Tilburg University

De mechanismes van algoritmische collusie

Svitak, Jan; van der Noll, Rob

Published in:
Tijdschrift voor Toezicht

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):
Svitak, J., & van der Noll, R. (2020). De mechanismes van algoritmische collusie. *Tijdschrift voor Toezicht*, 14(1), 14-27.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Tilburg University

De mechanismes van algoritmische collusie

Svitak, Jan; van der Noll, Rob

Published in:
Tijdschrift voor Toezicht

Publication date:
2020

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):
Svitak, J., & van der Noll, R. (2020). De mechanismes van algoritmische collusie. *Tijdschrift voor Toezicht*, 14(1), 14-27.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

De mechanismes van algoritmische collusie

Jan Sviták en Rob van der Noll*

14

De populariteit van zelflerende algoritmes heeft zorgen gewekt bij academici en mededingingstoezichthouders over het mogelijk prijsverhogende effect van algoritmes voor prijszetting. Experts waarschuwen dat zelflerende algoritmes stilzwijgend zouden kunnen samenspannen en hogere prijzen kunnen bewerkstelligen (colluderen). Aan de andere kant zijn er auteurs die stellen dat coördinatie zelfs voor algoritmes te complex is om zonder verboden informatie-uitwisseling te bereiken. Wij overbruggen het gat tussen theoretische zorgen en dit praktisch scepticisme. In dit artikel analyseren wij mechanismes die ervoor zorgen dat zelflerende algoritmes in een setting met concurrentie relatief hoge prijzen kunnen bereiken, met een onderscheid tussen vorming en stabiliteit van stilzwijgende samenspanning. We analyseren de praktische toepasbaarheid van deze mechanismes. Stabiliteit-bevorderende mechanismes kunnen voorkomen, maar we vinden het minder aannemelijk dat ook de prijsverhogende mechanismes in de praktijk kunnen worden gerealiseerd, vanwege de aanzienlijke kosten van experimentatie met prijzen, die essentieel is voor deze mechanismes. We concluderen dat het gebruik van AI (*Artificial Intelligence*) voor prijszetting mechanismes biedt die het risico op algoritmische collusie verhogen, maar de praktische belemmeringen beteken dat het risico op dit moment beperkt lijkt te zijn

tot specifieke markten of samenloop van omstandigheden.

Inleiding

In de laatste decennia zijn bedrijven steeds meer dynamisch gaan prijzen. Dat houdt in dat de prijzen continu aangepast worden aan de veranderende behoefte van consumenten voor het betreffende product.¹ Deze behoefte zal afhankelijk zijn van ontwikkelingen in de markt. De vraag is dus afhankelijk van bijvoorbeeld plaats, tijd en beschikbaarheid en prijzen van vergelijkbare producten bij concurrenten. Om de prijzen te kunnen aanpassen aan de omstandigheden moeten bedrijven in staat zijn om deze omstandigheden een rol te geven in de prijszettingstrategie. Den Boer² biedt een uitgebreid literatuuroverzicht over dynamisch prijzen en leren in een veranderende omgeving. Hoewel leren over marktomstandigheden altijd al onderdeel van de prijszetting was (denk bijvoorbeeld aan de inschatting van de vraag), heeft de vooruitgang in *computer science* in de laatste jaren ervoor gezorgd dat *machine learning* technieken steeds vaker worden gebruikt voor het oplossen van prijszettingspuzzels.

* Dhr. J. Sviták is econometrist bij het Economisch Bureau van de Autoriteit Consument en Markt en extern PhD student aan Tilburg University. Dr. R. van der Noll is senior econoom bij het Economisch Bureau van de Autoriteit Consument en Markt. De code met enkele aanvullende resultaten is beschikbaar als online appendix op <https://github.com/jan-svitak/Mechanismes-van-algoritmische-collusie>

1. W. Reinartz, 'Customizing prices in online markets', *Symphonya. Emerging Issues in Management* 2002, nr. 1, p. 55-65.
2. A.V. den Boer, 'Dynamic pricing and learning: historical origins, current research, and new directions', *Surveys in operations research and management science* 2015, nr. 1, p. 1-18.

