Learning Game-theoretic Equilibria via Query Protocols[*]

Paul W. Goldberg[†]

Abstract

Query complexity is a very widespread and recurring theme in the analysis of algorithms and computational complexity. Algorithms are assumed to have access to their input data via certain stylised *queries*, which impose a constraint on the way an algorithm can behave. In the context of computing equilibria of games, this is a relatively recent line of work, which we review here. The talk mostly focuses on the paper [6].

1 Background and generalities

Algorithmic Game Theory (AGT) has introduced the ideas of viewing games and auctions through the lenses of *computational complexity*, and *price of anarchy*; these are very prominent themes in AGT. Here, we try to give *query complexity* more recognition as a research theme. I hope to develop the following discussion into a more detailed survey.

There are many reasons for studying query complexity, some of which are specific to game-theoretic contexts, and others which have been articulated in earlier works — query complexity is a widespread theme in algorithms and computational complexity. We note the following:

- There may be some function on which we wish to avoid imposing any kind of constraint (computational or syntactic), other than certain constraints (such as monotonicity) that would not, on their own, allow the function to have a concise representation. As a result, the size of the "input data" may be exponentially large, when expressed as a function of parameters such as the number of players, or the number of items being sold/traded. In that situation it is standard to assume that the data is presented as an "black-box oracle" (for example, in the context of cake-cutting). In this kind of setting, any polynomial-time algorithm implicitly gives polynomial query complexity. Note, however, that most of this work is mainly focused on computational complexity rather than query complexity.

[†]Oxford University, UK `Paul.Goldberg@cs.ox.ac.uk`

- When the entity being queried is an agent, or player, it may be considered onerous to have to answer many queries (this consideration has been noted in various previous works in the context of matchings; also in the context of auctions). Then it is important to elicit information efficiently, and avoid asking for information that will not be used. A related point, is that a full revelation of bidders' preferences may elicit more information than is needed to allocate an item efficiently. Bidders may prefer to be economical with the truth[1], due to concerns about privacy, or due to concerns about their valuations being exploited against them, in subsequent auctions or negotiations.

- It is sometimes a good model for how data is made available to an algorithm.

- It imposes a benign constraint on how the algorithm may interact with the data, but with that constraint, one can obtain upper/lower bounds that are sometimes quite detailed and informative, serving as a criterion to distinguish between alternative solution concepts.[2]

- General understanding of algorithms being presented: for example, [14, 10] show how to compute exact CE of multiplayer games; it's useful to observe that the information obtained by these algorithms is the payoffs obtained by the players in response to mixed-strategy profiles constructed by the algorithm.

- It has a strong relationship with *communication complexity*, which in turn has been studied in a game-theoretic context; e.g. the "bisection auction" is advocated as having low communication complexity.

Comparing computational complexity against query complexity, it is often easier to obtain lower bounds for query complexity. Indeed, query complexity may be used as a criterion to separate, or distinguish, the difficulty of alternative problems, in situations where computational complexity fails to do so.

2   Details on payoff queries

The main focus of the talk is on equilibrium computation in the *payoff query* model. This has been studied in the papers [2, 3, 6, 7, 8, 9, 12], but the earliest reference (that I know of) to the idea is the blog post of [13]. It has been noted in [4] that [14] can be seen as a payoff-query algorithm, but the focus of [14] is on queries to mixed-strategy profiles rather than pure ones.

---

[1]In a more literal sense than the common usage of this phrase

[2]For example, separation of randomized and deterministic computation [1, 11] in the context of computing boolean functions. Ambainis et al. [1] note: "The advantage of query complexity is that we can often prove tight lower bounds and have provable separations between different computational models. This is in contrast to the Turing machine world where lower bounds and separations between complexity classes often have to rely on unproven assumptions."

That would appear to be a technical-looking distinction, but [14] requires *exact* payoffs for mixed-strategy queries to be obtained, and random sampling only obtains (additively) approximate ones. The focus of the talk on Fearnley et al. [6] which is the first published paper to study the pure-strategy payoff-query model. Besides to motivations mentioned above, the motivation for this paper came from experimental work (mainly by Wellman and co-authors) on *empirical game-theoretic analysis*. The following is an overview of the results of [6].

We study a variety of different settings. We first consider bimatrix games. Our first result is a lower bound for computing an exact Nash equilibrium: we show that computing an exact Nash equilibrium in a $k \times k$ bimatrix game has payoff query complexity k^2, even for zero-sum games. In other words, we have to query every pure strategy profile.

