Algorithmic Collusion in Multi-Product Pricing

Karsten Hansen, Kohei Hayashida, Kanishka Misra, and Mallesh Pai

Rady School of Management, UC San Diego

February 7, 2025

Use of AI methods is ubiquitous in pricing

- Pricing decisions are being automated
  - Real-time supply and demand shocks
  - Price discrimination
  - **Demand learning**
- Demand learning: pricing with unknown demand curves
  - Reinforcement learning: learn and earn

Literature we add to

Market outcomes with algorithmic sellers

TECHNOLOGY

Online pricing algorithms are gaming the system, and could mean you pay more

JULY 25, 2022 · 5:00 AM ET

kpbs

NEWSLETTERS SIGN IN NPR SHOP

NEWS CULT

THE WALL STREET JOURNAL.

Latest World **Business** U.S. Politics Economy Tech Markets & Finance Opinion Arts Lifestyle

BUSINESS | JOURNAL REPORTS: TECHNOLOGY

Beware Algorithms That Could Collude on Prices

In a study, two pricing algorithms learned on their own to raise prices together to unfairly high levels npr kpbs

NEWSLETTERS SIGN IN NPR SHOP

THE WALL STREET JOURNAL.

NEWS CULT

Latest World **Business** U.S. Politics Economy Tech Markets & Finance Opinion Arts Lifestyle

BUSINESS |

Bew

on P

In a stuc

togethe

Technology

Rent Going Up? One Company's Algorithm Could Be Why.

by Heather Vogell, ProPublica, with data analysis by Haru Coryne, ProPublica, and Ryan Little

Oct. 15, 2022, 5 a.m. EDT


ALGORITHMS AND COLLUSION
Competition policy in the digital age
npr
kpbs
NEWS
CULT
Latest
World
Bu
BUSINES
Bev
on
In a st
toget
HOP
REET JOURNAL.
Opinion
Arts
Lifestyle
One
orithm Could
is by Haru Coryne, ProPublica, and
OECD

Algorithmic price competition: **the fear of AI**

- *"Robo-sellers will increase the risk that oligopolists will coordinate prices above the competitive level"* Mehra (2015)
- Why is this non-trivial?
  - Algorithms: written and analyzed in stationary environments
  - Algorithmic competition: environment is endogenous/nonstationary

Algorithmic collusion ≡ Market prices are supra-competitive

- Can *independent* algorithms collude?
- Should policymakers be concerned?

Algorithmic price competition: **the fear of AI**

Can algorithms coordinate prices above the competitive levels?

- Yes! *independent algorithms* can lead to supra-competitive prices
  - Mechanisms in simulated markets: *facilitate repeated games* (Calvano et al 2020, Kline 2021), *correlated learning* (Hansen et al 2021), *timing* (Mackay and Brown 2021), *sophistication* (Asker et al 2021), *hub and spoke* (Harrington 2021)
  - Empirical: German gasoline markets (Assad et al 2023), Multifamily rentals (Calder-Wang and Kim 2024), E-commerce (Musolff 2024)
  - Theory: Prisoner's dilemma (Banchio Mantegazza 2022)

Current research

- **When** *independent* algorithms collude?
- **Markets** for policymakers to study

In this study, we

- Investigate algorithmic pricing in multi-product sellers
  - Multi-product pricing is high dimensional problem / computationally hard
- Show evidence that multi-product firms use simplified pricing algorithm
  - Ignoring cross price elasticity / product-by-product optimization
- Show algorithmic collusion is less likely to sustain when multi-product firms employ single-product pricing algorithm
  - ... both in theory and simulation
  - If firm could employ sophisticated (multi-product) algorithm, algorithmic collusion returns

Overview of talk

- Current knowledge: market outcomes with single product firms
- Extension: multi-product firms
- Firm behavior: multi-product firms use simplifications
- Implications: outcomes with simplifications
- Alternative simplifications

Multiple agent Q-learning: structure

- Q-learning (as in Calvano et al 2020, Klein 2021, Asker et al 2022)
  - Reinforcement learning with states
  - States: prices of all agents (discrete)
  - Actions: next price to charge (discrete)

- Q-learning setup:
  - Objective (discount factor δ) $E[\sum_{t=0}^{\infty} \delta^t \pi_t]$
  - Q function (Bellman's value function $V(s) = max_a(Q(a,s)))$

$$Q(a,s) = E(\pi|a,s) + \delta E[max_{a'} Q(a',s')|a,s)]$$

  - Iterative learning[1] (learning rate α)

$$Q_{t+1}(a,s) = Q_t(a,s) - \alpha(\pi_t + \delta E[max_{a'} Q(a',s')|a,s)] - Q_t(a,s))$$

  - Experimentation ('off-policy learning'): ϵ-greedy

[1] in single agent problems this default to ϵ-greedy in our simulations

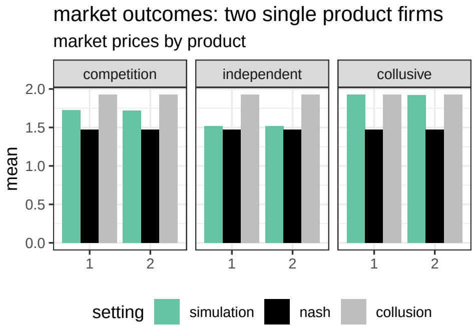
Multiple agent Q-learning: setup

- Multi-agent learning setup
  - Two symmetric single product firms
  - Actions: 15 potential prices
  - States (memory): prices charged in time $t-1$
  - Simulate demand from a logit
- Outcome (steady-state) metrics
  - Prices, profit and consumer surplus (loss)
  - Infer learned mechanism

