Algorithmic Game Theory: Some Greatest Hits and Future Directions

Tim Roughgarden*

Department of Computer Science, Stanford University, 353 Serra Mall, Stanford, CA 94305

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 aftershocks 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, ways. 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.

Sections 2–4 touch on the current dominant research trends in AGT, loosely following the organization of the first book in the field [94]; Section 5 highlights a number of prominent open questions. We discuss only (a subset of the) topics

* 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.

studied by "the STOC/FOCS community"; see [4, 54, 79, 123] for alternative perspectives on computer science and game theory.

2 Algorithmic Mechanism Design

Algorithmic mechanism design 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 [130], 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 [93] 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 (AMD)". (See [76, 101, 119] for related contemporaneous work on combinatorial auctions in the AI literature.) Auction design is the most obvious motivation for this subfield, but there are many others. See [92] for a list of traditional economic applications, together with [71] and [37] for overviews of two modern "killer applications" — keyword search auctions and spectrum auctions, respectively. The economic literature on mechanism design is very rich (e.g., [60]), 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 Ω, n participants each with a real-valued private objective function $t_i(\cdot)$ defined on Ω, and a designer objective function $f(t_1, \ldots, t_n)$. In the Vickrey auction, Ω has one outcome per participant (indicating the winner), $t_i(\omega)$ is v_i if i wins in ω 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 Ω. The AMD analog is only harder: simultaneously *determine the (private) t_i's* and optimize the corresponding f over Ω. Sometimes the latter problem is no more difficult that 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 $x(b_1, \ldots, b_n)$ and a payment algorithm $\pi(b_1, \ldots, b_n)$ to determine an outcome ω and payments $p_1, \ldots, p_n$, respectively; and 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*.[2] The allocation algorithm of a simple mechanism is essentially solving the classical optimization version of the problem with known t_i's (assuming all players bid truthfully, as they should).

Call an allocation algorithm *implementable* if, for some cleverly chosen payment algorithm π, coupling x with π yields a (truthful) simple mechanism. For a single-good auction, if x is the "highest-bidder" allocation algorithm, then defining π 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. There is a strong positive result in the latter scenario, achieved by a far-reaching generalization of the Vickrey auction known as the "VCG mechanism" (see e.g. [92]): for every mechanism design problem with a sum objective ($\sum_i t_i(\omega)$, and weighted variants), the optimal (not necessarily polynomial-time) allocation algorithm is implementable. This is not generally the case for non-sum objectives [10, 93].

[2] The usual term is "truthful, direct-revelation". Our restriction to simple mechanisms is partially but not fully without loss of generality; see Section 5.

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 remarkable: 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 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 [72]).

(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. Versions of the second goal 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"); ω_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 ω 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 other v_j's fixed) can only increase the ith component of the computed solution. 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.

Theorem 1 ([10, 90, 105]). *An algorithm for a single-parameter mechanism design problem is implementable if and only if it is monotone.*

Theorem 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 [10].

Archer and Tardos [10] 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 [43, 58]) are not monotone and hence not implementable [10]. Archer and Tardos [7, 10] devised a randomized monotone 2-approximation algorithm for the problem, and several subsequent papers gave monotone deterministic approximation algorithms (see Kovács [70] for the best bound of 2.8 and references). Very recently, Dhangwatnotai et al. [40] proved that, allowing randomization, monotone polynomial-time algorithms are competitive with arbitrary polynomial-time algorithms for makespan minimization.

Theorem 2 ([40]). *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 [24, 38] 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 [107] is a classical characterization of

implementable algorithms in terms of a certain shortest-path condition known as *cycle monotonicity* (see [132]) that is general but difficult to use to prove upper or lower approximation bounds (see [74] for an exception). Archer and Kleinberg [8] give a promising reformulation of Rochet's Theorem that could lend itself to new approximation bounds. Saks and Yu [118] 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 [8, 87] for new alternative proofs and [85] 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 Ω is an abstract outcome set and the t_i's are arbitrary real-valued functions on Ω, *Robert's Theorem* [106] 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 $\sum_i t_i(\omega)$. This should be viewed as a negative result, since affine maximizers have limited polynomial-time approximation capabilities in most important problems (see e.g. [41]). 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 [100] for a surprising new example) and characterize the additional implementable mechanisms for domains in which Robert's Theorem breaks down (see [20, 73] and [42, 33] 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 [24]), where Ω is the ordered partitions $(S_1, \ldots, S_n)$ 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 $\sum_i v_i(S_i)$.[3] 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 valuations with complements but no substitutes and largely succeeded in designing implementable polynomial-time algorithms with approximation ratios matching the best-possible ones for arbitrary polynomial-time algorithms (assuming $P \neq NP$) [76, 89]. Some of these

