Algorithmic Pricing Agents and Tacit Collusion: A Technological Perspective

Ashwin Ittoo[*] and Nicolas Petit[**]

Introduction

Amongst the wealth of concerns raised by Artificial Intelligence ("AI"), one is the risk that the deployment of algorithmic pricing agents on markets will increase occurrences of tacit collusion by orders of magnitude, and well beyond the oligopoly setting where such markets failures have been traditionally observed. This concern has already generated policy interest, and regulatory options are now commonly discussed at academic, commercial and official conferences. At the same time, however, we remain in lack of understanding of whether current AI technology holds the capabilities that entitle algorithmic pricing agents to autonomously enter into tacitly collusive strategies without human intervention. In this paper, we look at three plain-vanilla Reinforcement Learning ("RL") technologies, and attempt to understand whether their introduction at scale on markets can lead to tacit collusion. While we do not deny the fact that smart pricing agents can enter into tacit collusion and that regulators may be right to be vigilant, we find that there are several technological challenges in the general realm of RL that mitigate this risk.

Our paper proceeds in five steps. We first discuss the algorithmic tacit collusion conjecture (I). We then provide a non technical overview of reinforcement learning technologies (II). We then move on to discuss how naïve single agent Q-learning (III) and multi-agent Q-learning (IV) interact as market players. We close with a discussion of how technological challenges fragilize the algorithmic tacit collusion conjecture (V).

I. The Algorithmic Tacit Collusion Conjecture

The claim has recently been made that the generalization of pricing algorithms on markets, and in particular of AI in pricing algorithms, will make tacit collusion more common.[1] Tacit collusion occurs when oligopolists coordinate their prices (and/or any other variable) and jointly achieve supra-competitive profits, without the adoption of any institutional arrangement (a contract, a combination, an agreement, a joint-venture, a trade association, etc.).[2] Tacit collusion represents a deviation from the competitive equilibrium. It generates reductions in welfare similar in nature to those caused by cartels. Tacit collusion is a Nash equilibrium the emergence of which demands strict conditions to be met. Oligopolists must be able to effectively detect and punish any deviation by another oligopolists from the collusive equilibrium. Otherwise, each oligopolist will have an incentive to cheat on its rivals, and prices will revert to the competitive level.

---

[*] Associate Professor of Information Systems/Analytics, HEC Liège, ULiege ashwin.ittoo@ulg.ac.be.

[**] Professor, School of Law, ULiege and Research Professor, University of South Australia nicolas.petit@ulg.ac.be. We are grateful to Damien Ernst, Joe Harrington and Mark Patterson for helpful comments. All mistakes remain ours.

[1] A Ezrachi and ME Stucke. *Virtual Competition: The Promise and Perils of the Algorithm-Driven Economy* (2016). SK Mehra, 'Antitrust and the Robo-Seller: Competition in the Time of Algorithms' (2016) 100 Minn. L. Rev., 1323-1375.

[2] It is also known as "*conscious parallelism*" or "*oligopolistic interdependence*".

In economics, the concept of tacit collusion is well admitted. Since the early intuitions of Chamberlin in 1929,[3] non cooperative game theory has shown that when firms meet repeatedly and for an infinite amount of times, tacit collusion is a likely equilibrium under certain restricted conditions.[4] A point of relative agreement in the literature is that tacit collusion is only sustainable in relatively concentrated markets. That said, most papers do not take a definitive view on the *n* number of oligopoly firms at which tacit collusion occurs. Selten once famously wrote that "*4 are few and 6 are many*".

With algorithms, several pieces of legal scholarship have advanced the conjecture that tacit collusion will be sustainable in markets which no longer need significant levels of oligopoly concentration. Ezrachi and Stucke write that an "*industry's shift to pricing algorithms can spread tacit collusion beyond duopolies to markets with five or six large firms*".[5] In a paper on "*Robosellers*" to which firms delegate pricing, Merah sustains that "*automated pricing via algorithmic processing of collected mass data may tend to lead pricing above the competitive level, either via tacit collusion or more robust cartel formation*".[6] And a review paper of the OECD indicates that "*algorithms might affect some characteristics of digital markets to such an extent that tacit collusion could become sustainable in a wider range of circumstances possibly expanding the oligopoly problem to non-oligopolistic market structures*".[7]