Market outcomes: single product firms can collude

Replicate literature

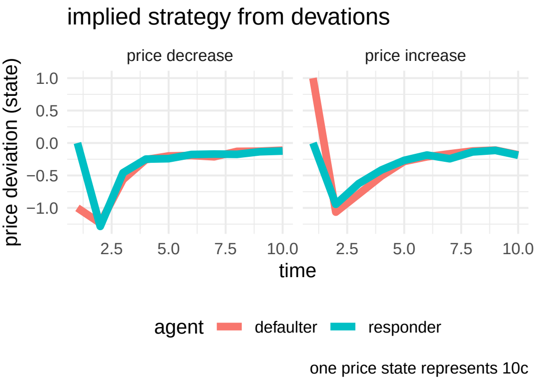
1. Competitive: Q-learning: prices supra-competitive (Calvano et al 2020)
2. Independent: ϵ-greedy: prices Nash (Hansen et al 2021)
3. Collusive: Joint maximization



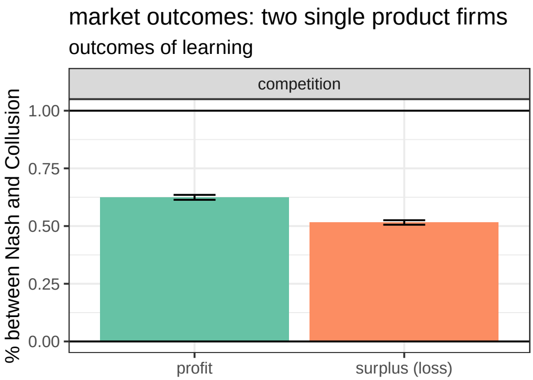
Market outcomes: single product firms learned strategy

Replicate literature

Inferred strategy:



Implications for policy:



Multi-agent Q-learning: theory (Banchio Mantegazza 2022)

Setup

- Prisoner's dilemma:

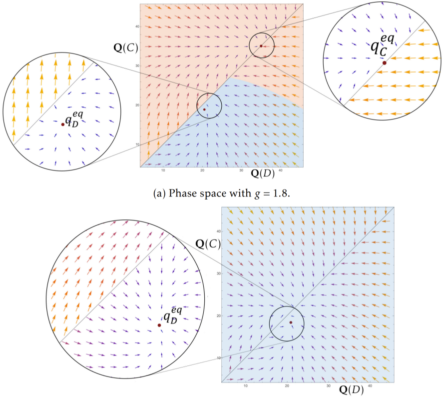
		Firm 2	
		$p_H \equiv 1$	p_L
Firm 1	$p_H \equiv 1$	$1, 1$	$0, 2p_L$
	p_L	$2p_L, 0$	p_L, p_L

- Complexity comes from learning in continuous time
  - Each firm follows Q-learning with experimentation (ϵ)

Market outcomes: single product firms can collude

Replicate Banchio Mantegazza (2022)

Proposition: if $p_L < p^*(\epsilon)$, there exists a collusive pseudo-steady state



Empirical evidence: German gasoline market

- Assad et al (2023): evidence consistent with algorithmic collusion
  - Focused on prices of E-10 gas and markets at postcodes
  - Infer adoption of algorithmic pricing in/around 2017
  - Prices increased in duopoly markets only when **both** adopted

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Firm behavior: multi-product firms use simplifications
4. Implications: outcomes with simplifications
5. Alternative simplifications

Summary

- Current knowledge provides evidence of pricing algorithms achieving supra-competitive outcomes
- Evidence limited to single product algorithms

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
   - Do the results extend to multi-product firms?
   - Are the strategies different?
3. Firm behavior: multi-product firms use simplifications
4. Implications: outcomes with simplifications
5. Alternative simplifications

Simulations: implications for market outcomes

Multi-product firms implications

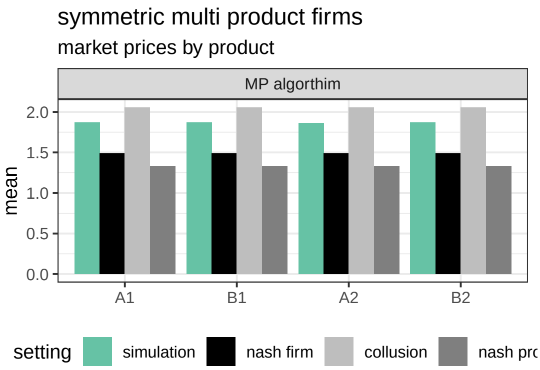
- Multi-product firms using pricing algorithms
  - Each sells two products
    - Firm 1 sells products A1 and B1
    - Firm 2 sells products A2 and B2
  - Algorithm assumes they complete on each product separately
    - A_i price considers $\{A_1, A_2, B_1, B_2\}$ prices
    - B_i price considers $\{A_1, A_2, B_1, B_2\}$ price
  - Demand is shared
    - Consumers pick between all four products (A_1, A_2, B_1, B_2)
    - Assume logit demand as before

- Repeat simulation assuming the firm sells two similar goods
  - Calvano et al. (2022) setup as before
  - Significant complexity: $15^4 = 50,625$ states and $15^2 = 225$ actions

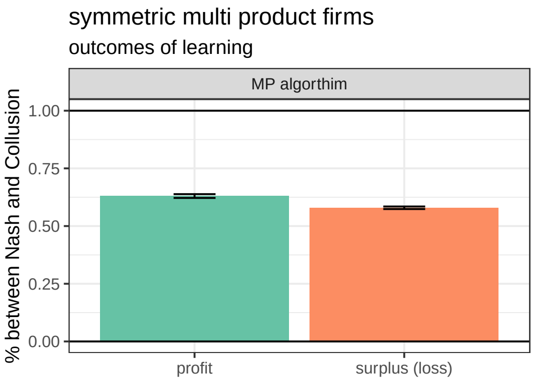
Simulations: implications for market outcomes

