. [122, §3.2]). Regardless, computing an MNE of a finite game is clearly a central equilibrium computation problem.

We focus on the two-player (“bimatrix”) case, where the input is two  $m \times n$  payoff matrices (one for each player) with integer entries.<sup>3</sup> There is an exponential-time algorithm for computing an MNE in bimatrix games, which enumerates over pairs of supports for the two players and solves a linear system for each to check for a feasible solution (see e.g. [125, 140, 151]). There is a less obvious (but still exponential-time [146]) “guided search” algorithm called the *Lemke-Howson (LH) algorithm* [100]; see [160] for a careful exposition. The LH algorithm is a path-following algorithm in the spirit of local search, but is not guided by an objective or potential function and thus does not obviously prove that computing a MNE of a bimatrix game is in *PLS*.

A related but apparently different subclass of *TFNP*, called *PPAD* (for “polynomial parity argument, directed version”), was defined in [124] to capture the complexity of this and related problems (mostly from combinatorial topology, such as computing approximate Brouwer fixed points). Its formal definition parallels that of *PLS*, with a *PPAD* problem consisting of the minimal ingredients necessary to execute a LH-like search procedure (again easily phrased as three polynomial-time algorithms). *PPAD*-hardness is viewed as a comparable negative result to *PLS*-hardness (for the same reasons). Computing an MNE of a bimatrix game is hard in this sense.

**Theorem 8 ([34, 47])** *The problem of computing an MNE of a bimatrix game is *PPAD*-complete.*

This hardness result extends to computing a natural notion of an “ $\epsilon$ -approximate MNE” for values of  $\epsilon$  as large as inverse polynomial [34], thus ruling out an FPTAS for approximating an MNE (unless  $P = PPAD$ ). Unlike *PLS*-completeness results, *PPAD*-completeness results are not known to have immediate unconditional consequences in the spirit of Corollary 7. Nevertheless, an exponential lower bound on the convergence time of certain dynamics to an MNE was recently proved in [75]. See also [125, 140] for pointers on positive and negative results about computing MNE in structured and random finite games.

The proof of Theorem 8 is necessarily intricate because it is a “Cook’s Theorem for *PPAD*” — all previously known *PPAD*-complete problems [124] have the flavor of “generic” complete problems. For example, instances of *PPAD*-complete fixed-point problems include an encoding of a polynomial-time algorithm that computes the values of some continuous function restricted to a subdivided simplex. Any reduction from a known *PPAD*-complete problem to computing MNE must therefore explicitly encode arbitrary polynomial-time computation in terms of a bimatrix game. Many of the first “concrete” *PLS*-complete problems required similarly intricate reductions (e.g. [147]). See [125] for a nice high-level survey of the proof of Theorem 8 and the sequence of results that led to it.

**Further Aspects of Equilibrium Computation.** Another genre of equilibrium computation problems bustling with activity is *market* or *price equilibria* — prices for goods at which decentralized and selfish exchange “clears the market”, yielding a Pareto efficient allocation of the goods.

---

<sup>3</sup>With three or more players, the problem appears to be harder [56]. Enumerating all and optimizing over the MNE of a bimatrix game are also harder problems [44, 68].

As with mixed Nash equilibria, market equilibria exist under weak conditions [10] but their efficient computation is largely open. The last five years have seen a number of new polynomial-time algorithms (surveyed in [156] and [41]) and a few scattered hardness results (see [41]), but many basic questions remain unresolved (see [156]).

Back in finite games, equilibrium computation in extensive-form games — specified by a game tree in which paths represent sequences of actions by the various players and by nature, see e.g. [161] — was studied early on by the AI community (surveyed in [89]) and more recently in the theoretical computer science literature (e.g. [108]). Equilibrium computation in the stochastic variants defined in [43] are, along with some number-theoretic problems like factoring, among the most prominent candidates for problems in  $(NP \cap coNP) \setminus P$  (see [85]).