Het toenemende gebruik van geavanceerde prijsalgoritmes³ heeft zorgen gewekt bij mededingings-toezichthouders en wetenschappers over de kans dat deze technieken het bereiken van stilzwijgende of expliciete collusie makkelijker maken. Ezrachi en Stucke⁴ beschrijven enkele scenario's waarin prijsalgoritmes gebruikt kunnen worden om hoge prijzen te bereiken. Ze concluderen dat simpele algoritmes met voorgeschreven regels ingezet kunnen worden om collusieafspraken uit te voeren. Ze vinden ook dat het gebruik van algoritmes de transparantie vergroot, hetgeen tot een hogere kans op stilzwijgende collusie leidt. Tot slot bespreken ze de mogelijkheid dat stilzwijgende afstemming bereikt zou kunnen worden door zelflerende algoritmes. In tegenstelling tot bij de andere scenario's kan het bij dit laatste scenario moeilijk zijn om de intentie om te colluderen aan te tonen, als het ontwerp van de algoritmes niet expliciet gestuurd werd richting collusie-uitkomsten. Tegelijkertijd kan het deze AI-algoritmes echter wel lukken om samen collusieve uitkomsten te bereiken. Mededingingsautoriteiten staan dan dus voor de uitdaging om deze uitkomsten te bestrijden met hun huidige instrumenten.

In dit artikel analyseren wij of, en zo ja waarom, AI-algoritmes in staat zijn om de coördinatie op prijzen makkelijker te maken. Dat doen wij door middel van simulaties en beschrijving van het ontwerp van de gebruikte AI-algoritmes. Wij kijken naar algoritmes die aan de volgende definitie van AI van Kaplan en Haenlein⁵ voldoen: 'a system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation'.

Wij focussen verder op algoritmes die zelflerend zijn, wat betekent dat ze actief leren.⁶ Actief leren houdt in dat een algoritme 'zelf' bepaalt of en wanneer een minder optimale keuze wordt gemaakt om leren over niet-geëxploreerde opties te faciliteren. Passief leren leunt aan de andere kant alleen op kennis over uitgevoerde acties.

De literatuur⁷ over het vermogen van zelflerende algoritmes om samen te spannen en hogere prijzen te hanteren neemt toe. Voor zover wij weten, ontbreekt echter een inzicht in hoe precies de uitkomst met hoge prijzen wordt bereikt en waarom het concurrentiemechanisme zijn werk niet doet. Academics bespreken in beperkte mate de kans op algoritmische collusie in de praktijk en focussen op langetermijnuitskomsten van simulaties. Er zijn ook onderzoekers die de algoritmische collusie in de praktijk onwaarschijnlijk vinden.⁸ In een notendop: ze vinden het coördinatieprobleem in reële markten zodanig complex dat het ook voor algoritmes niet op te lossen is zonder communicatie. Informatie-uitwisseling is echter illegaal, ook voor algoritmes. We willen dit gat tussen theoretische zorgen en praktisch scepticisme overbruggen, intuïtie bieden bij de experimentele resultaten, inzoomen op kortetermijnuitskomsten van de algoritmes en toelichten op welke manier deze technieken in reële markten gebruikt kunnen worden.