Multi-product firms using multi-product algorithms …

… result in **supra-competitive** price



… **can** disadvantage consumers



Simulations: implications for market outcomes

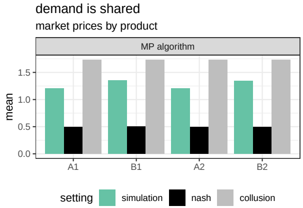
Understanding firms strategy

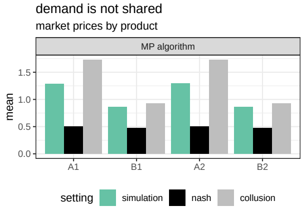
- Simplified model similar to theory model
  - Two (2) prices per product (Nash and Collusive)
  - $|$Action space$|$ is 4 and $|$state space$|$ is 16
  - Will consider two type of demand models:
    - Demand is shared
    - Demand is unrelated (true DGP has zero cross-price elasticity)
- Purpose of this simulation
  - Simple model replicates the complex model
  - Understand the strategies learned

Simulations: simplified model replicates complex model

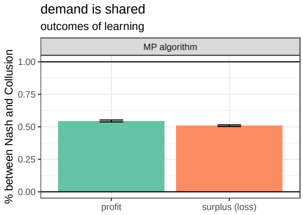
Multi-product firms using multi-product algorithms ...

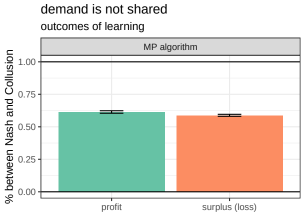
... result in **supra-competitive** price





... **can** disadvantage consumers





Simulations: learned strategy

Understanding firms strategy

- Strategy: Action taken under a given State (prices of all 4 products)
  - e.g. Agent 1's strategy: prices A_1 and B_1 given prices for A_1, B_1, A_2 and B_2

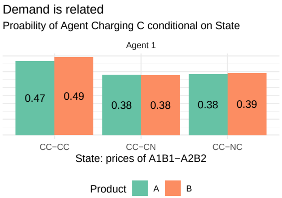
- Will consider two statistics
  1. Response to deviations from other firm reducing price
  2. Regression of Pr(charge C) as function of state

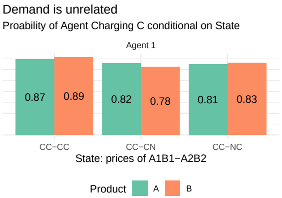
Details

Simulations: simplified model learned strategies

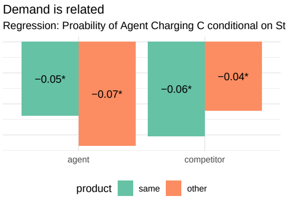
A deviation in one product ...

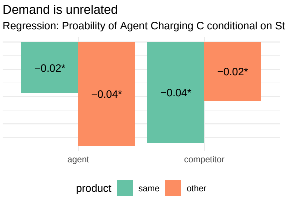
... result in reactions on **both** prices





... Pr(charge C) depends all **four** prices





Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Firm behavior: multi-product firms use simplifications
4. Implications: outcomes with simplifications
5. Alternative simplifications

Firms using multi-product algorithms can reach supra-competitive prices

- Strategy learned consistent with Multi-Market Contact
- Policy makers/researchers: observed prices depend on all products (own and competitive)
- Firms: solving a complex learning problem

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Firm behavior: multi-product firms use simplifications
   - Direct: single product algorithms for marketing decisions
   - Indirect: scraped prices of related goods on amazon.com
4. Implications: outcomes with simplifications
5. Alternative simplifications

Pricing in multi-product firms

- Multi-product pricing is difficult
  - Cross-price elasticity estimates are imprecise (Hitsch et al 2021)
  - Curse of dimensionality: multi-product learning
    - Consider n products with n_p potential prices each: action is a price vector
    - Multi-product pricing: optimize over ${n_p}^n$ feasible price vectors
    - No current tools to solve complexity
- Multi-product firms often use simplifications
  - DellaVigna and Gentzkow (2019) ignore cross-price efforts
  - Compiani and Smith (2022) provide evidence of single product pricing
  - In offline markets: category management practices
- Conjecture: multi-product firms use single-product algorithms
  - Reduce complexity from exponential to multiplicative
    - n single-dimensional optimization problems

Conjecture: multi-product firms use single product pricing

- Academic: exploration with unknown demand
  - Most assume single product (e.g., operations: Besbes and Zeevi 2009, marketing: Misra et al 2019)
  - Multi-product models assume logit demand (see Jain et al 2023)
- We will show the following evidence
  - Pricing patents
  - Not limited to pricing: evidence in advertising markets
  - Observed pricing patterns

Conjecture: multi-product firms use single product pricing

Walmart labs patent US 2019 / 0172082 A1[4]

- 2019 patent: "systems and methods for dynamic pricing"
  - Describes software and hardware required
  - Thompson sampling algorithm (bandit)
- Constant elasticity demand model assumed

$$d_i(p_i) = f_i \left(\frac{p_i}{p_{0,i}} \right)^{\gamma_{*,i}}$$

  - d_i: demand; p_i: price; $p_{0,i}$: baseline price (defined as prior day price)
  - f_i demand at baseline; $\gamma_{*,i}$: own price elasticity
- Does not account for cross price elasticity
  - Algorithm in public paper[2] only consider own price elasticity (slide 12)
  - Follow up patent (2021): competitor price triggers and personalized prices [3]

---

[2] https://arxiv.org/abs/1802.03050

[3] https://patents.google.com/patent/US10896433B2

[4] https://patents.google.com/patent/US20190172082A1

Conjecture: multi-product firms use single product algorithms: extends beyond pricing

- ZOZO is the largest fashion e-commerce company in Japan[5]
- Recommendation problem:
  - There are 3 slots in each page {left, center, right}

  - 80 candidate items to select from
  - Context effects: position, choice set likely exist in this setting