Other equilibrium concepts in finite games have also been studied recently. For correlated equilibria [12], an equilibrium concept with fundamental connections to no-regret learning algorithms (see [25]), sweeping positive algorithmic results are possible [126]. In repeated games, computing a Nash equilibrium is polynomial-time solvable in two-player games [104] but *PPAD*-hard with three or more players [28], despite the overwhelming number of equilibria guaranteed by the “folk theorem” for such games.

## 5 Future Directions

The astonishing and accelerating rate of progress in algorithmic game theory, nourished by deep connections with other areas of theoretical computer science and a consistent infusion of new motivating applications, leaves no doubt that it will continue to flourish for many years to come. There is presently a surplus of challenging open questions across all three of the areas surveyed in Sections 2–4; we record a small handful to prove the point.

We first mention some concrete problems that are well known in the AGT community. A few in AMD include: prove better upper or lower bounds on the achievable approximation guarantees of polynomial-time implementable algorithms for combinatorial auctions (see [27] for a reasonably current survey); characterize the multi-parameter domains for which affine maximizers are the only implementable algorithms (see [127] for the latest developments); and develop some understanding of the power of randomization in polynomial-time implementability (see [3] for an entry point). Some personal favorites involving equilibrium efficiency analyses are: determine the POA in atomic selfish routing networks with fractional routing and the POS in Shapley cost allocation games (see [42] and [64], respectively, for partial results); develop a general analytical technique to extract tight efficiency loss bounds from potential functions and/or variational inequalities (see [137]); and, in the spirit of [32], identify how to distribute delays (via an appropriate queuing policy) to minimize the worst-case POA in selfish routing networks. Central open questions in equilibrium computation include the complexity of computing approximate mixed-strategy Nash equilibria (see [29, 103, 155] for the state-of-the-art), the complexity of computing market equilibria with reasonably general (concave) participant utility functions (see [156]), and the complexity of computing an equilibrium in the stochastic games in  $NP \cap coNP$  defined in [43] (see also [85]).

Speaking more informally and long-term, we expect that all areas of AGT will (and should) grapple with appropriate models of agent behavior over the next several years. Some type of non-worst-case behavioral assumptions are inevitable for systems with independent participants: all of the results described in this survey, even the welfare guarantee of the simple Vickrey auction, depend on such assumptions. AGT has minimized controversy thus far by adopting well-known

notions from traditional game theory, such as the Nash equilibrium. But if traditional game theory applied “off the shelf” to contemporary computer science, there would be no need for AGT at all. See [67] for a compelling argument — made over a decade ago but more appropriate than ever — about why computer scientists need to completely rethink standard models of rationality and equilibrium concepts.

Behavioral assumptions are essential to address modern applications, yet are largely foreign to the mainstream “STOC/FOCS” mentality and its emphasis on minimal assumptions and worst-case analysis. Can we retain this unquestionably useful and well-motivated bias while expanding our field’s reach? *Of course*: shining examples of worst-case guarantees coupled with novel behavioral models have already begun to sprout in the AGT literature. For example: mechanism implementation in undominated strategies [17] and in ex post collusion-proof Nash equilibrium [120]; the price of total anarchy [24]; and the complexity of unit-recall games [58]. We expect that these are only the vanguard of what promises to be a rich and relevant theory.

**Acknowledgments.** We thank Moshe Babaioff, Liad Blumrosen, Peerapong Dhangwatnotai, Shaddin Dughmi, Michael Schapira, and Mukund Sundararajan for comments on a previous draft.

