COMBATING ANTI-COMPETITIVE BEHAVIOR INVOLVING ALGORITHMS: PLATFORM DESIGN AND ORGANIZATIONAL PROCESS

BY JUSTIN P. JOHNSON, ANDREW RHODES & MATTHIJS WILDENBEEST[1]



1 Cornell University, justin.johnson@cornell.edu; Toulouse School of Economics, andrew.rhodes@tse-fr.eu; Indiana University, mwildenb@indiana.edu.

CPI Antitrust Chronicle July 2020

www.competitionpolicyinternational.com

I. INTRODUCTION

In this article we examine potential anti-competitive effects associated with algorithms and what can be done about them.[2] We take the perspective not only of antitrust authorities but also of firms that wish to avoid being either the victim or unwitting accomplice of other firms' anti-competitive schemes.

Our analysis is structured as follows. In Section 2 we describe how algorithms can be deployed to more effectively implement classic anti-competitive schemes. We explain in detail why antitrust authorities not only need access to suspect algorithms but more importantly need to understand details of the organizational design process behind the algorithm. We also propose that online retail platforms can use their own algorithms to fight anti-competitive schemes perpetrated on their platforms. In Section 3 we describe ways in which "benign" algorithms (those designed with no anti-competitive intent) can be exploited by bad actors to achieve anti-competitive ends. Section 4 closes by discussing potential remedies to this problem.

II. ALGORITHMS AND IMPROVED IMPLEMENTATION OF ANTI-COMPETITIVE CONDUCT

Algorithms enable firms to rapidly collect and analyze data, and then quickly act upon the results of that analysis. Algorithms may therefore make it easier to implement a variety of anti-competitive practices. For example, it has been widely argued that algorithms may facilitate price collusion (Ezrachi & Stucke (2017)),[3] because they enable firms to monitor and quickly enforce cartel agreements. Evidence from algorithms simulated in the lab bears out this concern (Calvano et al. (2019), Klein (2019)).[4]

Algorithms can also support many other potentially anti-competitive practices. Consider vertical restrictions such as price parity clauses ("PPCs"), minimum advertised prices, and bans on selling in certain countries or through certain sales channels. Such restrictions could be specified in contracts, but could also be implemented informally. Algorithms make it easier for firms to check and enforce compliance with these restrictions. For example, an online travel agency could use algorithms to scan hotel websites for violations of PPCs, and then display non-compliant hotels less prominently on its website.[5] Hotel algorithms might thus learn over time to comply with the PPC, even absent a formal contract.

2 Although we examine potential anti-competitive effects of algorithms, we note that algorithms can certainly serve many pro-competitive ends, for example by better equilibrating supply and demand.

3 Ezrachi, A., & M.E. Stucke. 2017. "Artificial Intelligence and Collusion: When Computers Inhibit Competition." Illinois Law Review 2017:1785–1810.

4 Calvano, E., G. Calzolari, V. Denicolò, & S. Pastorello. 2019. "Artificial Intelligence, Algorithmic Pricing and Collusion." Working paper. Klein, T. 2019. "Autonomous Algorithmic Collusion: Q-Learning under Sequential Pricing." Working paper.

5 We emphasize that the vertical restraints under discussion need not be anti-competitive, in which case improved compliance could bring additional benefits.

Algorithms could also make predation easier. Algorithms may enable firms to identify new rivals and automatically take actions which reduce their profitability. Aside from aggressive pricing, various non-price actions could be implemented such as over-bidding on rivals' search keywords or manipulating product reviews.

We now discuss what can be done when algorithms are used for anti-competitive goals.