[3] Valuations are typically modeled either as a "black box" that can be queried or implicitly via a compact representation of size polynomial in m; an "efficient algorithm" in this context has running time polynomial in both n and m.

guarantees have been extended to general valuations (see [24]). Unfortunately, with complements, the underlying welfare maximization problem includes the Maximum Independent Set problem as a special case and thus reasonable approximation guarantees are possible only under strong additional assumptions (as in [9, 17]).

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, excellent (constant-factor) approximation guarantees appear possible, though challenging to obtain. Beginning in [75], a number of papers have proved constant-factor upper and lower bounds for polynomial-time approximation of welfare maximization with complement-free valuations by *non-implementable* algorithms; see [47] and [135] for two recent gems. Remarkably, no constant-factor *implementable* algorithm is known for any such problem. For problems with a sum objective, welfare maximization with complement-free bidders appears to be the most likely candidate to separate the power of implementable and non-implementable algorithms. See [100] for a very recent communication complexity-based separation, 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 [56]); revenue guarantees and cost-sharing mechanism design (see [61, 83]); online mechanism design, in which participants arrive and depart over time (surveyed in [102]); and new models and goals for Internet-suitable mechanism design, such as distributed mechanisms (see [48]) and mechanisms restricted to use little [84] or no [57, 78, 122] payments.

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., truthtelling) does not depend on the actions taken by the others. This was possible because a designer (like a search engine owner) had 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 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, in order 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 imposing additional control over the system is relatively small, and is particularly reassuring when implementing an optimal solution is infeasible (as in a typical Internet application).

We only address this question for the most popular modeling choices (Nash equilibria and the price of anarchy, respectively) 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" [115]. The model is originally from [18, 136] and is thoroughly discussed in [110]; the price of anarchy was originally suggested in [69] for a scheduling model, results on which are surveyed in [131].

Consider a directed multicommodity 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 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 that minimizes $\sum_{e \in P} c_e(f_e)$, 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: $f_P > 0$ for a path P implies $\sum_{e \in P} c_e(f_e)$ is minimum over all paths with the same source and destination (if not, some selfish users using this path would switch to a cheaper one). All equilibrium flows are interchangeable in that they have equal cost — $\sum_e c_e(f_e) f_e$, as in classical minimum-cost flow — and one is guaranteed to exist [18].

For example, in a "Pigou-like network" (named after [103]), r units of selfish users decide between two 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 always has no larger cost than the first, even in the worst case when it is fully congested. For this reason, routing all flow on the second edge is

an equilibrium. This equilibrium in generally suboptimal, in that it fails to minimize the cost $\sum_{e \in P} c_e(f_e)$ over all feasible flows. 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 such a selfish routing network is the ratio of 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, at least in this simple family of examples, 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 [109]. 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 in most interesting cases. 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, matching the lower bound noted above. See [113, 112] for recent surveys detailing these and related results.

While there is no explicit design aspect to these POA bounds, they nicely justify 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 [115]. Here we provide a conceptually similar but technically different result, which is a special case of the POA bounds in [109] (see also [110, §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 network literature (e.g. [19]). We say the network is *β-overprovisioned* for $\beta \in (0, 1)$ if, at equilibrium, at least a β fraction of each edge's capacity remains unused. The following tight bound on the POA holds for such networks.

Theorem 3 (Consequence of [109]). *The POA of a β-overprovisioned network is at most $\frac{1}{2}(1 + \frac{1}{\sqrt{\beta}})$.*

Thus even 10% extra capacity reduces the 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, we would like to use inefficiency measures such as the POA to inform how to *design* systems to have good equilibria.

Two variants of this idea have been explored in a number of different models: improving the POA of a given game (see [113] 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 [32], where a number open issues remain. See [62, 117] for surveys of other work on this important topic.