We kijken voornamelijk naar algemene mechanismes die onderdeel zijn van wat wij 'onschuldige' zelflerende algoritmes noemen. 'Onschuldig' betekent dat er geen prijsverhogende mechanismes doelbewust zijn ingebouwd. In die zin volgen wij Brown & MacKay.⁹ We bootsen situaties na waarin twee bedrijven met elkaar concurreren, in die zin dat als Bedrijf 1 de prijsstrategie kent van Bedrijf 2, het rationeel is voor Bedrijf 1 om zijn prijs net iets lager te zetten dan de prijs van Bedrijf 2 (en andersom). Voor deze setting voorspelt de logica van een *one-shot game* dus een relatief lage prijs voor beide bedrijven. Die uitkomst noemen we de competitieve prijs. We gebruiken de term 'evenwicht' losjes en die kan ook worden gelezen als uitkomst. Elke prijs boven het competitieve niveau is echter schadelijk voor afnemers en dat wordt de *benchmark* waartegen de resultaten van de simulaties worden afgezet. In dit artikel kijken wij naar prijsniveaus over de tijd en brengen in kaart hoe vaak die afwijken van de competitieve prijs. De definitie van collusie of de vraag of de uitkomsten van de simulaties juridisch zouden kwalificeren als collusie vallen buiten de reikwijdte van dit artikel. We beschrijven geen volledige collusiestrategie, maar het bereiken van hogere prijzen is daar wel een onderdeel van. Dit onderdeel vinden we op zichzelf belangrijk voor analyse. In dit artikel analyseren wij de algoritmes aan de hand van een beschrijving van de onderliggende mechanismes en hun prestatie in een gesimuleerd prijszettingsspel

3. Europese Commissie, *Final report on the E-commerce Sector Inquiry*, 2017; Oxera, *When algorithms set prices: Winners and losers*, 2017; J. Miklós-Thal en C. Tucker, 'Collusion by Algorithm: Does Better Demand Prediction Facilitate Coordination Between Sellers?', *Management Science* 2019, nr. 4, p. 1552-1561.
4. A. Ezrachi en M.E. Stucke, *Algorithmic collusion: Problems and counter-measures*, Submitted as background material at the Roundtable on Algorithms and Collusion at the OECD Competition Committee 2017; A. Ezrachi en M.E. Stucke, 'Artificial intelligence & collusion: When computers inhibit competition', *U. Ill. L. Rev.* 2017, p. 1775; A. Ezrachi en M.E. Stucke, 'Two artificial neural networks meet in an online hub and change the future (Of competition, market dynamics and society)', *Oxford Legal Studies Research Papers* 2017.
5. A. Kaplan en M. Haenlein, 'Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence', *Business Horizons* 2019, nr. 1, p. 15-25.
6. E. Calvano, G. Calzolari, V. Denicolò en S. Pastorello, 'Algorithmic pricing what implications for competition policy?', *Review of Industrial Organization* 2019, nr. 1, p. 155-171 (Calvano e.a. 2019a).

7. E. Calvano, G. Calzolari, V. Denicolò en S. Pastorello, 'Artificial intelligence, algorithmic pricing and collusion', *Working paper* 2019 (Calvano e.a. 2019b); K. Hansen, K. Misra en M. Pai, *Algorithmic Collusion: Supra-competitive Prices via Independent Algorithms*, Londen: Centre for Economic Policy Research 2020; T. Klein, *Autonomous Algorithmic Collusion: Q-Learning Under Sequential Pricing*, Amsterdam Law School Research Paper 2018, nr. 15/Amsterdam Center for Law & Economics Working Paper 2018, nr. 5.
8. T. Schrepel, 'Here's why algorithms are NOT (really) a thing', *Revue Concurrentialiste* 2017; U. Schwalbe, 'Algorithms, machine learning, and collusion', *Journal of Competition Law & Economics* 2019, nr. 4, p. 568-607; K. Kühn en S. Tadelis, *Algorithmic Collusion*, CRESSE 2017.
9. Z. Brown en A. MacKay, *Competition in Pricing Algorithms*, 2019, beschikbaar via SSRN 3485024.

waarin wij elke keer twee bedrijven met elkaar laten concurreren.