[5] Source: (https://github.com/st-tech/zr-obp)

Conjecture: multi-product firms use single product algorithms: extends beyond pricing

- ZOZO's optimization problem
  - Action: permutation of items
  - Curse of dimensionality: ${}^{80}P_3$ permutations, or $\sim 512k$ actions
- Their solution: top-3 Thompson sampling
  - Product by product bandit
  - Reduces action space to 80 or 0.02% of full problem
  - Assign {left, center, right} as {#1, #2, #3}
- Implications from a randomized field experiment
  - Released data from 7-day experiment in late Nov. 2019
  - Position effects (L, M, R) exists and are heterogeneous
  - Solving permutorial problem could increase overall CTR by at least[6] 10%

[6] insufficient data to estimate full choice set effects

Conjecture: multi-product firms use single product pricing

Prices on amazon.com

- Chen et al (2016) scrape amazon.com prices in 15 minute intervals
  - 1,955 leading products with a total of 33,246 sellers in 2014
  - 1,155 products with amazon.com as the seller
  - Define lowest price as lowest price of all other sellers
- Test-statistic to identify algorithmic sellers
  - *Correlation between amazon price and lowest from other sellers*
  - Correlation with the lowest price (prime only) is 0.34 (0.32)

Conjecture: multi-product firms use single product pricing

Prices on amazon.com

- Add metadata for amazon (He and McAuley 2016, McAuley et al 2015)
  - Identify **related** products as (examples on slide 13)
    - Also bought
    - Also viewed
    - Bought together
    - Buy after viewing
  - Matched products: 906 across 7 categories
- Did Amazon in 2014-15 set prices jointly across all products?

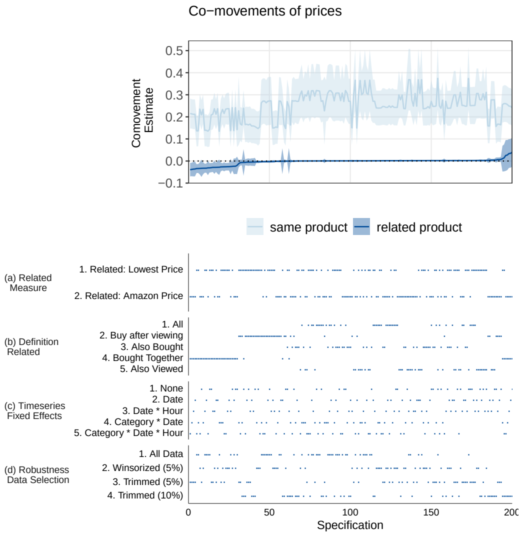
Extend Chen et al (2017) to compare prices *across* related products

Correlation between amazon's price for product 1 and ...

- ... Amazon's price for related product 2
- ... Lowest price for related product 2

Co-movement estimate

$$p_{j,t}^{amazon} = \beta_s p_{j,t}^{-amazon} + \beta_r p_{related(j),t} + \alpha_j + FE_t + \epsilon_{j,t}$$



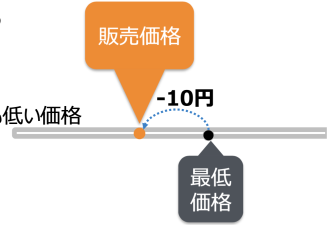
Pricing patterns consistent with single-product pricing

Amazon Japan’s slide for sellers in 2018

価格の自動設定の例 - ルールタイプ：競争力がある

ルール：最低価格*より10円低い価格を維持する
下限価格：490円
上限価格：600円

*同じ商品(ASIN)に出品されている価格のうち最も低い価格



①自動設定をスタート

②最低価格が550円 ➡ 出品者様の価格が540円(550-10円)に更新

③最低価格が510円 ➡ 出品者様の価格が500円(510-10円)に更新

④最低価格が450円 ➡ 440円が設定された下限価格490円を
下回るため、この場合は、出品者様の価格が490円に更新

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Firm behavior: multi-product firms use simplifications
4. Implications: outcomes with simplifications
5. Alternative simplifications

Summary

- Evidence that firms use single-product algorithms
- Simplifies complexity from exponential to multiplicative

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Firm behavior: multi-product firms use simplifications
4. Implications: outcomes with simplifications
   - Theory: collusive steady state is not stable
   - Simulation: prices can be sub-competitive
5. Alternative simplifications

Setup: theory and simulation

Assumed structure

- Multi-product firms: 1 and 2
  - Each sells two products
    - Firm 1 sells products A1 and B1
    - Firm 2 sells products A2 and B2

- *Assume* single product algorithms
  - A_i **price** (P_{Ai}) **considers** $\{A_1, A_2\}$ **prices** $(\{P_{A1}, P_{A2}\})$
  - B_i **price** (P_{Bi}) **considers** $\{B_1, B_2\}$ **prices** $(\{P_{B1}, P_{B2}\})$
  - *Assume* complete on each product separately
  - *Enforces* zero coordination between P_{Ai} and P_{Bi}

- True DGP: demand is shared
  - Consumers observe all prices $(P_{A1}, P_{A2}, P_{B1}, P_{B2})$
  - Consumers pick between all four products (A_1, A_2, B_1, B_2)

Theory: two product prisoner's dilemma

Cross product substitution

- Pricing setup
  - Each product has one of two prices p_H or p_L (p_L is Nash)
  - Each firm follows Q-learning with experimentation (ϵ)
- Demand setup
  - Each consumer has a "home" product
  - Number of consumers for each product scaled to 2 (as before)
  - 2δ consumers willing to switch to lower priced "non home" product
    - If $max(P_{A1}, P_{A2}) = p_H$ and $min(P_{B1}, P_{B2}) = p_L \Rightarrow 2\delta$ consumers switch from product A to product B