## References

- [1] H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. In *FOCS '06*, pages 613–622.
- [2] H. Ackermann and A. Skopalik. On the complexity of pure Nash equilibria in player-specific network congestion games. In *WINE '07*, pages 419–430.
- [3] G. Aggarwal, A. Fiat, A. V. Goldberg, J. D. Hartline, N. Immorlica, and M. Sudan. Derandomization of auctions. In *STOC '05*, pages 619–625.
- [4] E. Altman, T. Boulogne, R. El Azouzi, T. Jiménez, and L. Wynter. A survey on networking games in telecommunications. *Computers & Operations Research*, 33(2):286–311, 2006.
- [5] N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. In *SODA '07*, pages 189–198.
- [6] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS '04*, pages 295–304.
- [7] A. Archer and R. D. Kleinberg. Truthful germs are contagious: a local-to-global characterization of truthfulness. In *EC '08*.
- [8] A. Archer, C. H. Papadimitriou, K. Talwar, and É. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *SODA '03*, pages 205–214.
- [9] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *FOCS '01*, pages 482–491.
- [10] K. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22(3):265–290, 1954.
- [11] I. Ashlagi, D. Monderer, and M. Tennenholtz. On the value of correlation. In *UAI '05*, pages 34–41.
- [12] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.
- [13] B. Awerbuch, Y. Azar, and Y. Bartal. On-line generalized Steiner problem. In *SODA '96*, pages 68–74.
- [14] B. Awerbuch, Y. Azar, A. Epstein, V. S. Mirrokni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. In *EC '08*.

- [15] M. Babaioff, M. Feldman, and N. Nisan. Combinatorial agency. In *EC '06*, pages 18–28.
- [16] M. Babaioff, R. D. Kleinberg, and C. H. Papadimitriou. Congestion games with malicious players. In *EC '07*, pages 103–112.
- [17] M. Babaioff, R. Lavi, and E. Pavlov. Single-value combinatorial auctions and implementation in undominated strategies. In *SODA '06*, pages 1054–1063.
- [18] J. Bae, E. Beigman, R. Berry, M. Honig, and R. Vohra. Efficiency bounds for sequential resource allocation auctions. In *CDC '07*, pages 765–770.
- [19] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK '03*, pages 72–87.
- [20] M. J. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
- [21] D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice-Hall, 1987. Second Edition, 1991.
- [22] S. Bikhchandani, S. Chatterji, R. Lavi, A. Mu'alem, N. Nisan, and A. Sen. Weak monotonicity characterizes dominant strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.
- [23] A. Blum, E. Even-Dar, and K. Ligett. Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games. In *PODC '06*, pages 45–52.
- [24] A. Blum, M. Hajiaghayi, K. Ligett, and A. Roth. Regret minimization and the price of total anarchy. In *STOC '08*, pages 373–382.
- [25] A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. In Nisan et al. [119], chapter 4, pages 79–101.
- [26] L. Blumrosen and N. Nisan. On the computational power of iterative auctions. In *EC '05*, pages 29–43.
- [27] L. Blumrosen and N. Nisan. Combinatorial auctions. In Nisan et al. [119], chapter 11, pages 267–299.
- [28] C. Borgs, J. Chayes, N. Immorlica, A. T. Kalai, V. S. Mirrokni, and C. H. Papadimitriou. The myth of the folk theorem. In *STOC '08*, pages 365–372.
- [29] H. Bosse, J. Byrka, and E. Markakis. New algorithms for approximate Nash equilibria in bimatrix games. In *WINE '07*, pages 17–29.
- [30] M. Charikar, H. Karloff, C. Mathieu, J. Naor, and M. Saks. Online multicast with egalitarian cost sharing. In *SPAA '08*.
- [31] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative multicast and facility location games. In *EC '06*, pages 72–81.
- [32] H. Chen, T. Roughgarden, and G. Valiant. Designing networks with good equilibria. In *SODA '08*, pages 854–863.
- [33] P.-A. Chen and D. Kempe. Altruism, selfishness, and spite in traffic routing. In *EC '08*.
- [34] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of two-player Nash equilibria. *Journal of the ACM*, 2008.
- [35] S. Chien and A. Sinclair. Convergence to approximate Nash equilibria in congestion games. In *SODA '07*, pages 169–178.
- [36] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *ESA '05*, pages 59–70.
- [37] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. In *ICALP '04*, pages 345–357.