We follow the network cost-allocation example in [27], which was motivated by the network formation games of [6] (see [111, 127] for relevant surveys). As in a selfish routing network, each player selects a path in a multicommodity network to minimize its incurred cost. For technical convenience, we now assume that each player controls a single (non-negligible) unit of flow and uses a single path to route it. The 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 — for installing infrastructure or leasing a large fixed amount of bandwidth, say — to 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. Our benchmark is the minimum-cost way of connected all of the players' source-sink pairs, a Steiner connectivity problem (equivalent to the minimum-cost Steiner tree problem if all players share a common sink vertex). An obvious question is: *how should we distribute costs 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 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 $P_1, \ldots, P_k$, the cost incurred by the ith 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. Shapley cost-sharing always leads to at least one equilibrium [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 [69], and this example yields a linear lower bound for the worst-case POA of Shapley cost-sharing (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 are in one-to-one

correspondence to the possible outputs of well-studied greedy online algorithms for Steiner connectivity problems [13, 59]. 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 ([27]). *In undirected cost-allocation games, ordered cost-sharing attains the minimum-possible 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 all networks. These turn out to be precisely the finite "concatenations" of weighted Shapley values (in the sense of [65]); Shapley cost-sharing is the special case of uniform weights and no concatenation, while ordered cost-sharing arises from the concatenation of k different one-player (trivial) Shapley values. No method of this type can outperform ordered cost-sharing by more than a constant factor [27].

In directed networks, it is easy to show that all cost-sharing methods, including ordered ones, have linear worst-case POA (like Shapley cost-sharing). We can obtain a more refined comparison by analyzing the ratio of the best (instead of the worst) Nash equilibrium and a minimum-cost solution, a quantity known as the price of stability (POS). The worst-case POS of Shapley cost-sharing in directed networks is precisely the kth Harmonic number $\mathcal{H}_k \approx \ln k$ [6]. A consequence of the classification above is that no other method has superior worst-case POS (or POA).

Theorem 5 ([27]). *In directed cost-allocation games, Shapley cost-sharing attains the minimum-possible worst-case POS and POA.*

Further Aspects of Quantifying Inefficiency. We have barely scratched the surface of recent work on equilibrium efficiency analyses. Many different models of routing games have studied from this perspective — following [108, 116], often in the more abstract guise of "congestion games" — see [68, 112] for an incomplete survey. See [94, Chapters 19-21] and [113] for overviews of efficiency analyses in some other application domains. See [5, 31] for efficiency analyses of equilibrium concepts other than Nash equilibria. See [15, 28, 66] for recent efficiency guarantees in models that allow altruistic and/or malicious participants, rather than only self-interested ones.

In addition to the aforementioned work on designing games with efficient equilibria, a second current and important trend in the area 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 this assumption by positing natural experimentation strategies

(or "dynamics") and proving that they quickly reach a (possibly approximate) equilibrium; see [14, 21, 30, 44, 50] for a sampling of examples. The second is to prove POA-like guarantees on system performance that apply even if such experimentation strategies fail to converge to an equilibrium. Remarkably, such bounds exist in, for example, the selfish routing networks discussed in this section; see [53, 86] and [22] 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 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 outcome?*

These questions are important for two reasons. Algorithms for equilibrium computation can be useful practically, for example in game-playing (e.g. [52]) and for multi-agent reasoning (see [124] for an introduction). Second, resolving the computational complexity of 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 [104]).

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 general games with two or more players [29, 39]. 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, as well as surveys of other central equilibrium computation problems (such as 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 the 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 $c_e : \{1, 2, \ldots, k\} \rightarrow \mathcal{R} +$, describing the per-player cost along an edge as a function of its number of users, is non-decreasing. Similarly to the cost allocation games in Section 3, in a (pure-strategy) Nash equilibrium (PNE) $P_1, \ldots, P_k$, each player simultaneously chooses a *best response*: a path with minimum-possible cost $\sum_e c_e(f_e)$, given the choices of others.

Best-response dynamics (BRD) is a simple model of experimentation by players over time: while the current outcome is not a PNE, choose an arbitrarily 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 cycles forever in many games. In an atomic selfish routing network, however, every iteration of BRD strictly decreases the *potential function* $\Phi(P_1, \ldots, P_k) = \sum_{e \in E} \sum_{i=1}^{f_e} c_e(i)$, and thus BRD is guaranteed to terminate, necessarily at a PNE [88, 108]. The number of distinct outcomes is generally exponential in the size of the network and the number of players; does convergence require polynomial or exponential time? Can we compute a PNE of such a game by other means in polynomial time?