Theory: two product prisoner's dilemma

Cross product substitution

Implication for competition between P_{B1}, P_{B2} (symmetric for P_{A1}, P_{A2})

- Case 1: If both P_{A1}, P_{A2} are p_H,

$$
\begin{array}{cc|c|c|}
 & & \multicolumn{2}{c}{P_{B2}} \\
 & & p_H \equiv 1 & p_L \\
\hline
P_{B1} & p_H \equiv 1 & 1, 1 & 0, 2(p_L + \delta) \\
\hline
 & p_L & 2(p_L + \delta), 0 & p_L + \delta, p_L + \delta \\
\hline
\end{array}
$$

- Case 2: If either P_{A1}, P_{A2} are p_L,

$$
\begin{array}{cc|c|c|}
 & & \multicolumn{2}{c}{P_{B2}} \\
 & & p_H \equiv 1 & p_L \\
\hline
P_{B1} & p_H \equiv 1 & 1 - \delta, 1 - \delta & 0, 2p_L \\
\hline
 & p_L & 2p_L, 0 & p_L, p_L \\
\hline
\end{array}
$$

Theory: two product prisoner's dilemma

Cross product substitution

- **Result:** For any p_L, there exists $\underline{\delta} \in [0, 1 - p_L)$ such that if $\delta > \underline{\delta}$, the unique steady state is for both firms to charge price p_L in both markets
- Intuition:
  - From the single product proposition we had if p_L is low enough, there are not sufficient incentives for the algorithm to play Nash
  - Cross-product substitution provides exactly that incentive
- Implication: when multi-product firms use single product algorithms, supra-competitive prices are harder to sustain

Simulations: implications for market outcomes

Multi-product firms implications

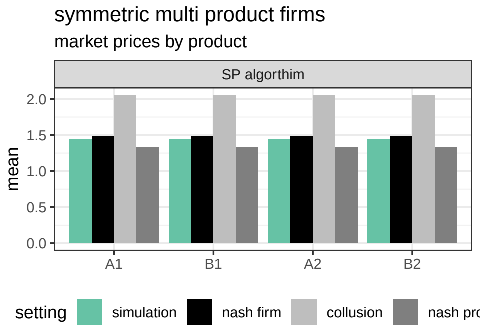
- Multi-product firms using single product algorithms
  - Each sells two products
    - Firm 1 sells products A1 and B1
    - Firm 2 sells products A2 and B2
  - algorithm assumes they complete on each product separately
    - A_i price considers $\{A_1, A_2\}$ prices
    - B_i price considers $\{B_1, B_2\}$ prices
    - **forces** zero-correlation between A_i and B_i
  - demand is shared
    - consumers pick between all four products (A_1, A_2, B_1, B_2)
    - assume logit demand as before

- repeat simulation assuming the firm sells two similar goods
  - Calvano et al. (2022) setup as before

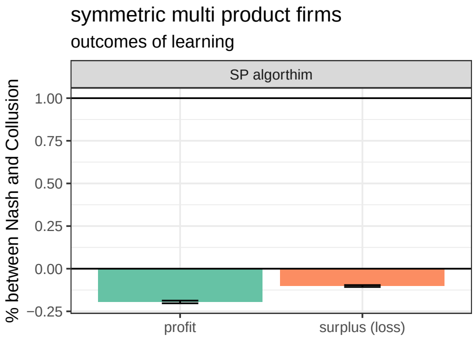
Simulations: implications for market outcomes

Multi-product firms using single product algorithms ...

... Result in prices **at or below Nash**



... **Do not** disadvantage consumers



Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Conjecture: multi-product firms use single-product algorithms
4. Implications: outcomes with single product versus multi-product algorithms
5. Alternative simplifications

Summary

- Single-product algorithms reverse prior results
- Supra-competitive prices less likely when firms simplify pricing

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Conjecture: multi-product firms use single-product algorithms
4. Implications: outcomes with single product versus multi-product algorithms
5. Alternative simplifications: firms set constant markups

Simulations: implications for market outcomes

- Alternative simplification: constant markups
  - Multi-product pricing problem one dimensional
  - Consistent with logit demand
  - Could include potentially unrelated products
- Implications for a pricing game
  - *Assumes* commitment

Setup: theory

Assumed structure

- Multi-product firms: 1 and 2
  - Each sells two products
    - Firm 1 sells products A1 and B1
    - Firm 2 sells products A2 and B2
- *Assume* single product algorithms
  - *Assume* perfect coordination between P_{Ai} and P_{Bi}
  - **Firm i charges the same price for both A_i and B_i ($P_{Ai} = P_{Bi} = P_i$)**
  - Firm i's price P_i considers historical prices $(\{P_1, P_2\})$
- True DGP: demand is shared
  - Consumers observe all prices $(P_{A1}, P_{A2}, P_{B1}, P_{B2})$
  - Consumers pick between all four products (A_1, A_2, B_1, B_2)

Markup problem is similar to single product problem

Theory

- Implied Prisoner's dilemma:

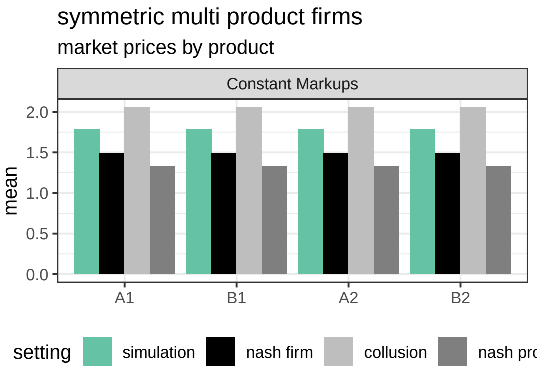
Firm 1 ($P_{A1} = P_{B1}$)		Firm 2 ($P_{A2} = P_{B2}$): $p_H \equiv 1$	Firm 2 ($P_{A2} = P_{B2}$): p_L
	$p_H \equiv 1$	$2, 2$	$0, 4p_L$
	p_L	$4p_L, 0$	$2p_L, 2p_L$

- **Result:** identical to incentives in single-product markets!