- [38] G. Christodoulou, E. Koutsoupias, and A. Vidali. A characterization of 2-player mechanisms for scheduling. Submitted, 2008.
- [39] C. Chung, K. Ligett, K. Pruhs, and A. Roth. The price of stochastic anarchy. In *SAGT '08*, pages 303–314.
- [40] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
- [41] B. Codenotti and K. Varadarajan. Computation of market equilibria by convex programming. In Nisan et al. [119], chapter 6, pages 135–158.
- [42] R. Cominetti, J. R. Correa, and N. E. Stier Moses. The impact of oligopolistic competition in networks. *Operations Research*, 2008.
- [43] A. Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.
- [44] V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. In *IJCAI '03*, pages 765–771.
- [45] P. Cramton. Spectrum auctions. In *Handbook of Telecommunications Economics*, chapter 14, pages 605–639. 2002.
- [46] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [47] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibria. *SIAM Journal on Computing*, 2008.
- [48] P. Dhangwatnotai, S. Dobzinski, S. Dughmi, and T. Roughgarden. Truthful approximation schemes for single-parameter agents. In *FOCS '08*.
- [49] S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *APPROX '07*, pages 89–103.
- [50] S. Dobzinski and N. Nisan. Limitations of VCG-based mechanisms. In *STOC '07*, pages 338–344.
- [51] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *STOC '05*, pages 610–618.
- [52] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. In *STOC '06*, pages 644–652.
- [53] S. Dobzinski and M. Sundararajan. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *EC '08*.
- [54] A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost sharing connection games. In *EC '07*, pages 84–92.
- [55] L. Epstein and J. Sgall. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica*, 39(1):43–57, 2004.
- [56] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. In *FOCS '07*, pages 113–123.
- [57] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibria. In *ICALP '03*, pages 502–513.
- [58] A. Fabrikant and C. H. Papadimitriou. The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond. In *SODA '08*, pages 844–853.
- [59] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *STOC '04*, pages 604–612.
- [60] R. Fadel and I. Segal. The communication cost of selfishness. *Journal of Economic Theory*, 2008.
- [61] U. Feige. On maximizing welfare when utility functions are subadditive. In *STOC '06*, pages 41–50.

- [62] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.
- [63] J. Feigenbaum, M. Schapira, and S. Shenker. Distributed algorithmic mechanism design. In Nisan et al. [119], chapter 14, pages 363–384.
- [64] A. Fiat, H. Kaplan, M. Levy, S. Olonetsky, and R. Shabo. On the price of stability for designing undirected networks with fair cost allocations. In *ICALP '06*, pages 608–618.
- [65] S. Fischer and B. Vöcking. On the evolution of selfish routing. In *ESA '04*, pages 323–334.
- [66] D. Fotakis, S. Kontogiannis, and P. Spirakis. Atomic congestion games among coalitions. *ACM Transactions on Algorithms*, 2008.
- [67] E. J. Friedman and S. J. Shenker. Learning and implementation on the Internet. Working paper, 1997.
- [68] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.
- [69] A. Gilpin, T. Sandholm, and T. B. Sorensen. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker. In *AAAI '07*.
- [70] M. X. Goemans, V. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *FOCS '05*, pages 142–151.
- [71] A. V. Goldberg and J. D. Hartline. Collusion-resistant mechanisms for single-parameter agents. In *SODA '05*, pages 620–629, 2005.
- [72] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
- [73] M. H. Halldórsson, J. Y. Halpern, E. L. Li, and V. S. Mirrokni. On spectrum sharing games. In *PODC '04*, pages 107–114.
- [74] J. Y. Halpern. Computer science and game theory: A brief survey. In S. N. Durlauf and L. E. Blume, editors, *Palgrave Dictionary of Economics*. 2008.
- [75] S. Hart and Y. Mansour. The communication complexity of uncoupled Nash equilibrium procedures. *Games and Economic Behavior*, 2008.
- [76] J. Hartline and A. Karlin. Profit maximization in mechanism design. In Nisan et al. [119], chapter 13, pages 331–362.
- [77] J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *STOC '08*, pages 75–84.
- [78] A. Hayrapetyan, É. Tardos, and T. Wexler. The effect of collusion in congestion games. In *STOC '06*, pages 89–98.
- [79] D. Hochbaum and D. B. Shmoys. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM J. Comput.*, 17(3):539–551, 1988.
- [80] M. Imase and B. M. Waxman. Dynamic Steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3), 1991.
- [81] M. O. Jackson. A crash course in implementation theory. *Social Choice and Welfare*, 18(4):655–708, 2001.
- [82] K. Jain and M. Mahdian. Cost sharing. In Nisan et al. [119], chapter 15, pages 385–410.
- [83] K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *STOC '01*, pages 364–372.