De opbouw van dit artikel is als volgt:

- Beschrijving verschillende types zelflerende algoritmes;
- Hypotheses over prijsverhogende mechanismes van zelflerende algoritmes;
- Opzet en resultaten van de simulaties;
- Manier van toepassing van reinforcement learning algoritmes in de praktijk;
- Conclusies over de hypothesen en de risico's op col-lusie door 'onschuldige' zelflerende algoritmes.

Zelflerende algoritmes

Zelflerende algoritmes worden gebruikt voor een optimale uitvoering van diverse taken. In dit artikel concen-treren wij ons op de taak van de optimale prijszetting. Een belangrijke eigenschap van zelflerende algoritmes is dat ze geen historische data nodig hebben voor de berekening van prijzen, maar leren door ervaring. Ze ont-wikkelen strategieën die uit historische data niet naar voren zouden komen. Dat is een verschil met traditione-le statistische methoden die een grote hoeveelheid histo-riche data vereisen nog voordat ze een prijs kunnen bepalen. Het gebruik van zelflerende algoritmes draagt dus bij aan de oplossing van problemen met beperkte databeschikbaarheid, onvoldoende variatie in prijzen en endogeniteit.

Er zijn vele types zelflerende algoritmes. Wij gebruiken een classificatie op basis van complexiteit en de manier waarop informatie over de marktomstandigheden in besluitvorming wordt meegenomen. De algoritmes krij-gen een aantal keuzes (mogelijke prijzen) die winst gene-deren. De algoritmes observeren bij elke (eigen) prijs hoeveel winst die keuze oplevert en registreren dat. Dit probleem wordt het *multi-armed bandit* probleem genoemd. Algoritmes houden alleen rekening met het effect van de verschillende keuzes op winst zoals weer-gegeven in Figuur 1. Men kan denken aan een reeks fruitautomaten met verschillende winkansen en winsten. Het algoritme probeert dan de fruitautomaat te identi-ficeren die de hoogste winst oplevert. De algoritmes moe-ten dus alle beschikbare keuzes exploreren. In het geval van fruitautomaten betekent het bij elke machine een paar keer spelen en bij prijszetting betekent het uitpro-beren van verschillende prijzen. Het kiezen van de optie met grootste verwachte winst noemen wij exploitatie. De uitruil tussen exploratie en exploitatie, ofwel tussen het verzamelen van informatie over keuzes en het uit-voeren van de beste actie op basis van beschikbare informatie, is een kernprobleem voor alle zelflerende algoritmes. Met te weinig exploratie gaat het algoritme de actie uitvoeren die op dat moment het beste lijkt, en niet voldoende leren over alternatieven. Mogelijke bete-re alternatieven worden hierdoor gemist. Het kan echter ook gebeuren dat de geïdentificeerde beste actie inder-daad optimaal is. Dan verlaagt de exploratie de winst.

Het algoritme berekent voor elke beschikbare keuze een waarde van een criterium, die bepaalt welke optie geko-zen dient te worden. Op elk beslismoment wordt dus de optie met de hoogste score op het criterium gekozen en de geobserveerde winst wordt gebruikt om de waarden van het criterium te updaten. Exploratie kan direct inge-bouwd zijn in het criterium¹⁰ of geïmplementeerd worden door het algoritme met een bepaalde kans een willekeurige actie te laten kiezen.¹¹

Figuur 1 Een multi-armed bandit probleem

In complexere situaties is de winst niet alleen afhanke-lijk van de acties van het algoritme zelf maar ook van de omgeving. De omgeving waarin het algoritme opereert, kan in kaart worden gebracht als we naar het spel kijken als een *contextual bandit* probleem.¹² De winst wordt nu bepaald door de combinatie van uitgevoerde actie en de omstandigheden in de omgeving zoals weergegeven in Figuur 2. Een *contextual bandit* algoritme houdt nog steeds een criteriumwaarde bij voor elke optie. De waarde wordt nu ook gebaseerd op de geobserveerde omstandigheden (context), naast de winst. Dat kan bijvoorbeeld door middel van een statistisch model, dat de geobserveerde context als input gebruikt, of door een aparte berekening van de criteriumwaarden voor elke gedefinieerde versie van de omgeving. De geobserveerde omstandigheden kunnen veranderen en het algoritme is dus in staat om op elk beslismoment die actie kiezen die naar verwachting de beste is in die situatie.