This literature remains, however, scant on the empirical evidence of algorithmic tacit collusion. Admittedly, the legal scholarship has attempted to document what it perceives to be perverse instatiations of the problem. Yet, the evidence adduced remains below the threshold of accuracy, consistency, reliability and exhaustiveness necessary to drive the point home. Besides stories from mainstream news agencies and press outlets, most of the evidence consists of extrapolations from past or ongoing cases where the use of online technology on concentrated markets has introduced enough transparency to ultimately degenerate into man-made tacit collusion.[8] For instance, Ezrachi and Stucke discuss the case of retail petrol oligopoly markets in Germany, Chile and Australia where Government mandated online price posting. They show how Government-imposed real-time pricing obligations paradoxically facilitated tacit collusion. With this, they draw a parallel with the similar effect that could obtain with introduction of pricing algorithms in oligopoly markets. But in all those cases, several essential determinants conducive to tacit collusion were arguably already present, and the use of algorithmic technology simply removed the last obstacle to it.

---

[3] E. H. Chamberlin, "Duopoly: Value Where Sellers Are Few", (1929) 44 *Quarterly Journal of Economics*, 63,
p.65.

[4] S. Feuerstein, "Collusion in Industrial Economics – A Survey", (2005) 5(3) *Journal of Industry, Competition and Trade,* 163-198,

[5] A. Ezrachi and ME Stucke, *Algorithmic Collusion: Problems and Counter-Measures – Note*, Roundtable on Algorithms and Collusion 21-23 June 2017 (adding: "*the nature of electronic markets, the availability of data, the development of similar algorithms, and the stability and transparency they foster, will likely push some markets that were just outside the realm of tacit collusion into interdependence*").

[6] S Mehra, "Antitrust and the Robo-Seller: Competition in the Time of Algorithms", 100 Minnesota Law Review, 2015.

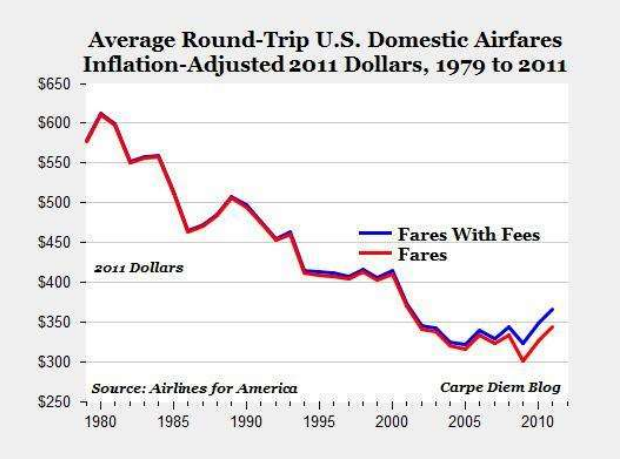
[7] OECD, Algorithms and Collusion - Background Note by the Secretariat, 21-23 June 2017.

[8] A related set of cases brought as evidence in the discussion features the use of algorithms to support a previouslay agreed collusive scheme. See, OECD, *Algorithms and Collusion*, Background Note by the Secretariat, DAF/COMP(2017)4 | (in particular, Box 9 and the discussion of the Topkins case), available at https://one.oecd.org/document/DAF/COMP(2017)4/en/pdf

What this means is subtle, but critical, namely that algorithms were not determinatively, and perhaps not even significantly causal of tacit collusion.

A second, and equally crucial point, is that none of those cases features *humanless* tacit collusion. This is important, because at the heart of the algorithmic tacit collusion conjecture lies a situation in which pricing and quantity decisions are *entirely* delegated to algorithm-driven robo-sellers.[9] In other words, we still remain in lack of understanding of whether algorithmic pricing agents can autonomously enter into tacitly collusive strategies, without human intervention.

Those uncertainties that we stress do not arise in a vacuum. Pricing algorithms have been brought to bear on many markets for years, and available anecdotal evidence does not lean in support of the conjecture. Across past decades, the long term price levels observed in algorithmic-intense industries like airlines[10] and electricity[11] denote a steady decline (see graphs below). True that such trends do not preclude periodical occurrences of tacit collusion. Yet, the tacit collusion conjecture envisions a systemic tacit collusion effect, which is not borne out by historical evidence. In fact, regulators have tinkered with those markets in order to improve the coordination of market players, which is not the state of nature.