Computing a PNE of an atomic selfish routing game is a member of $TFNP$ ("total functional NP), an intriguing class of search problems for which all instances have a (short and efficiently verifiable) witness [82]. 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 $NP = coNP$ [82]. 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 a NP search problem — and thus unlikely to contain complete problems (see [63, 125]). 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 [114] in motivating the appropriate subclass. View the definition of NP (existence of short witnesses and an efficient verifier) as a 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 games appears to be easier 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" [64]. A PLS problem is abstractly 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 new candidate solution with strictly better objective

function value. PLS can be phrased as a syntactic class and it therefore admits a generic complete problem [64]. The analog of Cook's Theorem (a reduction from the generic complete problem to a concrete one), proved in [64], states that a particular local search problem for Boolean circuits called "Circuit Flip" is PLS-hard. Circuit Flip has been used to establish the PLS-completeness of many other problems (e.g. [121, 137]).

Solutions of a PLS problem correspond to local optima, and one can obviously be found (generally in exponential time) via local search. 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 define two outcomes to be neighbors if they differ in the path of only one player. Local minima then correspond to the PNE of the game.

Solving a PLS problem means computing a locally optimal solution by whatever means (not necessarily by local search). 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 [46] 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 apparent difficult problems, it is generally believed that $P \neq PLS$. PLS-hardness should therefore viewed as strong evidence that a $TFNP$ search problem is not solvable in polynomial time. Computing a PNE of a (multicommodity) atomic selfish routing network is hard in this sense.

Theorem 6 ([46]). *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, 46, 126] for further PLS-completeness results on 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 (or BRD, in the present context). Even if $P = PLS$, the following corollary holds.

Corollary 1. *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. A collection of mixed-strategies is a *(mixed-strategy) Nash equilibrium (MNE)* if every player simultaneously chooses a mixed strategy maximizing its expected utility, given the mixed strategies chosen by the others. Resorting to mixed strategies is necessary to establish

the existence of Nash equilibria in arbitrary finite games with two or more players [91], but they are not without conceptual controversy (see e.g. [96, §3.2]). Regardless, computing an MNE of a finite game is clearly a central equilibrium computation problem.

First consider the two-player ("bimatrix") case, where the input is two $m \times n$ payoff matrices (one for each player) with integer entries. There is a non-obvious exponential-time algorithm for computing an MNE in bimatrix games, which enumerates over all possible pairs of supports for the two players and solves a linear system for each to check for a feasible solution (see e.g. [98, 114, 124]). There is a still less obvious "guided search" algorithm, the *Lemke-Howson (LH) algorithm* [77]; see [133] for a careful exposition. Its worst-case running time is exponential [120]. 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 [97] 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 (again easily phrased as three polynomial-time algorithms) necessary to execute a LH-like search procedure. $PPAD$-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 7 ([29, 39]). *The problem of computing an MNE of a bimatrix game is $PPAD$-complete.*

This hardness result trivially applies to games with any constant number of players. It extends to computing a natural notion of an "ϵ-approximate MNE" for values of ϵ as large as inverse polynomial [29], thus ruling out an FPTAS for computing ϵ-approximate MNE (unless $P = PPAD$). Unlike PLS-completeness results, $PPAD$-completeness results are not known to have immediate unconditional consequences in the spirit of Corollary 1. However, a lower bound on the convergence time of certain dynamics to an MNE was recently proved in [55] (without relying on Theorem 7).

The proof of Theorem 7 is necessarily intricate because in the result is a "Cook's Theorem for $PPAD$" — while several $PPAD$-complete problems were previously known [97], all of them have the flavor of "generic" complete problems, in which an instance includes a description of an arbitrary polynomial-time algorithm. For example, instances of $PPAD$-complete fixed-point problems included an encoding of a polynomial-time algorithm that computes the values of some continuous function restricted to a subdivided simplex. The proof of Theorem 7 effectively encodes arbitrary computation in terms of a bimatrix game, so its sophistication should come as no surprise. Many of the first "non-generic" PLS-complete problems required similarly intricate reductions

(e.g. [121]). See [98] for a nice high-level survey of the proof of Theorem 7 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. As with mixed Nash equilibria, such equilibria exist under weak conditions [11] but their efficient computation is largely open. The last five years have seen a number of new polynomial-time algorithm (surveyed in [129] and [34]) and a few scattered hardness results (see [34]), but many basic questions remain open (see [129]).