- [84] R. Johari. The price of anarchy and the design of scalable resource allocation mechanisms. In Nisan et al. [119], chapter 21, pages 543–568.
- [85] D. S. Johnson. The NP-completeness column: Finding needles in haystacks. *ACM Transactions on Algorithms*, 3(2), 2007. Article 24.
- [86] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.
- [87] E. Kalai and D. Samet. On weighted Shapley values. *International Journal of Game Theory*, 16(3):205–222, 1987.
- [88] G. Karakostas and A. Viglas. Equilibria for networks with malicious users. *Mathematical Programming*, 110(3):591–613, 2007.
- [89] D. Koller and A. Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1-2):167–215, 1997.
- [90] S. C. Kontogiannis and P. G. Spirakis. Atomic selfish routing in networks: A survey. In *WINE '05*, pages 989–1002.
- [91] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS '99*, pages 404–413.
- [92] A. Kovács. Tighter approximation bounds for LPT scheduling in two special cases. In *CIAC '06*, pages 187–198.
- [93] S. Lahaie, D. Pennock, A. Saberi, and R. Vohra. Sponsored search auctions. In Nisan et al. [119], chapter 28.
- [94] R. Lavi. Computationally efficient approximation mechanisms. In Nisan et al. [119], chapter 12, pages 301–329.
- [95] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *FOCS '03*, pages 574–583.
- [96] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. In *FOCS '05*, pages 595–604.
- [97] R. Lavi and C. Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *EC '07*, pages 252–261.
- [98] B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *EC '01*, pages 18–28.
- [99] D. Lehmann, L. I. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002.
- [100] C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *SIAM Journal*, 12(2):413–423, 1964.
- [101] H. Levin, M. Schapira, and A. Zohar. Interdomain routing and games. In *STOC '08*, pages 57–66.
- [102] N. Linial. Game-theoretic aspects of computing. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 2, chapter 38, pages 1339–1395. 1994.
- [103] R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *EC '03*, pages 36–41.
- [104] M. L. Littman and P. Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39(1):55–66, 2005.
- [105] N. Megiddo and C. H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

- [106] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. In *EC '07*, pages 1–10.
- [107] M. Mihail, C. H. Papadimitriou, and A. Saberi. On certain connectivity properties of the Internet topology. *Journal of Computer and System Sciences*, 72(2):239–251, 2006.
- [108] P. B. Miltersen and T. B. Sørensen. Fast algorithms for finding proper strategies in game trees. In *SODA '08*, pages 874–883.
- [109] V. S. Mirrokni, M. Schapira, and J. Vondrak. Tight information-theoretic lower bounds for combinatorial auctions. In *EC '08*.
- [110] V. S. Mirrokni and A. Vetta. Convergence issues in competitive games. In *APPROX '04*, pages 183–194.
- [111] D. Monderer. Monotonicity and implementability. In *EC '08*.
- [112] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.
- [113] D. Mookherjee and S. Reichelstein. Dominant strategy implementation of Bayesian incentive compatible allocation rules. *Journal of Economic Theory*, 56(2):378–399, 1992.
- [114] A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI '02*, pages 379–384.
- [115] A. Mu'alem and M. Schapira. Mechanism design over discrete domains. In *EC '08*.
- [116] J. F. Nash. Equilibrium points in  $N$ -person games. *Proceedings of the National Academy of Science*, 36(1):48–49, 1950.
- [117] N. Nisan. Introduction to mechanism design (for computer scientists). In Nisan et al. [119], chapter 9, pages 209–241.
- [118] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1/2):166–196, 2001.
- [119] N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [120] N. Nisan, M. Schapira, and A. Zohar. Best-reply mechanisms. Working paper, 2008.
- [121] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006.
- [122] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [123] A. Ozdaglar and R. Srikant. Incentives and pricing in communication networks. In Nisan et al. [119], chapter 22, pages 571–591.
- [124] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.
- [125] C. H. Papadimitriou. The complexity of finding Nash equilibria. In Nisan et al. [119], chapter 2, pages 29–51.
- [126] C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 2008.
- [127] C. H. Papadimitriou, M. Schapira, and Y. Singer. On the hardness of being truthful. Manuscript, 2008.
- [128] D. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, University of Pennsylvania, 2001.