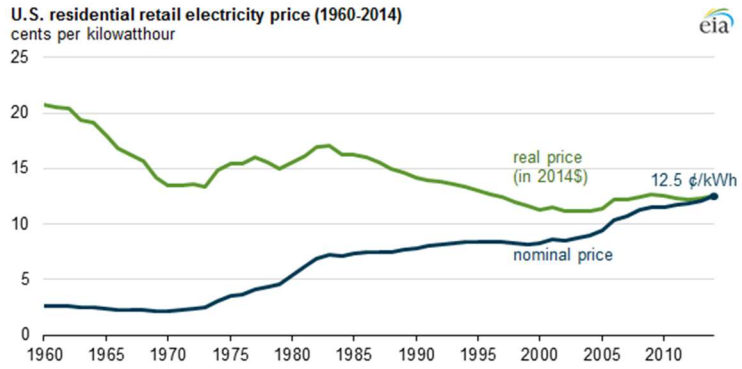


Source: D. Thompson, The Atlantic, 2013

[9] S. Mehra, *supra*.

[10] D. Thompson, The Atlantic, "How Airline Ticket Prices Fell 50% in 30 Years (and Why Nobody Noticed)", 28 February 2013, available at https://www.theatlantic.com/business/archive/2013/02/how-airline-ticket-prices-fell-50-in-30-years-and-why-nobody-noticed/273506/

11 U.S. Energy Information Administration, *Monthly Energy Review*, 16 March 2015, available at https://www.eia.gov/todayinenergy/detail.php?id=20372



Source: U.S. Energy Information Administration, 2015

As if this was not all, a growing number of anecdotes upset the idea that algorithms are good at coordinating. The United Airlines ("UA") passenger removal scandal is a case in point. What we see here is in reality two things. *First*, we observe a coordination failure between distinct algorithms – a corporate scheduling algorithm, a corporate financial algorithm and a customer value algorithm – used in a same firm to allocate overbooked seats, which take erroneous decisions sequentially without looking at the big picture.[12] *Second*, we see a firm that experiences algorithmic tyranny by delegating its entire decision-making process to algorithms, by contrast to other – and many firms – who keep a human in the loop to avoid market backlash.

Short of empirical evidence, discrete attempts have been made by economists to discuss the issue in formal terms. In an early paper, Schinkel discusses a bootlegged version of the video game *Ms. PacMan* which features three ghosts whose mission is to catch Ms. PacMan.[13] In the game, the ghosts have creativity. As a result, the ghosts always win. Yet, Schinkel does not discuss whether the ghosts resort to coordination to achieve this equilibrium.

Salceldo develops a more realistic model in which two firms use algorithms to set prices.[14] An important feature is that at localized points in time, each duopolist is able to "*decode*" the other's algorithm, and revise its own algorithm in response. The hypothesis studied in the paper posits that player 1 moves to adopt an algorithm that prices competitively, but is programmed to match any price increases by its rival. In this scenario, the best response of player 2 is to read this as a "*proposal*" to increase prices in parallel. As long as player 1 cannot revise its algorithm too quickly, player 2 will read this as a firm commitment to increase prices, and thus move away from the competitive equilibrium price. In the model, prices are fixed in the short run, so that algorithms can only be revised infrequently. The model's spectacular conclusion is that collusion is "*inevitable*". In addition, it suggests that this equilibrium is robust to an increase in the number of firms.

However, Salceldo's competition v imitation optimization function describes a restricted type of signalling competition with credible commitments in oligopoly markets. A more general specification robust to various forms of oligopoly markets

[12] C. Perez, "How Algorithms and Authoritarianism Created a Corporate Nightmare at United", Medium, 14 April 2017, available at https://medium.com/intuitionmachine/how-algorithms-and-authoritarianism-created-a-corporate-nightmare-at-united-92d9bbdf1144

[13] Schinkel, Maarten Pieter. *Market oversight games*. Vol. 378. Amsterdam University Press, 2011.

[14] Salcedo, Bruno. "Pricing Algorithms and Tacit Collusion." *Manuscript, Pennsylvania State University* (2015).

would feature an optimizing algorithm with a general profit maximization function. The algorithm would be programmed to select a price level, not simply to make a binary choice between zero profits (competition) and full coordination (imitation). Besides, one can question whether the assumptions of decoding and of instant optimization are realistic. He wonders whether Q-learning systems can approximate this. Last, Salcedo's paper does not discuss the impact of competitive entry.