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. [134] — was studied early on by the AI community (surveyed in [67]) and more recently in the theoretical computer science literature (e.g. [85]). Special classes of extensive-form games defined in [36] are, along with some number-theoretic problems like factoring, among the most prominent candidates for problems in $(NP \cap coNP) \setminus P$ (see [63]). 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 [23]), sweeping positive algorithmic results are possible [99]. In repeated games, computing a Nash equilibrium is polynomial-time solvable in two-player games [81] but $PPAD$-hard with three or more players [25], 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 implementable algorithms for combinatorial auctions (see [24] for a reasonably current survey); characterize the multi-parameter domains for which affine maximizers are the only implementable algorithms (see [100] 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 [35] and [49], respectively, for partial results); develop a general analytical technique to extract tight efficiency loss bounds from potential functions and/or variational inequalities (see [111]); and, in the spirit of [27], 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 [26, 80, 128] for the state-of-the-art), the complexity of computing market equilibria with reasonably general (concave) participant utility functions (see [129]), and the complexity of the stochastic games in $NP \cap coNP$ defined in [36] (see also [63]).

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 assumption is 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 modern computer science applications, there would be no need for AGT at all. See [51] for a compelling argument — made over a decade ago but more appropriate than ever — about why models of rationality and equilibrium concepts should be completely rethought given the characteristics of an Internet-like strategic environment.

Behavioral assumptions are essential to address modern computer 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 [16] and in ex post collusion-proof Nash equilibrium [95]; the price of total anarchy in [22]; and the complexity of unit-recall games [45]. If history is any guide, these represent only the vanguard of what promises to be a rich and relevant theory.

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. *Mechanisms for Discrete Optimization with Rational Agents*. PhD thesis, Cornell University, 2004.
8. A. Archer and R. D. Kleinberg. Truthful germs are contagious: a local-to-global characterization of truthfulness. In *EC '08*.
9. 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.
10. A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *FOCS '01*, pages 482–491.
11. K. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22:265–290, 1954.
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, R. D. Kleinberg, and C. H. Papadimitriou. Congestion games with malicious players. In *EC '07*, pages 103–112.
16. M. Babaioff, R. Lavi, and E. Pavlov. Single-value combinatorial auctions and implementation in undominated strategies. In *SODA '06*, pages 1054–1063.
17. Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK '03*, pages 72–87.
18. M. J. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
19. D. P. Bertsekas and R. G. Gallager. *Data Networks*. Prentice-Hall, 1987. Second Edition, 1991.
20. 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.
21. 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.
22. A. Blum, M. Hajiaghayi K. Ligett, and A. Roth. Regret minimization and the price of total anarchy. In *STOC '08*.
23. A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. In Nisan et al. [94], chapter 4, pages 79–101.
24. L. Blumrosen and N. Nisan. Combinatorial auctions. In Nisan et al. [94], chapter 11, pages 267–299.
25. 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*.
26. H. Bosse, J. Byrka, and E. Markakis. New algorithms for approximate Nash equilibria in bimatrix games. In *WINE '07*, pages 17–29.
27. H. Chen, T. Roughgarden, and G. Valiant. Designing networks with good equilibria. In *SODA '08*, pages 854–863.
28. P.-A. Chen and D. Kempe. Altruism, selfishness, and spite in traffic routing. In *EC '08*.
29. X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of two-player Nash equilibria. *Journal of the ACM*, 2008.
30. S. Chien and A. Sinclair. Convergence to approximate Nash equilibria in congestion games. In *SODA '07*, pages 169–178.