Figuur 2 Een contextual bandit probleem

Tot slot kunnen wij nog een stap verder gaan en in kaart brengen hoe uitgevoerde acties de omgeving beïnvloe-den. Als wij aan prijszetting denken, kunnen wij ons voorstellen dat prijzen van concurrenten een onderdeel van de omgeving zijn. De gehanteerde prijzen (uitge-

10. Het criterium wordt bijvoorbeeld hoger als de optie weinig keren wordt geobserveerd of als er grote variantie is in de winsten voor deze optie.

11. ϵ -greedy strategie kiest bijvoorbeeld de optimale optie met de kans $1 - \epsilon$ en een toevallige optie met de kans ϵ . Andere strategieën, bijvoor-beeld *Thompson-sampling*, kunnen bij de toevallige selectie gebruikma-ken van een kansverdeling op basis van het criterium en zo de naar ver-wachting betere opties met grotere kans kiezen; R.S. Sutton en A.G. Barto, *Introduction to reinforcement learning* (Vol. 135), Cambrid-ge: MIT Press 1998.

12. P. Auer, N. Cesa-Bianchi, Y. Freund en R.E. Schapire, 'The nonstochastic multiarmed bandit problem', *SIAM journal on computing* 2002, nr. 1, p. 48-77.

voerde acties) hebben dan potentieel niet alleen invloed op de huidige winst, maar ook op de toekomstige omgeving, omdat de concurrent kan reageren op de gehanteerde prijzen. Zo worden indirect ook de toekomstige optimale keuzes en winst beïnvloed zoals getoond in Figuur 3. Vanwege het circulaire karakter van de verbanden wordt deze situatie een *reinforcement learning* probleem genoemd. Een *reinforcement learning* algoritme berekent een criteriumwaarde voor elke beschikbare actie gegeven de geobserveerde omgevingskenmerken. Het criterium bevat echter ook een representatie van de toekomstige (verdisconteerde) winsten. De criteriumwaarde wordt dus niet geüpdatet met een combinatie omgeving-actie-winst voor één periode maar met de informatie over meerdere periodes.

Collusiebevorderende mechanismes

Voor de drie types lerende algoritmes verkennen we hoe deze technieken tot hogere prijzen kunnen leiden. Dit doen wij aan de hand van een simpele marktomgeving: een markt met twee bedrijven, die twee horizontaal gedifferentieerde producten aanbieden. De bedrijven zetten simultaan de prijzen en hebben te maken met vaste marginale kosten van 0.

Aan de hand van deze uitgangspunten en een gespecificeerd vraagmodel¹³ kunnen wij de winsten voor alle combinaties van prijzen uitrekenen en aan de hand daarvan de concurrentieprikkels illustreren. De winsten zijn weergegeven in de *heat map* in Figuur 4. De concrete

hoogte van de winsten zou veranderen met een ander vraagmodel maar de algemene mechanismes van concurrentie blijven van toepassing bij elk vraagmodel met heterogene maar substitueerbare producten. De kleur toont hoe hoog de winst voor elk bedrijf wordt bij een bepaalde combinatie van prijzen in de betreffende periode. De grafiek kan worden gezien als een soort tabel waar kolommen de winsten per prijs weergeven, gegeven de actie van de concurrent. Rijen geven de winsten afhankelijk van de prijs van de concurrent. Gegeven de prijs van de concurrent kunnen wij bepalen welke beschikbare actie optimaal is door te kijken naar de kleur. Dat komt neer op het kiezen van de prijs, die in de betreffende rij de maximale winst oplevert. Deze optimale reacties zijn met groene vakjes gemarkeerd. Als de concurrent bijvoorbeeld de prijs van 0.6 hanteert, is 0.5 de optimale reactie omdat deze meer winst oplevert dan alle andere opties.