Perhaps more progress is to expect from the fields of algorithmic game theory and artificial economics, which lies at the interesection of economics and computer science. In this emerging field, a seminal experiment has shown that RL algorithmic agents in a Cournot oligopoly can reach a tacit collusion equilibrium.[15] Yet, subsequent studies have demonstrated that this finding is not robust to small perturbations in costs, prices or other parameters, thereby suggesting that the constraints that undermine tacit collusion amongst human agents are equally present for RL algorithmic agents.[16] This finding may not be surprising since the main inspiration for temporal-difference learning principle – upon which Q-learning algorithm is based – is the learning process in animals/humans.

With this background, we investigate the tacit collusion conjecture from a technological perspective. We provide an analysis of a subset of mainstream RL techniques called Q-learning agents. We chose them because Q-learning agents can learn about their environment and competitors, and potentially, enter into tacit collusion.[17] Our goal is to objectively assess the actual collusive capabilities of RL algorithms on markets.

II. Overview of Reinforcement Learning

Our study focuses on reinforcement learning algorithms. RL is a suitable framework to study the interaction of profit maximizing algorithmic agents because it shares many analogies with the situation of oligopolists in markets:

- RL is a trial-and-error approach. With RL, a pricing agent thus has to try different strategies (policies/actions) before deciding which one to choose
- A RL pricing agent faces an exploration v exploitation tradeoff in choosing the next action, which can be compared to pricing tradeoffs in oligopoly markets
- Rewards are delayed as a RL pricing agent does not immediately know whether it chose the best policy (i.e. whether they took the most "*optimal*" action), in line with the idea that prices on markets do not adjust instantly

[15] Waltman, Ludo, and Uzay Kaymak. "Q-learning agents in a Cournot oligopoly model." Journal of Economic Dynamics and Control 32.10 (2008): 3275-3293.
[16] Izquierdo, Segismundo S., and Luis R. Izquierdo. "The "Win-Continue, Lose-Reverse" rule in Cournot oligopolies: robustness of collusive outcomes." Advances in Artificial Economics. Springer, Cham, 2015. 33-44.
[17] Note that other studies on algorithms and collusion that do not reach completely robust results have been produced recently, yet they do not cover Q-learning algorithms. See, for instance, Crandall, Jacob W., et al. "Cooperating with Machines." *arXiv preprint arXiv:1703.06207* (2017). And for an overview, see Deng, Ai, (2017). "When Machines Learn to Collude: Lessons from a Recent Research Study on Artificial Intelligence", available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3029662

- A RL pricing agent tries to learn the best actions to outperform its opponents or to reach a consensus (Nash equilibrium of tacit collusion). In that respect, RL encourages or discourages certain actions via positive or negative feedback

In RL, a pricing agent learns by interacting with its environment, which can be assumed to be dynamic. At any given time, t, the agent is considered to be in a state, *s*. It chooses an action, *a*, to execute, e.g. to raise its price, which results in a corresponding change in s. The agent receives a reward, which denotes the utility of making this decision.

One of the most popular RL techniques is the Q-learning algorithm. It has been extensively studied in the past. Its simplicity and ease-of-use has made it a *de-facto* choice in a diverse range of applications, such as robotic control, video games or ad-placement on the web. The Q-Learning algorithm has also been commonly employed for dynamic pricing applications like airline fares or bids placed on wholesale electricity markets.

However, and as can be seen from those examples, most existing applications of Q-learning algorithms consist in what can be called single-agent Q-learning. They are unsuitable for multi-agent environments as they ignore the other agents' rewards. Therefore, as we discuss below, a single-agent configuration is often too restrictive for real-world, complex problems, such as dynamic pricing in oligopolistic markets, with n potentially competing agents. Following a brief description of the single agent Q-learning algorithm – which we call naïve Q-learning – we move on to discuss the case of multiple agents (multi-agent). Specifically, we discuss the MinMax Q-learning algorithm for zero-sum games, and the NashQ Learning for general-sum games.

III. **Naïve Q-Learning Algorithm**

In essence, a naïve Q-learning algorithm is a profit-maximizing agent which, in any given state s, chooses an action a that yields the maximum utility. The utility is often expressed as $Q(s, a)$. To compute this value, the agent takes into account the immediate reward for taking the action as well as the discounted long-term reward. The latter is often operationalized as the Q value in the resulting state.