- [129] D. C. Parkes. Online mechanisms. In Nisan et al. [119], chapter 16, pages 411–439.
- [130] A. C. Pigou. *The Economics of Welfare*. Macmillan, 1920.
- [131] M. O. Rabin. Effective computability of winning strategies. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory Games*, volume 3. Princeton University Press, 1957.
- [132] K. Roberts. The characterization of implementable choice rules. In J.-J. Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–349. 1979.
- [133] J. C. Rochet. A necessary and sufficient condition for rationalizability in a quasilinear context. *Journal of Mathematical Economics*, 16(2):191–200, 1987.
- [134] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.
- [135] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.
- [136] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
- [137] T. Roughgarden. Potential functions and the inefficiency of equilibria. In *Proceedings of the International Congress of Mathematicians*, volume III, pages 1071–1094, 2006.
- [138] T. Roughgarden. Routing games. In Nisan et al. [119], chapter 18, pages 461–486.
- [139] T. Roughgarden. Selfish routing and the price of anarchy. *OPTIMA*, 74:1–15, 2007.
- [140] T. Roughgarden. Computing equilibria: A computational complexity perspective. *Economic Theory*, 2008.
- [141] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.
- [142] T. Roughgarden and É. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 49(2):389–403, 2004.
- [143] T. Roughgarden and É. Tardos. Introduction to the inefficiency of equilibria. In Nisan et al. [119], chapter 17, pages 443–459.
- [144] M. E. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *EC '05*, pages 286–293.
- [145] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1):1–54, 2002.
- [146] R. Savani and B. von Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74(2):397–429, 2006.
- [147] A. A. Schäffer and M. Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.
- [148] J. Schummer and R. V. Vohra. Mechanism design without money. In Nisan et al. [119], chapter 10, pages 243–265.
- [149] S. Shakkottai, R. Srikant, A. Ozdaglar, and D. Acemoglu. The price of simplicity. *IEEE Journal on Selected Areas in Communications*, 2008.
- [150] Y. Shoham. Computer science and game theory. *Communications of the ACM*, 2008.
- [151] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2008.
- [152] M. Sipser. On relativization and the existence of complete sets. In *ICALP '82*, pages 523–531.
- [153] A. Skopalik and B. Vöcking. Inapproximability of pure Nash equilibria. In *STOC '08*, pages 355–364.

- [154] É. Tardos and T. Wexler. Network formation games and the potential function method. In Nisan et al. [119], chapter 19, pages 487–516.
- [155] H. Tsaknakis and P. G. Spirakis. An optimization approach for approximate Nash equilibria. In *WINE '07*, pages 42–56.
- [156] V. V. Vazirani. Combinatorial algorithms for market equilibria. In Nisan et al. [119], chapter 5, pages 103–134.
- [157] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.
- [158] B. Vöcking. Selfish load balancing. In Nisan et al. [119], chapter 20, pages 517–542.
- [159] R. Vohra. Paths, cycles and mechanism design. Working paper, 2007.
- [160] B. von Stengel. Computing equilibria for two-person games. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, volume 3, chapter 45, pages 1723–1759. North-Holland, 2002.
- [161] B. von Stengel. Equilibrium computation for two-player games in strategic and extensive form. In Nisan et al. [119], chapter 3, pages 53–78.
- [162] J. Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC '08*, pages 67–74.
- [163] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers, Pt. II*, volume 1, pages 325–378, 1952.
- [164] M. Yannakakis. Computational complexity. In E. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, chapter 2, pages 19–55. 1997.