Voor de bedrijven is de optimale uitkomst dat beide bedrijven de prijs 0.9 rekenen, de competitieve prijs¹⁴ is 0.4. In Figuur 4 kunnen wij echter zien dat als de concurrent de collusieprijs van 0.9 hanteert, de optimale reactie is om de prijs op 0.6 te zetten. De reactie daarop is verdere verlaging naar 0.5 en vervolgens naar 0.4. De optimale reactie op 0.4 is om de prijs van 0.4 te blijven hanteren en daarom is dit punt een evenwicht in deze situatie. De collusieprijs is gelijk aan 0.9 omdat het de hoogste gezamenlijke winst oplevert – het is namelijk de optie met de hoogste winst op de diagonaal in Figuur 4 waar beide bedrijven gelijke prijzen hanteren.

Hieronder zullen we bespreken hoe de optimale reacties die volgens de logica in Figuur 4 in een eenmalig spel tot een competitieve prijs leiden, mogelijk niet worden gerealiseerd door het gebruik van zelflerende algoritmes.

Exploratiemechanisme

Zelfs de eenvoudigste lerende algoritmes leren door exploratie en lossen het dilemma van exploratie versus exploitatie op. Veel algoritmes hebben geen specifieke weergave van het spel dat ze spelen, en daardoor is exploratie vaak de enige manier om te leren. In de context van prijszetting betekent dit dat de algoritmes per definitie verschillende prijzen zullen uitproberen – inclusief hoge prijzen. Als concurrenten tegelijkertijd met hogere prijzen experimenteren, zullen ze collusiewinsten verdienen en mogelijk leren deze prijzen als optimaal te zien. Elke toevallige exploratie kan zo worden gezien als een poging om op hogere prijzen te coördineren. Hoe meer de manier van exploratie van

13. Logitmodel met constante gelijk aan 5 en de prijsparameter gelijk aan -5.

14. Bertrand-Nash evenwicht.

Figuur 4 Winsten voor verschillende combinaties van prijzen

concurrenten op elkaar lijkt, hoe groter de kans dat de coördinatie lukt.¹⁵

Ook voor geavanceerdere algoritmes blijft exploratie essentieel. Het actief leren maakt het namelijk mogelijk om uitkomsten buiten het evenwichtspad te leren en potentieel alternatieve evenwichten te ontdekken. Volgens Fantenberg en Levine¹⁶ is het actief leren cruciaal voor het bereiken van coördinatie in herhaalde spellen. In een rapport van het Witte Huis¹⁷ wordt geconcludeerd dat experimenteren met prijzen nu al in praktijk voorkomen.

Datagedreven mechanisme

Het observeren van de omgeving maakt het mogelijk om op prijzen van concurrenten te reageren. Een strategie waarbij de prijzen van de concurrent worden gevolgd zal de stabiliteit van een eventueel evenwicht met hogere prijzen bevorderen omdat het afwijken van de gehanteerde prijs niet gezien wordt als optimale actie. De prikkel om net iets lager te prijzen en op die manier veel extra verkopen binnen te halen wordt niet gevoeld. Het is echter niet duidelijk hoe dit mechanisme de vorming van collusie zou kunnen bevorderen. Theoretisch gezien is het mogelijk dat een *contextual bandit* algoritme de strategie van de concurrent leert en daarop inspeelt als de geobserveerde strategie overeenkomt met collusie. Om de strategie te kunnen doorgronden is echter voldoende variatie in de data nodig wat het belang van exploratie versterkt. De werking van dit mechanisme is helemaal afhankelijk van de strategieën gehanteerd door concurrenten en wij gaan hier dus niet diep op in.