Formally, the (naïve) Q-learning procedure can be written as follows:

- For each state-action pair, initialize a Q-table (matrix), for e.g. such that each values in the table is 0 initially
- At each time step, the agent, in state s, chooses an action a which maximizes $Q(s, a)$ if the agent is in an exploitaton mode. It can choose another action, if he decides to explore its environment.
- This action will generate an immediate reward and will transform the state to move from s to s'. A corresponding table entry is then updated with the immediate and discounted estimated future rewards, as indicated in the equation below

$$Q_{t+1}(s,a) = (1-\alpha_t)Q_t(s,a) + \alpha_t[r_t + \beta \max_b Q_t(s',b)]$$

where,

- r_t: immediate reward, available from the payoff matrix.
- $\max_b Q_t(s', b)$: highest utility (Q) value of action in next state
- α: learning rate
- β: discount factor

The above procedure is repeated until a terminal state is reached.

In essence, a Q-learning agent attempts to find the best actions, known as the optimal policy, π.

As mentioned earlier, (naïve) Q-learning has been predominantly employed in single-agent applications. For instance, (naïve) Q-learning is used as decision support system for central authorities tasked with allocation choices: electricity capacity, traffic management, logistics, etc. Moreover, (naïve) Q-learning can also assist decentralized players make profit maximizing decisions, without however considering other agents strategies. A common example is yield management by airlines. While ticket pricing is beyond the control of an airline company, a significant influence can be exercised through Q-learning algorithms over seat allocation and overbooking.[18] However, in oligopoly markets, we contemplate the involvement and interaction of several decentralized agents.

To apply (naïve) Q-learning in oligopoly markets, one possibility would be to straightforwardly apply Q-learning to each of the agents. However, this approach has a major limitation[19]: the environment is non-stationary due to the presence of other agents, who are all adapting their behavior (in response to each other). In those circumstances, the theoretical underpinnings of Q-learning (e.g. convergence towards an optimal policy) no longer apply. Put differently, naïve Q-learning simply does not apply. It breaks down.

Several alternative technologies have been proposed to overcome the single-agent limitation of the (naïve) Q-learning algorithm. We discuss hereafter the two most popular multi-agent solutions, namely MinMax Q-learning and Nash Q-learning. We give more exposure to Nash Q-learning as it can be considered as a more general case of MinMax Q-learning. Furthermore, even though these algorithms can handle any number n of agents, we discuss a duopoly scenario where $n=2$ for ease of understanding. This choice is also apposite, because by selecting an environment that is most endogenously conducive to tacit collusion, we give maximum credence to the tacit collusion conjecture, and therefore cannot be accused of a bias against it.

IV. Multi-Agent Q-Learning

[18] Gosavi and Bandla, A Reinforcement Learnign Approach _____.
[19] Hu, Junling, and Michael P. Wellman. "Nash Q-learning for general-sum stochastic games." *Journal of machine learning research* 4.Nov (2003): 1039-1069.

Two variants of Q-learning agents consider multi agents settings and introduce the theoretical possibility of interaction and interdependence necessary to generate tacit collusion.We discuss them in turn.

1. Zero-Sum Games and MinMax Q-Learning

Zero-sum games are an instance of multi-agent environments. The peculiarity of such games is that one agent's gains correspond to the other agents' losses and vice-versa. The agents have strictly "*opposing*" interests.

Consider the case of a 2-agent (player) zero-sum game. Since these 2 agents have exactly the opposite interests (rewards), their respective Q-matrices (c.f. naïve Q-learning procedure presented earlier), , are related, such that:

$$M^1 = -M^2$$

Thus, the game can simplified by considering either M^1 or M^2. The MinMax[20] Q-learning algorithm exploits this relationship: If an agent knows its Q-matrix, it can also deduce that of its opponents. Consequently, it can minimize the rewards of the other agents' rewards, and choose its own optimal action after that.

Formally, MinMax Q-learning can be expressed as shown below (we omitted some indices to simplify the equation compare the naïve Q-learning)

$$Q(s,a,o) = (1-\alpha)Q(s,a,o) + \alpha(r + \gamma V(s'))$$

such that

$$V(S) = \max_{\pi_S} \min_{o} \sum_{a} Q(s,a,o)\pi_S(a)$$

where $\pi_S(a)$ gives the probability of following the agent performing action a.