We then turn our attention to approximate Nash equilibria, where we obtain some more positive results. With the standard assumption that all payoffs lie in the range $[0, 1]$, we show that, when $2 \leq i \leq k - 1$, the payoff query complexity of computing a $(1 - \frac{1}{i})$-approximate Nash equilibrium is at most $2k - i + 1$ and at least $k - i + 1$. We also observe that, when $\epsilon \geq 1 - \frac{1}{k}$, no payoff queries are needed at all, because an ϵ-Nash equilibrium is achieved when both players mix uniformly over their pure strategies.

The query complexity of computing an approximate Nash equilibrium when $\epsilon < \frac{1}{2}$ appears to be a challenging problem, and we provide an initial lower bound in this direction: we show that the payoff query complexity of finding a ϵ-approximate Nash equilibrium for $\epsilon = \mathcal{O}(\frac{1}{\log k})$ is $\Omega(k \cdot \log k)$. This gives an interesting contrast with the $\epsilon \geq \frac{1}{2}$ case. Whereas we can always compute a $\frac{1}{2}$-approximate with $2k - 1$ payoff queries, there exists a constant $\epsilon < \frac{1}{2}$ for which this is not the case.

Having studied payoff query complexity in bimatrix games, it is then natural to look for improved payoff query complexity results in the context of "structured" games. In particular, we are interested in *concisely represented* games, where the payoff query complexity may be much smaller than the number of pure strategy profiles. As an initial result in this direction, we consider graphical games, where we show that for graphical games with constant degree d, a Nash equilibrium can be found with a polynomial number of payoff-queries. This algorithm works by discovering every payoff in the game, however unlike bimatrix games, this can be done without querying every pure strategy profile.

Finally, we focus on two different models of congestion games. We consider the case of *parallel links*, where the game has a origin and destination vertex, and m parallel links between them. We show both lower and upper bounds for this setting. If n denotes the number of players, then we obtain a $\log(n) + m$ payoff query lower bound, which applies to both query models. We obtain an upper bound of $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log \log(m)} + m\right)$ normal-queries. Note that there are $n \cdot m$ different payoffs in a parallel links game, and so our upper bound implies that you do not need to discover the entire payoff function in order to solve a parallel links game.

Subsequently we consider the more general case of symmetric network

congestion games on directed acyclic graphs. We show that if the game has m edges and n players, then we can find a Nash equilibrium using $m \cdot n$ payoff queries. The algorithm discovers every payoff in the game, but it only queries a small fraction of the pure strategy profiles.

3 Conclusions and further work

We first consider open questions in the setting of payoff queries, which has been the main setting for the results presented here. We then consider alternative query models.

Open questions concerning payoff queries. In the context of strategic-form games, there are a number of open problems. In [6], we show a super-linear lower bound on the payoff query complexity when ϵ is allowed to depend on k. Can we prove a super-linear lower bound for a constant ϵ? Is there a deterministic algorithm that can find an ϵ-Nash equilibrium with $\epsilon < \frac{1}{2}$ without querying the entire payoff matrices? [7] achieve $\epsilon < \frac{1}{2}$ with the use of randomization, but doing so with a deterministic algorithm appears to be challenging. Finally, when $2 \leq i \leq k-1$, we have shown that the payoff query complexity of finding a $(1 - \frac{1}{i})$-Nash equilibrium lies somewhere in the range $[k-i+1, 2k-i+1]$. Determining the precise payoff query complexity for this case is an open problem.

For congestion games, our lower bound of $\log n + m$ arises from a game with two parallel links and a one-player game with m links. The above-noted upper bound (on the number of normal queries) is a poly-logarithmic factor off from this lower bound, with the factor depending on m. Can this factor be improved? It seems unlikely that the dependence of this factor on m can be completely removed, in which case, in order to provide tight bounds, a single lower bound construction that depends simultaneously on n and m would be necessary.

For symmetric network congestion games on DAGs it is unclear whether the payoff query complexity is sub-linear in n. Non-trivial lower and upper bounds for more general settings, such as asymmetric network congestion games (DAG or not) or general (non-network) congestion games would also be interesting.

Other query models. We have defined a payoff query as given by a *pure* (not mixed) profile s, since that is of main relevance to empirical game-theoretic modelling. Furthermore, if s was a mixed profile, it could be simulated by sampling a number of pure profiles from s and making the corresponding sequence of pure payoff queries. An alternative definition might require a payoff query to just report a single specified player's payoff, but that would change the query complexity by a factor at most n.