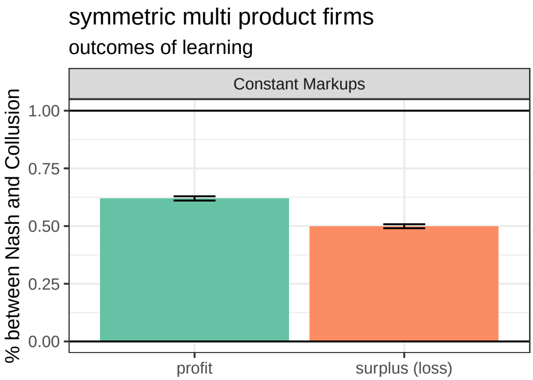
Simulations: implications for market outcomes

Multi-product firms using constant markup product algorithms ...

... Result in **supra-competitive** prices



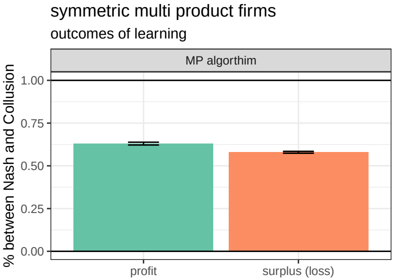
... **Disadvantage** consumers

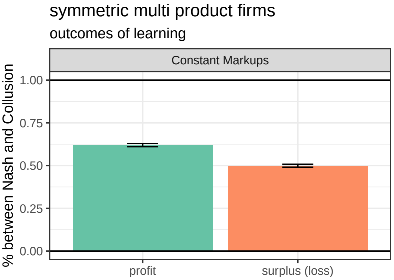


Simulations: implications for market outcomes

Multi-product firms using constant markup product algorithms versus full information

- Constant markup models: **same profits** as a Multi-Product algorithm with less complexity





- Constant markup: mechanically enforces multi-market contact strategies

Empirical markets: constant markup algorithms

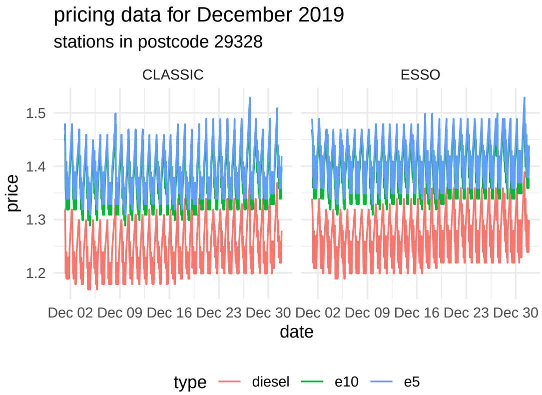
German Gasoline Market

- Assad et al (2021): evidence consistent with algorithmic collusion
  - Focused on prices of E-10 gas and markets at postcodes
  - Infer adoption of algorithmic pricing in/around 2017
  - Prices increased in duopoly markets only when **both** adopted
- German gas station data[7]
  - 15,650 gas stations between 2015 and 2020
  - Include 305 million price changes for gas (within day)
- Gas stations are multi-product firms
  - In these data all stations sell diesel, E-5 and E-10
  - Diesel and gas are *independent* in short term demand

[7] https://dev.azure.com/tankerkoenig/_git/tankerkoenig-data

Empirical markets: constant markup algorithms

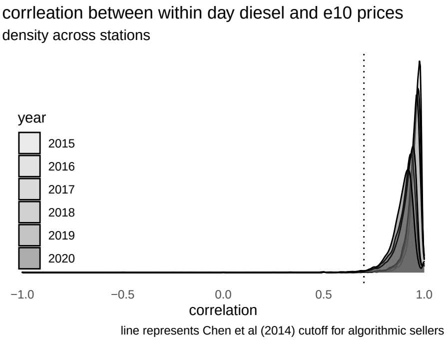
German Gasoline Market

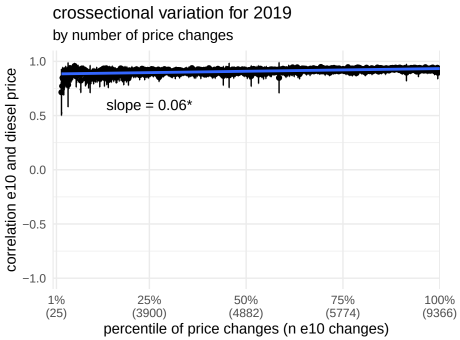


- Stylized fact 1: prices of all types of gas move together!

Empirical markets: constant markup algorithms

German Gasoline Market maintained correlation as stations adopted algorithms





- Stylized fact 2: pricing algorithms maintain correlation in unrelated goods

Overview of talk

1. Current knowledge: market outcomes with single product firms
2. Extension: multi-product firms
3. Conjecture: multi-product firms use single-product algorithms
4. Implications: outcomes with single product versus multi-product algorithms
5. Alternative simplifications: firms set constant markups

Summary

- Outcome: markup algorithms can result in supra-competitive prices
  - Enables multi-market contact strategies
- Simple empirical tests

Summary

- Multi-product pricing can reach supra-competitive outcomes
  - Strategies learned include multi-market contact
  - However, significant complexity
- Simplification: multi-product firms use single-product algorithms
  - Show evidence in practice
  - Multi-product firms using single-product algorithms can reach sub-competitive prices

Summary

- Multi-product pricing can reach supra-competitive outcomes
  - Strategies learned include multi-market contact
  - However, significant complexity
- Simplification: multi-product firms use single-product algorithms
  - Show evidence in practice
  - Multi-product firms using single-product algorithms can reach sub-competitive prices
- Alternative: multi-product firms constant markup algorithms
  - Can reach supra-competitive prices
  - Achieve multi-product price profits with reduced complexity
- Policymakers/empirical research: markets to study
  - Consider co-movement of prices: perfect (constant markup), positive (multi-product), zero (single product)