A. Understanding the Organizational Process Behind an Algorithm

If antitrust authorities suspect that an algorithm is perpetrating an anti-competitive scheme, a natural instinct is to gain access to the algorithm itself. Although sensible, in many (perhaps most) cases, mere inspection of the algorithm might not be very revealing or useful in court. The reason is that it seems unlikely (to us) that an algorithm's underlying code will have something that reads like "engage in anti-competitive conduct." Algorithms are extremely complex and their exact operation may depend on how they were trained, and so inspecting the algorithm might not provide clear and convincing evidence of wrong doing. Rather, it is more likely that a firm seeking to engage in bad conduct would instead tweak the design or training of an existing complex algorithm in a way that nudges the algorithm towards the desired conduct. These tweaks may be difficult to detect or otherwise separate from legitimate design choices, and may in any event not constitute strong evidence on their own of anti-competitive intent.

Therefore, we instead suggest that authorities focus on the human and organizational aspects of algorithm design rather than just the algorithm itself. Some leading questions that authorities should ask include the following. When were important design elements introduced or changed, and what was the business context? What were internal deliberations surrounding any design decisions, not just at the management level but also at the engineering or programming level? When was the training halted, and why? Was the algorithm later retrained on different data, and if so why? Such historical analysis of the algorithm's evolution may reveal much more about the intentions of the organization and consequently shed more light on whether anti-competitive conduct occurred or was attempted.

To better understand why we feel our simple questions regarding algorithm design are important (and perhaps more insightful than studying the algorithms themselves), consider first the hypothetical example of a platform seeking to enforce a vertical restraint such as a price parity restriction. As discussed earlier, one way of enforcing such a restriction is through the platform's ranking algorithm. This might be accomplished in a subtle fashion (especially if such restrictions are illegal). For instance, rather than directly scanning rival platforms and downranking sellers that set lower prices on those platforms, the platform might already have a good sense of which firms are violating a restriction and then enforce the restriction by slightly altering its complex algorithm in a way that "just happens" to penalize the offending firms.

An authority might ask, for example, did the platform's change of its algorithm coincide with the emergence of a rival platform or following changes in the pricing behavior of certain hotels? Did the platform experiment with multiple versions of its ranking algorithm, and if so what drove the final selection process? Even if some details of the algorithm are opaque or justifiable for legitimate business purposes in the abstract, information about the organizational process that influenced its design may be extremely revealing and difficult to justify (assuming an offense has indeed occurred).

Now consider a real-world example: Amazon's product-ranking algorithm. One concern might be whether Amazon is anti-competitively showing a preference for its own products (this is separate from any price-parity issues). According to a recent article in the Wall Street Journal, this algorithm assigns weights to over one hundred variables as part of its procedure for determining how products are displayed to consumers.[6] That article highlights internal discussions at Amazon focused on whether the algorithm should factor in a measure of variable profitability that might benefit Amazon's own product rankings. These discussions occurred as Amazon's private-label business was expanding. That same article hints at even more subtle ways in which an algorithm might be tweaked to exhibit self-preferencing. For example, to achieve the goal of self-preferencing its own products, it may be that Amazon need neither directly uprank its own products nor even include variable profitability measures; it may be enough to find and include variables that are correlated with measures of variable profitability (and given the expansive data possessed by Amazon, this may be quite feasible).

6 See "Amazon Changed Search Algorithm in Ways that Boost its Own Products," by Dana Mattioli, September 16, 2019.

We emphasize that we are not asserting anything untoward in our example involving Amazon, or that it self-preferences (indeed, even if it does, many companies regularly self-preference their own products for display and we are not suggesting that such tactics need be anti-competitive in general). Rather, we are emphasizing the many subtle design aspects of algorithms, that these designs may change at critical historical junctures, and that revealing internal discussions may take place. Focusing on these human aspects and the business contexts surrounding historical design decisions and changes in the design of algorithms may well prove to be more valuable to authorities than merely inspecting the algorithms themselves.