For more details, we refer the reader to the work of Littman[18]

MinMax Q-learning performs better than the naïve Q-learning in multi-agent environments. However, MinMax Q-learning can only be applied to support decision-making in situations where the agents pursue opposite goals. Thus, their implementation in dynamic pricing applications on oligopoly markets is highly unlikely. In such settings, algorithmic pricing agents (for instance, pricing bots from rival e-commerce companies) all pursue a same goal of profit maximization. Given that

[20] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Eleventh International Conference on Machine Learning*, pages 157–163. New Brunswick, 1994

the basic tenet of MinMax Q-learning is that the agents pursue different goals, it is hard to envisage that they can have ability and/or incentives to engage into collusion, tacit or otherwise.

2. General-Sum Games and NashQ-Learning

General-sum games can be considered as a more generic type of zero-sum game. They are less restrictive than zero-sum games – the latter constraining the agents to pursue opposite rewards – and therefore of higher relevance to the study of the tacit collusion conjecture on oligopoly markets.

In this variant, the agents' respective matrices, and are unrelated. Recall that in zero-sum games, these matrices would be "*opposite*" (negative) of each other, and thus, would sum up to zero.[21] Moreover, in general-sum games, agents are unaware of each other's payoff function and state-transition probabilities.

A common solution in general-sum games is that agents reach a Nash equilibrium:[22] each agent's action is the best response to other agents' actions, and no agent can achieve a better result by deviating (we refer to such agents as Nash Q-leaning agents).

In light of this, it cannot be excluded that Nash Q-learning agents could potentially enter into collusion. For instance, profit-maximizing Nash Q-learning agents could set their prices in response to each other until a point is reached where no agent has any incentive to deviate from this price level given what it expects others to do (the condition of Nash equilibrium is reached). And there is theoretical way to define a Nash Q-value which corresponds to the expected sum of (discounted) rewards when all agents follow specific Nash equilibrium strategies.[23]

We can briefly illustrate the operation of a Nash Q-learning pricing agents, considering n=2 agents for simplicity. As will be seen, this simple example nonetheless highlights some of the key challenges with Nash Q-learning for multi-agents, general-sum games environment.

To start, let us recall that (i) in the case of single-agent (naïve) Q-learning, we only had one agent, and, thus, only one payoffs matrix, with each entry indicating the payoffs for different state-action pairs; and (ii) in the case of multi-agent zero-sum games, the agents' payoffs matrices were strictly the opposite of each other, and summed to zero (in such games one agent's gains is exactly the other agents' losses).

[21] Hu, Junling, and Michael P. Wellman. "Nash Q-learning for general-sum stochastic games." *Journal of machine learning research* 4.Nov (2003): 1039-1069.

[22] Busoniu, Lucian, Robert Babuska, and Bart De Schutter. "A comprehensive survey of multiagent reinforcement learning." *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*(2008).

[23] Hu, Junling, and Michael P. Wellman. "Multiagent reinforcement learning: theoretical framework and an algorithm." *ICML*. Vol. 98. 1998. This contrasts with the single-agent, naïve Q-learning scenario, whereby rewards depend only on the agent's individual strategy. For the multi-agent (general-sum) case, the naïve Q-learning algorithm has to be extended by incorporating the joint actions of the other agents in the environment.

The situation is different for general sum-games. Here, the agents' payoffs matrices are unrelated to each other. We therefore need to explicitly represent information about the other agents.

For a given agent k, $k=1,2$ in our 2-agents setting, let the Q-matrix for state s be $Q^k(s)$ with

- Rows: a^1 A^1; actions of agent 1
- Columns: a^2 A^2; actions of agent 2
- Entries: $Q^k(s,a^1, a^2)$; payoffs of agent k in state s, considering joint agent actions

For instance, in a given state , the Q-matrices, for each agent can be:

$$M^1 = \begin{array}{c} \\ a_1^1 \\ a_2^1 \end{array} \begin{array}{c} \begin{array}{ccc} a_1^2 & a_2^2 & a_3^2 \end{array} \\ \begin{pmatrix} 1 & -2 & 4 \\ 0 & 1 & 1 \end{pmatrix} \end{array} \qquad M^2 = \begin{array}{c} \\ a_1^1 \\ a_2^1 \end{array} \begin{array}{c} \begin{array}{ccc} a_1^2 & a_2^2 & a_3^2 \end{array} \\ \begin{pmatrix} 2 & 1 & 0 \\ 0 & -3 & 2 \end{pmatrix} \end{array}$$

We see here that the payoffs for agent 1 performing action 1 when agent 2 performs action 3 =4 (top right cell of M^1)

Suppose that agent 1 updates its Q-values according to

$$Q^1(s, a^1, a^2) = (1 - \alpha)Q^1(s, a^1, a^2) + \alpha[r^1 + \beta\pi^1(s')Q^1(s')\pi^2(s')]$$

Where $\pi^1(s'), \pi^2(s')$ is a Nash equilibrium.