Appendix

Simulations: implications for market outcomes

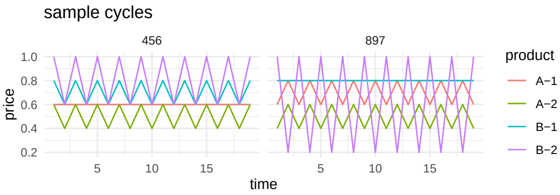
Multi-product algorithm derived strategies: **steady state pricing**

- steady state in this simulation does **not** mean single prices for a good
- example of a 2 time period inferred cycle:
  - Firm 1's strategy (state $\rightarrow$ action) at two states
    1. $\{p_{a1}, p_{b1}, p_{a2}, p_{b2}\} \rightarrow \{p^{\alpha}_{a1}, p^{\alpha}_{b1}\}$
    2. $\{p^{\alpha}_{a1}, p^{\alpha}_{b1}, p^{\alpha}_{a2}, p^{\alpha}_{b2}\} \rightarrow \{p_{a1}, p_{b1}\}$
  - Firm 2's strategy at two states
    1. $\{p_{a1}, p_{b1}, p_{a2}, p_{b2}\} \rightarrow \{p^{\alpha}_{a2}, p^{\alpha}_{b2}\}$
    2. $\{p^{\alpha}_{a1}, p^{\alpha}_{b1}, p^{\alpha}_{a2}, p^{\alpha}_{b2}\} \rightarrow \{p_{a2}, p_{b2}\}$
  - steady state prices osculate between $\{p_{a1}, p_{b1}, p_{a2}, p_{b2}\}$, $\{p^{\alpha}_{a1}, p^{\alpha}_{b1}, p^{\alpha}_{a2}, p^{\alpha}_{b2}\}$
    - say at time 1 we are at prices state $\{p_{a1}, p_{b1}, p_{a2}, p_{b2}\}$
    - time 2: Firm 1 sets $\{p^{\alpha}_{a1}, p^{\alpha}_{b1}\}$ and Firm 2 sets $\{p^{\alpha}_{a2}, p^{\alpha}_{b2}\}$
    - time 3: Firm 1 sets $\{p_{a1}, p_{b1}\}$ and Firm 2 sets $\{p_{a2}, p_{b2}\}$
    - ...
  - cycle does not represent a mixed pricing strategy

Simulations: implications for market outcomes

Multi-product algorithm derived strategies: **steady state pricing**

- cycles (3-4 time periods) of prices as steady state outcomes



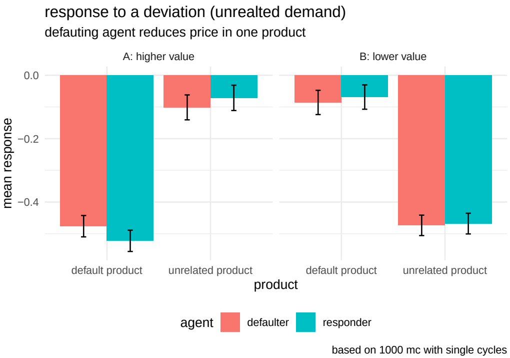
  - demand shared: correlation[8] between a firm's prices = 0.11 (0.07,0.14)
  - demand not shared: correlation between a firm's prices = -0.01 (-0.05,0.03)
- empirical implications: steady state multi-product prices:
  - timing: prices good change at the same time
  - direction: if demand is (not) shared then prices are (not) correlated

[8] to avoid trivial correlations we consider cycles more the 2

Simulations: implications for market outcomes

Multi-product firms can punish deviations in both products even in independent markets

- deviations to steady state prices when products are unrelated in demand



- consistent with multi-market contact

overview of talk

1. conjecture: multi-product firms use single product algorithms
2. theory: could using single product algorithms be optimal?
3. simulations: outcomes with single product versus multi-product algorithms
4. implication: implied tests in empirical markets
    - stylized facts with multi-product algorithms
        - timing: prices move together (cycles)
        - without shocks: correlations in prices for *related* goods
        - with shocks: correlations in prices including *unrelated* goods
    - objective: understand if these are met in the German gasoline market

implication: markets likely with multi-product algorithms

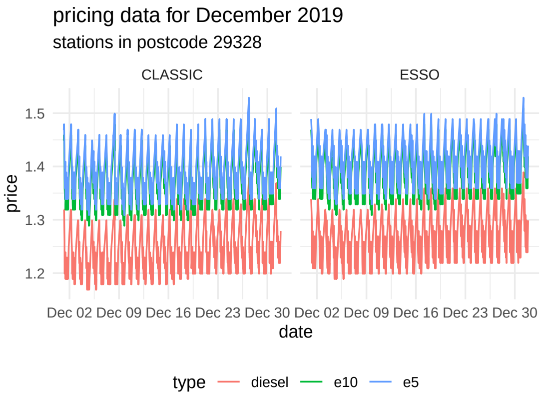
German Gasoline Market

- Assad et al (2021): evidence consistent with algorithmic collusion
  - focused on prices of E-10 gas and markets at postcodes
  - infer adoption of algorithmic pricing in/around 2017
  - prices increased in duopoly markets only when **both** adopted
- German gas station data[9]
  - 15,650 gas stations between 2015 and 2020
  - include 305 million price changes for gas (within day)
- gas stations are multi-product firms
  - in these data all stations sell diesel, E-5 and E-10
  - diesel and gas are *independent* in short term demand