B. Using Algorithms to Directly Fight Anti-Competitive Behavior

Another intriguing possibility is that algorithms might be recruited as a force for good in the fight against anti-competitive conduct by either other algorithms or humans. Earlier work has suggested, for example, that algorithms might be used to detect or screen for anti-competitive behavior in a variety of contexts (see Johnson & Sokol (2020) for a discussion of how algorithms might be used to detect collusion in online marketplaces, especially collusion over non-price variables such as manipulation of product reviews).[7]

Here we discuss how algorithms can be used even more directly as part of an offensive against collusion. To set ideas, suppose that sellers are colluding on an online marketplace or platform (perhaps using algorithms to do so), and that the platform would prefer that sellers not collude.[8]

As emphasized and analyzed formally in our own study of algorithmic collusion and the duties that platforms have to police their marketplaces (Johnson et al. (2020)), an online platform has tools at its disposal to fight against collusive behavior. In particular, such platforms have the ability to influence the consumer search process using their own algorithms.[9] Although this power is often viewed in a negative light by those concerned about abuses by platforms, the same power can be used to strike a blow against collusion.

The idea is simple: when a platform guides consumer search it can substantially lower the ranking of firms with higher prices and instead promote firms setting lower prices.[10] We call this technique for steering demand price-directed prominence and show that it and related techniques may substantially destabilize a cartel and lower prices (even if sellers are using algorithms to implement a collusive scheme). The intuition is as follows. Collusion becomes harder to sustain when individual firms expect higher profits from abandoning the cartel. Because price-directed prominence steers additional demand and hence profits to firms that cut prices, this technique naturally makes it harder for collusion to exist.

There are two important conclusions. First, algorithms can be used to fight specific anti-competitive conduct, in this case collusion. Thus, platforms can use their powers actively to promote pro-competitive outcomes by designing their own marketplace algorithms to alter the attractiveness or feasibility of anti-competitive schemes. Second, we find some support for the idea that platforms can improve competitive outcomes by treating sellers in a non-neutral or asymmetric fashion.

It may be surprising that a platform behaving in a non-neutral fashion may be promoting pro-competitive outcomes. To understand why this non-neutrality may benefit consumers, we briefly discuss an extension of the basic idea highlighted above which we call dynamic price-directed prominence. This extended technique works by rewarding firms that priced low in the recent past by steering demand to them today even if other sellers somewhat undercut their price. That means that today the platform is treating sellers in a non-neutral fashion and (seemingly) rewarding higher prices. But the trick is that a seller can only achieve this privileged position if it cut prices in the past, and we show that this provides additional incentives for sellers to deviate from cartels, lowering prices and benefiting consumers. The reason this technique works so well is that it makes attempts by cartel members to punish the deviating firm (by undercutting its price subsequent to the deviation) less effective by giving the deviating firm a (small) cushion against such punishments. By insulating firms against cartel punishments — but only when they have first cut prices — the stability of a cartel is further undermined, lowering prices.

7 Johnson, J., & D.D. Sokol. Forthcoming. "Understanding AI Collusion and Compliance," Cambridge Handbook of Compliance.

8 A platform may put consumers first if it is focused on increasing its market share, or faces tough competition. In other circumstances a platform might prefer higher prices, but such is not our focus.

9 Johnson, J., A. Rhodes, & M. Wildenbeest. 2020. "Platform Design when Sellers Use Pricing Algorithms." Working paper.

10 Indeed, the Amazon Marketplace already uses this idea in its "buy box." The buy box works for products that are completely undifferentiated and provided by multiple sellers (or resellers). A single firm is displayed in a special and highly prominent box and purportedly accounts for up to 80% of sales within any given category. The exact process used to select which firm is shown is not publicly known but having a lower price is believed to improve the odds of being selected.

III. EXPLOITING WEAKNESSES IN ALGORITHMS FOR ANTI-COMPETITIVE ENDS

Above we focused on how algorithms may automate the ability to collect and analyze data and respond to that analysis in a way that improves the ability to implement classic anti-competitive schemes. In this sense, algorithms can be seen as "more perfect" human operators. Here we take a different perspective and discuss how bad actors might exploit weaknesses in the algorithms of other, non-offending firms in an attempt to achieve their own anti-competitive ends.