However, in order to determine $\pi^2(s')$, agent 1 has to learn about $Q^2(s')$. This implies that agent 1 has to maintain another Q-matrix, containing the Q-values for the agent 2, assuming that agents can observe each other's rewards and last actions. For an n-agent environment, each agent has to maintain n such Q-matrices

The upshot of this is that total space requirement is nmA^n; exponential in the number of agents and perhaps even infinite. This space requirement is one of the main limitations of multi-agent Q-learning, hindering its deployment in real-life, complex situations. Some of the other important limitations will be discussed next.

V. Limitations of Multi-Agent Q-Learning and the Tacit Collusion Conjecture

This rudimentary background suffices to pinpoint several challenges[24][25][26] that must be overcome before smart pricing agents which rely on sophisticated RL mechanisms can adapt to each other's action, and potentially enter into tacit collusion.

1. Challenge 1: Preference Specification

At the design stage, it is doubtful that RL pricing agents that follow a standard profit maximization function can reach a tacit collusion equilibrium. In multi-agent Q-learning, each agent's rewards are related to each other, and cannot be optimized independently. Joint actions and rewards should be considered. And the ability of each oligopolists to specify such joint actions and rewards for its own RL pricing agent payoffs matrix necessitates access to internal information on competitors that is in principle private.

Of course one can argue that collusion does not require knowledge of other firms' payoff functions but rather other firms' strategies. In this respect, it may be is enough to know how other firms will react. Moreover, companies may be able to rely on relatively simple market monitoring tools (in the form of spiders or bots) to spy on their competitors prices. For instance, in supposedly simple markets like airlines, where firms enjoy stable positions, little entry, mature technology and similar input costs, such payoffs matrix may be easy to draw. At the same time, it is totally unclear that firms can understand each others strategies when personalised and dynamic pricing are combined, even when they resort to data monitoring software. This is because the range of price points over which oligopolists must coordinate is virtually infinite, and is a function of the number of individual customers times the number of time units spent on digital markets. Moreover, the use of data mining technology does not eliminate the need to know the underlying business strategy. Observing that your rival launches an aggressive sales promotion does not tell you if he is on the verge of bankrupcy or if instead he feels strong enough to attempt to bankrupt its rivals. Put differently, data mining does not elicit patterns and knowledge that can be trusted automatically without verification.[27] Lastly, the monitoring of rivals' prices may be a plausible tactic on B2C markets, but much less on B2B markets where information is not public.

A possible solution could involve the specification of Nash equilibrium as a preference function: the RL pricing agent should strive to attain the situation where it is not better off not deviating from its current state given what it expects other RL agents to do. However, this specification is alone insufficient to give rise to tacit collusion, for many ambient and environmental factors will affect the stability of a collusive Nash equilibrium, that are beyond the control of the RL pricing agent. Additional

[24] Busoniu, Lucian, Robert Babuska, and Bart De Schutter. "A comprehensive survey of multiagent reinforcement learning." *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*(2008).
[25] Tesauro, Gerald. "Extending Q-learning to general adaptive multi-agent systems." Advances in neural information processing systems. 2004.
[26] Hu, Junling, and Michael P. Wellman. "Nash Q-learning for general-sum stochastic games." Journal of machine learning research 4.Nov (2003): 1039-1069
[27] C. Rygielski et al., Data mining techniques for customer relationship management, Technology in Society 24 (2002) 483–502.

mechanisms on top of the RL pricing agents would be needed to ensure convergence, like contacts between oligopolists to exchange information.

Last, the RL preference function should be defined so as to ensure the stability of the learning mechanism of agents, as well as their adaption to other agents' dynamic behavior.