Dit mechanisme kan wel zeer relevant zijn voor het ontwikkelen van algoritmes, die als specifiek doel hebben om te colluderen. In dit artikel focussen wij echter op 'onschuldige' algoritmes gebaseerd op algemene principes van lerende algoritmes. Het is wel belangrijk om te beseffen dat het met informatie over de omgeving mogelijk is om volgedrag of beloning-strafstrategieën in te

bouwen. Een beloning-strafstrategie houdt in dat een afwijking van de collusieve prijs gevolgd wordt door een reactie van de concurrent, die de afwijkende onderneming straft voor de afwijking door een lagere prijs te zetten zodat er duidelijk verschil ontstaat tussen de winsten met en zonder collusie. Deze strategie vormt volgens Harrington¹⁸ de basis van de definitie van collusie.

Vooruitkijkende strategieën

Zoals besproken in de vorige paragraaf houden *reinforcement learning* algoritmes specifiek rekening met invloed van hun acties op toekomstige omstandigheden en winsten. In de context van prijszetting betekent dit dat het algoritme niet alleen kijkt naar de gevolgen van de gehanteerde prijs voor huidige winst maar ook naar de gevolgen voor toekomstige prijzen (eigen en van concurrenten) en bijhorende winst. Dit maakt het theoretisch mogelijk dat de optimale reacties van Figuur 4 veranderen. Wij hebben al besproken dat de situatie zoals weergegeven in Figuur 4 tot het competitieve evenwicht van 0.4 leidt omdat het altijd op korte termijn winstgevend is om van een hogere prijs af te wijken en een groter marktaandeel binnen te halen met een lagere prijs.

Een *reinforcement learning* algoritme zal echter leren dat deze afwijking tot een reactie van de concurrent leidt en dat het bereikte voordeel door de prijsverlaging maar tijdelijk is. Het algoritme kan namelijk een periode van stabiele prijszetting op hoog prijsniveau vergelijken met een periode waarin prijsverlagingen voorkomen. Vergelijking van deze twee stromen van winsten leidt dan mogelijk tot een realisatie dat het blijven hanteren van hoge prijzen optimaal is omdat de winst over de gehele geobserveerde periode hoger is dan over de periode van een prijsoorlog. *Reinforcement learning* algoritmes houden dus expliciet rekening met het herhaalde karakter van het spel en hebben de maximalisatie van de

15. Hansen e.a. 2020.

16. D. Fudenberg en D.K. Levine, 'Whither game theory? Towards a theory of learning in games', *Journal of Economic Perspectives* 2016, nr. 4, p. 151-70.

17. White House report, *Big Data and Differential Pricing*, 2015.

18. J.E. Harrington, 'Developing Competition Law for Collusion by Autonomous Artificial Agents', *Journal of Competition Law & Economics* 2018, nr. 3, p. 331-363.

netto contante waarde van de toekomstige winsten over de hele duur van het spel als doel.¹⁹

Het exploreren – dus actief leren – blijft ook voor dit mechanisme essentieel. Anders kunnen de algoritmes namelijk de verschillende strategieën en de resulterende winsten niet vergelijken en er niet van leren.

Simulaties

Wij voeren simulaties uit in een omgeving die overeenkomt met Figuur 4. Wij laten twee bedrijven met elkaar concurreren door middel van verschillende algoritmes. Ze kunnen allebei in elke periode uit tien mogelijke prijzen tussen 0.1 en 1 kiezen. Deze keuze vindt simultaan plaats. De bedrijven kunnen dus elkaars historische prijzen observeren maar zijn niet in staat om binnen dezelfde periode op elkaars keuzes te reageren. De vraag naar deze producten volgt een logitmodel.²⁰ Het logitmodel staat heterogeniteit tussen producten toe, dat is realistischer dan (perfecte) homogeniteit. Bovendien wordt het logitmodel veel gebruikt in de toegepaste industriële organisatie.²¹ In die literatuur wordt ervan uitgegaan dat bedrijven de winst maximaliseren. In dit artikel hangt de precieze doelfunctie die wordt geoptimaliseerd echter af van het ontwerp van het algoritme; de variaties daarin zullen we bespreken.