[9] https://dev.azure.com/tankerkoenig/_git/tankerkoenig-data

implication: markets likely with multi-product algorithms

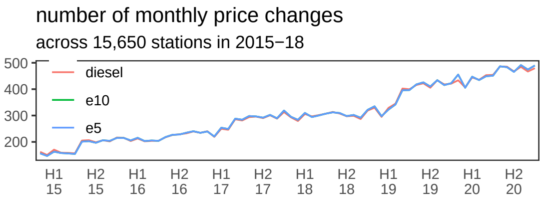
German Gasoline Market

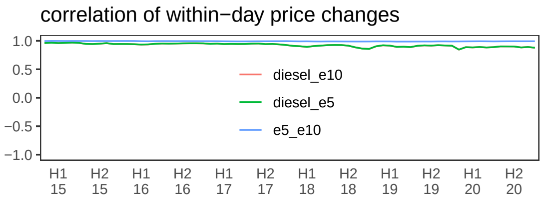


- stylized fact 1: prices of all types of gas move together!

implication: markets likely with multi-product algorithms

German Gasoline Market

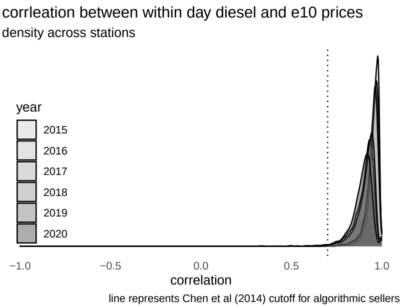


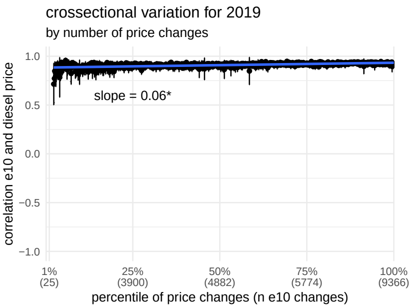


- we find an increase in number of price changes in 2017
- however gas and diesel continue to move together

implication: markets likely with multi-product algorithms

German Gasoline Market maintained correlation as stations adopted algorithms





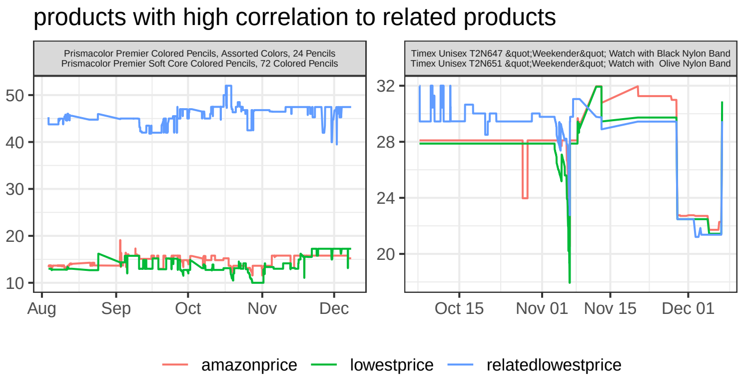
- stylized fact 2: pricing algorithms maintain correlation in unrelated goods

summary

1. conjecture: multi-product firms use single product algorithms
   - show evidence consistent with single product algorithms in practice
2. theory: could using single product algorithms be optimal?
   - for a general problem - no!
3. simulations: implications for market outcomes
   - multi-product firms using single product algorithms can reach sub-competitive prices
   - multi-product firms using multi-product algorithms can reach supra-competitive prices
4. implication: markets likely with multi-product algorithms
   - steady state prices move at the same time
   - without price shocks: correlations in prices across related products
   - with price shocks: correlations in prices across unrelated products

Conjecture: multi-product firms use single product algorithms: prices on amazon.com

Note: examples where prices are correlated



Walmart labs algorithm

https://arxiv.org/abs/1802.03050

Algorithm 1: Architecture of the proposed dynamic pricing engine.

Input: A basket $\mathcal{B}$, and time period T over which we intend to maximize cumulative revenue

for $t \leftarrow 1$ ***to*** T **do**

1. For each item $i \in \mathcal{B}$ calculate their demand forecasts using the demand forecaster.
2. For each item $i \in \mathcal{B}$ calculate their price elasticities $\gamma_{\star,i}$.
3. Solve the MAX-REV optimization problem, shown in Equation (7) to obtain new prices $\mathbf{p}_t$.
4. Apply these prices and observe the revenue obtained R_t.

end

Conjecture: multi-product firms use single product pricing

Examples of related products (He and McAuley 2016 and McAuley et al 2015)

Philips AVENT BPA Free Classic Infant Starter Gift Set	The First Year's Infant To Toddler Tub with Sling, Blue
Camco 40043 TastePURE Water Filter with Flexible Hose Protector	Camco 40055 Brass Water Pressure Regulator
American Baby Company 100% Cotton Value Jersey Knit Fitted Portable/Mini Sheet, Celery	American Baby Company 100% Organic Cotton Interlock Fitted Pack N Play Sheet, Natural
Darice 80-Piece Deluxe Art Set	Pro Art 18-Piece Sketch/Draw Pencil Set
Dove Bar Soap, Sensitive Skin Unscented, 4 Ounce, 16 Count	Quilted Northern Ultra Plush Bath Tissue, 48 Double Rolls
Minecraft: Essential Handbook: An Official Mojang Book	Minecraft: Redstone Handbook: An Official Mojang Book
Foscam FI9821P Plug & Play Megapixel 1.0 Megapixel 1280 x 720 Wireless/Wired Pan/Tilt IP Camera with IR-Cut (Black)	Foscam FI8910W Pan & Tilt IP/Network Camera with Two-Way Audio and Night Vision (Black)
NETGEAR N450 WiFi DOCSIS 3.0 Cable Modem Router (N450-100NAS)	NETGEAR N600 WiFi DOCSIS 3.0 Cable Modem Router (C3700)

also bought also viewed bought together buy after viewing

Back