How an algorithm works depends on how it is trained. A frequent criticism in the machine-learning community is that trained algorithms may exhibit "over fitting": the algorithm performs very well in predicting outcomes or optimal actions when current data looks like past data, but sometimes very poorly when faced with "out-of-sample" data. This problem may be exacerbated when the algorithms work in a model-free manner (as many do), that is when they are not designed from a first-principles approach based on details of the underlying economic environment and the strategies that firms use.

To see how this may be troublesome for competition, consider a hypothetical online marketplace that uses an algorithm to determine the rankings of various products that are displayed to potential consumers (and suppose further that the platform has no desire to engage in anti-competitive conduct). It may be that training data suggests that consumers tend to prefer well-established brands in a particular category and so the algorithm may actively promote such products over smaller, less-established brands. This can be problematic.

One reason is that data from the historical training period may not be representative of the current economic environment; innovation rates and consumer preferences may well fluctuate over time, perhaps especially in digital markets. This means that what might have been a sensible historical strategy (of not heavily promoting smaller brands) might be a biased strategy today. Second, even if the platform would ideally not wish to promote smaller brands in a given category, problems could again arise if the algorithm were deployed to different categories, as may well occur due to data limitations.

Additionally, an algorithm's performance is sensitive to the underlying objective function that a human designer specifies. This is particularly relevant for unsupervised training processes or algorithms that use reinforcement learning techniques to attempt to find optimal decision-making rules. For example, one objective function a platform might assign to an algorithm is the maximization of profits. Although that sounds simple, the reality is more complicated because of the difference between short-run profits and long-run profits. Measuring long-run profits — and accurately assessing how actions today impact long-run profits — may be very challenging, leading algorithm designers to focus more on short-run profits.

But a new product may take time and exposure to be successful. Enough consumers need to try it and leave positive reviews, reviews which may also describe dimensions along which the new product is superior to other products. Consequently, in the short run promoting a new product may be bad for profits even though it is sensible in the long run. In the reinforcement learning literature from computer science this is known as the problem of "delayed rewards" (Sutton & Barto (2018)).[11]

From the standpoint of threatening competition, the weaknesses of algorithms just described are gravest when other players seek to exploit them for their own anti-competitive goals. For instance, consider an incumbent supplier interested in stifling nascent competition. This manufacturer (or its algorithmic agent!) may understand that a new product represents a potential future threat and furthermore understand how to game the platform's ranking algorithm so as to limit this threat.

The incumbent might engage in temporary tactics to attempt to convince the ranking algorithm not to prominently display the entrant's product. The incumbent might lower its price, not only limiting the early sales of the entrant but furthermore convincing the algorithm that the entrant's product is not in high demand, thereby causing the algorithm to not promote the product in the future. Or the incumbent may be aware of other details of how the algorithm works. For instance, perhaps the algorithm gives a ranking boost to products that have high inventories (assuming the platform handles fulfillment), which the incumbent could game by temporarily boosting its own inventory levels. Especially because these tactics may only be deployed occasionally, they may be successful and the platform may be an unwitting accomplice to anti-competitive conduct. In some cases, a bad actor might even strive to manipulate the training of the algorithm—imagine if an incumbent could convince the platform's algorithm that small brands don't sell well, in effect unfairly creating the ranking bias against such products discussed above.

11 Sutton, R.S. & A.G. Barto. 2018. Reinforcement Learning: An Introduction. MIT Press.

One question is which firms are better able to identify and engage in the types of tactics described above. One possibility is that larger technologically savvy firms are more likely to invest in the required data acquisition and analysis and algorithmic development.