Our main results have related to exact payoff queries, though other query models are interesting too.

A very natural type of query is a *best-response query*, where a strategy s is chosen, and the algorithm is told the players' best responses to s. In

general s may have to be a mixed strategy; it is not hard to check that pure-strategy best response queries are insufficient; even for a two-player two-action game, knowledge of the best responses to pure profiles is not sufficient to identify an ϵ-Nash equilibrium for $\epsilon < \frac{1}{2}$. *Fictitious Play* can be regarded as a query protocol that uses best-response queries (to mixed strategies) to find a Nash equilibrium in zero-sum games, and essentially a 1/2-Nash equilibrium in general-sum games. We can always synthesize a pure best-response query with $n(k-1)$ payoff queries. Hence, for questions of polynomial query complexity, payoff queries are at least as powerful as best-response queries.

Are there games where best-response queries are much more useful than payoff queries? If k is large then it is expensive to synthesize best-response queries with payoff queries. A simple algorithm of Daskalakis, Mehta and Papadimitriou finds a $\frac{1}{2}$-Nash equilibrium via only two best-response queries, whereas Theorem 7 of [6] notes that $\mathcal{O}(k)$ payoff queries are needed.

A *noisy* payoff query outputs an observation of a random variable taking values in $[0, 1]$ whose expected value is the true payoff. Alternative versions might assume that the observed payoff is within some distance ϵ from the true payoff. Noisy query models might be more realistic, and they are suggested by by the experimental papers on querying games. However in a theoretical context, one could obtain good approximations of the expected payoffs for a profile s, by repeated sampling.

Recently, Chen et al. [5] showed that for general n-player games, there is an exponential lower bound on the number of payoff queries required to compute an ϵ-Nash equilibrium, for constant ϵ. This answered a question of [12]; previous an exponential lower bound was known for the stronger solution concept of ϵ-well-supported Nash equilibrium [2]. These results indicate that for large games, some kind of structure has to be assumed for their payoff function, in order to obtain a positive result for query complexity.

References

[1] A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, and J. Smotrovs. Separations in Query Complexity Based on Pointer Functions. *ECCC Report 98* (2015).

[2] Y. Babichenko. Query Complexity of Approximate Nash Equilibrium. *ArXiv tech rept.* 1306.6686 (2013); also in STOC 2014.

[3] Y. Babichenko and S. Barman. Query complexity of correlated equilibrium. *ArXiv tech rept.* 1306.2437 (2013).

[4] U. Bhaskar, K. Ligett, L.J. Schulman, and C. Swamy. Achieving Target Equilibria in Network Routing Games without Knowing the Latency Functions *ArXiv tech rept.* 1408.1429 (2014).

[5] X. Chen, Y. Cheng, and B. Tang. Well-Supported versus Approximate Nash Equilibria: Query Complexity of Large Games *ArXiv tech rept.* 1511.00785 (2015).

[6] J. Fearnley, M. Gairing, P.W. Goldberg and R. Savani. Learning Equilibria of Games via Payoff Queries. *Journal of Machine Learning Research* Vol. 16, pp. 1305–1344 (2015).

[7] J. Fearnley and R. Savani. Finding Approximate Nash Equilibria of Bimatrix Games via Payoff Queries. In *Proc. of 15th ACM-EC.* pp. 657–674 (2014).

[8] P.W. Goldberg and A. Roth. Bounds for the Query Complexity of Approximate Equilibria. *ACM Transactions on Economics and Computation* 4(4), pp. 24:1–25 (2016).

[9] P.W. Goldberg and S. Turchetta. Query Complexity of Approximate Equilibria in Anonymous Games. *ArXiv tech rept.* 1412.6455 (2015).

[10] A.X. Jiang and K. Leyton-Brown. Polynomial-time Computation of Exact Correlated Equilibrium in Compact Games. *Procs. of ACM-EC*, 119–126 (2011). *GEB* (2013).

[11] S. Mukhopadhyay and S. Sanyal. Towards Better Separations between Determinstic and Randomized Query Complexity. *ECCC Report 107* (2015).

[12] S. Hart and N. Nisan. The Query Complexity of Correlated Equilibria. *6th SAGT*; *ArXiv tech rept.* 1305.4874 (2013).

[13] N. Nisan. `http://agtb.wordpress.com/2009/11/19/the-computational-complexity-of-pure-nash/` (2009).

[14] C.H. Papadimitriou and T. Roughgarden. Computing Correlated Equilibria in Multi-Player Games. *Journal of the ACM* 55(3), article 14 (2008).