31. G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *EC '05*, pages 59–70.
32. G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. In *ICALP '04*, pages 345–357.
33. G. Christodoulou, E. Koutsoupias, and A. Vidali. A characterization of 2-player mechanisms for scheduling. Submitted, 2008.
34. B. Codenotti and K. Varadarajan. Computation of market equilibria by convex programming. In Nisan et al. [94], chapter 6, pages 135–158.
35. R. Cominetti, J. R. Correa, and N. E. Stier Moses. The impact of oligopolistic competition in networks. 2008.
36. A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.
37. P. Cramton. Spectrum auctions. In *Handbook of Telecommunications Economics*, chapter 14, pages 605–639. 2002.
38. P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
39. C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of comuting a Nash equilibria. *SIAM Journal on Computing*, 2008.
40. P. Dhangwatnotai, S. Dobzinski, S. Dughmi, and T. Roughgarden. Truthful approximation schemes for single-parameter agents. Submitted, 2008.
41. S. Dobzinski and N. Nisan. Limitations of VCG-based mechanisms. In *STOC '07*, pages 338–344.
42. S. Dobzinski and M. Sundararajan. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *EC '08*.
43. L. Epstein and J. Sgall. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica*, 39(1):43–57, 2004.
44. E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibria. In *ICALP '03*, pages 502–513.
45. A. Fabrikant and C. H. Papadimitriou. The complexity of game dynamics: BGP oscillations, sink equlibria, and beyond. In *SODA '08*, pages 844–853.
46. A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *STOC '04*, pages 604–612.
47. U. Feige. On maximizing welfare when utility functions are subadditive. In *STOC '06*, pages 41–50.
48. J. Feigenbaum, M. Schapira, and S. Shenker. Distributed algorithmic mechanism design. In Nisan et al. [94], chapter 14, pages 363–384.
49. 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.
50. S. Fischer and B. Vöcking. On the evolution of selfish routing. In *ESA '04*, pages 323–334.
51. E. J. Friedman and S. J. Shenker. Learning and implementation on the Internet. Working paper, 1997.
52. 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*.
53. M. X. Goemans, V. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *FOCS '05*, pages 142–151.
54. 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.
55. S. Hart and Y. Mansour. The communication complexity of uncoupled Nash equilibrium procedures. *Games and Economic Behavior*, 2008.
56. J. Hartline and A. Karlin. Profit maximization in mechanism design. In Nisan et al. [94], chapter 13, pages 331–362.
57. J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *STOC '08*.

58. 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.
59. M. Imase and B. M. Waxman. Dynamic Steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3), 1991.
60. M. O. Jackson. A crash course in implementation theory. *Social Choice and Welfare*, 18(4):655–708, 2001.
61. K. Jain and M. Mahdian. Cost sharing. In Nisan et al. [94], chapter 15, pages 385–410.
62. R. Johari. The price of anarchy and the design of scalable resource allocation mechanisms. In Nisan et al. [94], chapter 21, pages 543–568.
63. D. S. Johnson. The NP-completeness column: Finding needles in haystacks. *ACM Transactions on Algorithms*, 3(2), 2007. Article 24.
64. 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.
65. E. Kalai and D. Samet. On weighted Shapley values. *International Journal of Game Theory*, 16(3):205–222, 1987.
66. G. Karakostas and A. Viglas. Equilibria for networks with malicious users. *Mathematical Programming*, 110(3):591–613, 2007.
67. D. Koller and A. Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1-2):167–215, 1997.
68. S. C. Kontogiannis and P. G. Spirakis. Atomic selfish routing in networks: A survey. In *WINE '05*, pages 989–1002.
69. E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS '99*, pages 404–413.
70. A. Kovács. Tighter approximation bounds for LPT scheduling in two special cases. In *CIAC*, pages 187–198, 2006.
71. S. Lahaie, D. Pennock, A. Saberi, and R. Vohra. Sponsored search auctions. In Nisan et al. [94], chapter 28.
72. R. Lavi. Computationally efficient approximation mechanisms. In Nisan et al. [94], chapter 12, pages 301–329.
73. R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *FOCS '03*, pages 574–583.
74. R. Lavi and C. Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *EC '07*, pages 252–261.
75. B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *EC '01*, pages 18–28.
76. 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.
77. C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *SIAM Journal*, 12(2):413–423, 1964.
78. H. Levin, M. Schapira, and A. Zohar. Interdomain routing and games. In *STOC '08*, 2008.
79. 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.
80. R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *EC '03*, pages 36–41.
81. M. L. Littman and P. Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39(1):55–66, 2005.
82. N. Megiddo and C. H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.
83. A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*, pages 1–10, 2007.