Regardless of the identity of the offender, some of these tactics might be transparent to a human observer familiar with the details of the market, underscoring the risks of leaning too heavily on algorithms that have no inherent understanding of the underlying economic and strategic environment. Such an observer might understand that incumbents have different incentives when faced with new entry, and understand the basic economics of predation, consequently being alert or attuned to bad behavior by certain incumbents.

The lack of human oversight can lead to an additional anti-competitive risk associated with the (mis)training and possible exploitation of algorithms, which is that the affected parties might never realize that anti-competitive conduct occurred. A victim that doesn't understand whether or how it was harmed is presumably much less likely to raise a red flag and further investigate or seek legal remedy. Thus, it may become easier for offending parties to avoid legal risks, especially as more affected parties lean on algorithms which lack human understanding of the underlying economic environment.

IV. RESPONDING TO PROBLEMS IN THE TRAINING OR EXPLOITATION OF ALGORITHMS

Here we describe some high-level steps that firms can take to limit the anti-competitive harm from bad actors trying to exploit weaknesses in their algorithms.

First, an algorithm's designer may decide that the functioning of the algorithm should not be fully transparent to all market participants. For example, returning to our earlier predation discussion, a seller seeking to game the ranking algorithm and thus prey upon a new rival will find doing so more difficult if some details of the platform's ranking algorithm process are obscure. There are several ways that an algorithm's functioning can be kept obscure and thus resistant to abuse.[12] One way is to frequently update the details of the algorithm, or to add noise to its output. Another way is to limit how often the algorithm's output adjusts to inputs provided by market participants. For example, a platform may not change product rankings every time a seller updates their price, keyword, or inventory level. In doing this, the platform may limit a seller's ability to understand the exact rules of the algorithm.

Second, humans should be kept partially in the decision-making loop. A major advantage of algorithms in the first place is that they automate decisions, and so oversight needs to be selective. The circumstances where oversight is more important depends on the underlying economic environment. Returning to the example of an online marketplace designing its ranking algorithm, a human overseer might recognize that periods of market entry are different from other periods. Such an overseer might examine how incumbent firms respond to entrants and simply ask what role their own algorithms are playing. For example, such an overseer might discover that sudden price cuts following entry are followed by equal price increases if the new entrant fails, and question whether the algorithm unwisely failed to promote the new product.

If the set of likely anti-competitive actions is fairly small and easy to recognize, it may even be possible to train the algorithms to discover such conduct and respond appropriately. If feasible, this is clearly an attractive option as it allows algorithms to be broadly applied across many markets while making anti-competitive conduct less attractive. Or, using human insights as to what anti-competitive conduct might look like, the platform might use algorithms to monitor activity across many markets and further investigate those which exhibit warning signs.

Even if designing the perfect algorithm is not feasible, human awareness of how algorithms are being used is important for alerting actors to possible abuses by other parties, allowing them to seek legal recourse.

Finally, it must be kept in mind that anti-competitive strategies such as predation or exclusion involve long-run payoffs, with potentially short-run sacrifices by the offending firms. This means that focusing on short-run payoffs can lead to bad outcomes. Despite the difficulties in accurately measuring the long-run effect on profits from an action today, merely abandoning that goal is not sensible. Once again, human insight can help guide algorithmic development based on an understanding of the economics, for example by adopting policies that ensure some level of promotion for new products to give them opportunities to grow. In addition, there are also tools from the computer science literature specifically geared towards learning the optimal policy in environments with long-delayed rewards, for example reinforcement learning (especially when

12 In other cases, the algorithm's designer may prefer transparency. For example, the price-directed prominence technique discussed earlier will be most effective when sellers understand that low prices translate into more prominent display.

combined with “eligibility traces” (Sutton & Barto (2018))).[13] Designers of algorithms can implement these but first they need to know whether they are operating in an environment where such delayed rewards are an important element of competition.

13 Sutton, R.S. & A.G. Barto. 2018. Reinforcement Learning: An Introduction. MIT Press.