2. Challenge 2: Formalizing the Environment and the Data Problem

As was just hinted, the specification of an appropriate preference function is insufficient, in itself, to lead multi-agent Q-Learning systems to tacitly collude. Additional steps must be taken to feed the Q-Learning pricing agent with data.

First, payoffs matrices must be defined that provide enough information on the RL pricing agent environment, including on its competitive environment. If the rewards of other RL pricing agents are not observable – which they should not in a competitive market without collusion in its initial state – the construction of a payoffs matrix will be complicated. True that it remains possible to use publicly available information as a proxy, yet this information will be far from perfect. For instance, a vendor like an airlines company can define a script that mimicks an online user, and who then can view the airfares of rival airlines. But this information gives just access to pricing data, which is just one parameter in the revenue equation (which includes costs, and other factors). Moreover, this type of data will only be observable in B2C markets, and less likely to be available in B2B markets.

Second, if we model the RL pricing agent environment as a stagewise game, one needs to define what exactly constitutes a stage and how long are the periods within stages. Each oligopolist may have a different view on this, giving rise to heterogeneity in RL pricing agents decisions.

Last, other economic variables may have to be taken into account for the calculation of payoffs, including demand forecasts, countervailing buyer power (including vertical integration strategies), input prices, etc. This adds another pinch of uncertainty and heterogeneity in the performance of multi-agent Q-Learning systems.

3. Challenge 3: Non-stationary Agents and Preference Construction

As mentioned in Challenge 1, a RL pricing agent must keep track of other learning RL pricing agents, resulting in a non-stationary environment. Agents are confronted with a trying to "*shoot a moving target*" problem. Each agent's optimal actions changes as a result of the policies (actions) other agents. Non-stationarity can potentially invalidate the convergence properties of Q-learning agents.

To fix this problem, a possibility consists in specifying that RL pricing agents should converge towards a Nash equilibrium. Besides the fact that this would necessitate oligopolists to agree on algorithmic preference specification – which would be unlawful –under the so-called folk theorem, several Nash equilibria can exist, including some in

which prices do not converge due to preference construction, time preferences, and discount factors.[28]

4. Challenge 4: Scalability

Scalability is a major concern for Q-learning algorithms. As previously mentioned, the total space requirement of an environment with *n* agents, A possible actions (for each agent), and m states is nmA^n, which is exponential in the number of agents.

Most of the algorithms' run-time is taken up by calculating Nash equilibrium for updating the Q-function. For matrix games, the computational complexity of finding an equilibrium is unknown, and approximate methods are often used in *n*-player games .

5. Challenge 5: Exploration v Exploitation

Another challenge concerns the exploration v exploitation trade-off. The exploitation choice consists in selecting the action with the maximum Q-value for each state. The exploration choice consists in investigating to improve current knowledge (for instance by randomly selecting actions with a probability ε). The exploration v exploitation decision confronts Q-learning agents with a trade-off between exploiting their current knowledge and exploring to improve the current knowledge. In a multi-agent setting, this is challenging, because an RL Q-Learning agent focusing too much on exploration will disclose insufficient pricing information, and in turn destabilize the learning mechanism of other RL Q-Learning agent willing to explore.

A related issue arises if an agent faces a situation of tie-break. This happens when the agent is faced with two actions that will yield two rewards of the same value. It will have difficulty in determining which action to take (to break the tie).

Conclusion

In this paper, we have stressed that the algorithmic tacit collusion conjecture should not presently be taken as a given. Significant technological challenges exist that undermine the capabilities of Q-learning algorithms to approach a tacit collusion equilibrium.

That said, the introduction of Deep RL agents (like Deep Q-Networks) on markets may alleviate some of the obstacles to tacit collusion that we have identified. In particular, Deep RL agents may be quite effective at learning the Q-values of rival oligopolists. To date, Deep RL agents have mostly been used in the context of game AI (for example, playing Atari 2600 games v humans).[29] And they have proven able to outperform humans. With this background, it remains to be seen whether they can actually replicate or surpass human conduct and eventually enter more easily into tacit collusion. At the time being, this is still an open research question.

[28] The folk theorem asserts that any individually rational outcome can arise as a Nash equilibrium in infinitely repeated games with sufficiently little discounting. See, for its original formulation, Fudenberg, Drew, and Eric Maskin. "The folk theorem in repeated games with discounting or with incomplete information." *Econometrica: Journal of the Econometric Society* (1986): 533-554.

[29] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.