84. 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.
85. P. B. Miltersen and T. B. Sørensen. Fast algorithms for finding proper strategies in game trees. In *SODA '08*, pages 874–883.
86. V. S. Mirrokni and A. Vetta. Convergence issues in competitive games. In *APPROX '04*, pages 183–194.
87. D. Monderer. Monotonicity and implementability. In *EC '08*.
88. D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.
89. A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI '02*, pages 379–384.
90. R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.
91. J. F. Nash. Equilibrium points in N-person games. *Proceedings of the National Academy of Science*, 36(1):48–49, 1950.
92. N. Nisan. Introduction to mechanism design (for computer scientists). In Nisan et al. [94], chapter 9, pages 209–241.
93. N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1/2):166–196, 2001.
94. N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.
95. N. Nisan, M. Schapira, and A. Zohar. Best-reply mechanisms. Working paper, 2008.
96. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
97. C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48:498–532, 1994.
98. C. H. Papadimitriou. The complexity of finding nash equilibria. In Nisan et al. [94], chapter 2, pages 29–51.
99. C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multiplayer games. *Journal of the ACM*, 2008.
100. C. H. Papadimitriou, M. Schapira, and Y. Singer. On the hardness of being truthful. Manuscript, 2008.
101. D. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, University of Pennsylvania, 2001.
102. D. C. Parkes. Online mechanisms. In Nisan et al. [94], chapter 16, pages 411–439.
103. A. C. Pigou. *The Economics of Welfare*. Macmillan, 1920.
104. 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.
105. J. Riley and W. Samuelson. Optimal auctions. *American Economic Review*, 71:381–92, 1981.
106. K. Roberts. The characterization of implementable choice rules. In J.-J. Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–349. 1979.
107. J. C. Rochet. A necessary and sufficient condition for rationalizability in a quasilinear context. *Journal of Mathematical Economics*, 16:191–200, 1987.
108. R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.
109. T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.
110. T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
111. T. Roughgarden. Potential functions and the inefficiency of equilibria. In *Proceedings of the International Congress of Mathematicians*, volume III, pages 1071–1094, 2006.
112. T. Roughgarden. Routing games. In Nisan et al. [94], chapter 18, pages 461–486.
113. T. Roughgarden. Selfish routing and the price of anarchy. *OPTIMA*, 74:1–15, 2007.
114. T. Roughgarden. Computing equilibria: A computational complexity perspective. *Economic Theory*, 2008.

115. T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.
116. T. Roughgarden and É. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 49(2):389–403, 2004.
117. T. Roughgarden and É. Tardos. Introduction to the inefficiency of equilibria. In Nisan et al. [94], chapter 17, pages 443–459.
118. M. E. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *EC '05*, pages 286–293.
119. T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1):1–54, 2002.
120. R. Savani and B. von Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74(2):397–429, 2006.
121. A. A. Schäffer and M. Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.
122. J. Schummer and R. V. Vohra. Mechanism design without money. In Nisan et al. [94], chapter 10, pages 243–265.
123. Y. Shoham. Computer science and game theory. *Communications of the ACM*, 2008.
124. Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2008.
125. M. Sipser. On relativization and the existence of complete sets. In *ICALP '82*, pages 523–531.
126. A. Skopalik and B. Vöcking. Inapproximability of pure Nash equilibria. In *STOC '08*.
127. É. Tardos and T. Wexler. Network formation games and the potential function method. In Nisan et al. [94], chapter 19, pages 487–516.
128. H. Tsaknakis and P. G. Spirakis. An optimization approach for approximate Nash equilibria. In *WINE '07*, pages 42–56.
129. V. V. Vazirani. Combinatorial algorithms for market equilibria. In Nisan et al. [94], chapter 5, pages 103–134.
130. W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.
131. B. Vöcking. Selfish load balancing. In Nisan et al. [94], chapter 20, pages 517–542.
132. R. Vohra. Paths, cycles and mechanism design. Working paper, 2007.
133. B. von Stengel. Computating 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.
134. B. von Stengel. Equilibrium computation for two-player games in strategic and extensive form. In Nisan et al. [94], chapter 3, pages 53–78.
135. J. Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC '08*.
136. 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.
137. M. Yannakakis. Computational complexity. In E. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, chapter 2, pages 19–55. 1997.