De uitkomst die we verwachten op basis van eenmalige interactie is 0.4; dit noemen we de competitieve prijs. De collusieprijs is 0.9. In de literatuur wordt vaak ook de monopolieprijs genoemd. Als er maar één product door één bedrijf werd aangeboden in deze markt zou de optimale prijs 0.8 zijn. De monopolieprijs is lager dan de collusieprijs omdat bedrijven bij het bestaan van twee producten de heterogeniteit tussen de producten en de verschillende voorkeuren van consumenten kunnen benutten. Voor de drie typen algoritmes bespreken we nu hoe de prijs wordt bepaald.

Multi-armed bandit algoritmes

Om de exploratie hypothese in de puurste vorm te toetsen gebruiken wij een simpel *multi-armed bandit* algoritme, dat geen rekening houdt met de omgeving en alleen eigen keuze en winst observeert per periode. Concreet maken wij gebruik van het UCB1-tuned algoritme.²²

UCB staat voor *Upper Confidence Bound*. Op basis van de in voorgaande periodes gerealiseerde winsten berekent het algoritme een criteriumwaarde, bestaande uit de gemiddelde winst voor elke mogelijke prijs plus een exploratiebonus. De exploratiebonus hangt positief af van de variantie en hangt negatief af van hoe vaak de actie is gespeeld. De criteriumwaarde vormt dus de *upper bound* op de verwachte winst van een bepaalde actie. Als het algoritme veel variatie in winsten ziet bij een bepaalde prijs wordt dat gezien als een kans dat de winst vaker bij de bovengrens zal liggen en deze opties krijgen daarom een hogere exploratiebonus. Dat maakt dat deze algoritmes ‘optimistisch’²³ worden genoemd. Grote variantie in geobserveerde winsten bij een bepaalde prijs betekent ook dat het algoritme niet goed weet wat de ‘ware’ verwachte winst is voor deze optie. Daarom is het de moeite waard om over deze optie meer te leren en deze onzekerheid weg te nemen.

Het UCB1-tuned algoritme berekent concreet in elke periode T voor elke beschikbare prijs p de criteriumwaarde (*upper bound*) Q_p ²⁴ als volgt:

$$Q_p = \bar{\pi}_p + \sqrt{\frac{\log T}{n_p} \min\left(\frac{1}{4}, V_p\right)}$$
$$V_p = \frac{1}{n} \sum_{t=1}^T \pi_{pt}^2 - \bar{\pi}_p^2 + \sqrt{\frac{2 \log T}{n_p}}$$

waar $\bar{\pi}_p$ de gemiddelde winst is voor de prijs p , π_{pt} de winst is voor prijs p in periode t en n_p weergeeft hoe vaak prijs p werd gekozen.

De bedrijven kiezen aan het begin van het experiment elke prijs één keer in een toevallige volgorde. Zo kan voor elke actie een criteriumwaarde worden berekend. In elke periode wordt vervolgens de prijs met de hoogste *upper bound* gekozen en de waarden worden na elke periode geüpdatet.

Het algoritme gaat uit van het bestaan van een optimale prijs en is daarnaar op zoek. Omdat er geen direct verband is tussen de beslissingen in verschillende periodes en geen kennis over hoe lang het spel doorgaat, wordt geen gebruik gemaakt van verdiscontering. Het UCB1-tuned algoritme is wel zo ontworpen om de optimale uitruil tussen exploitatie en exploratie te vinden in een stabiele omgeving en dus de som van de verwachte winsten over het hele spel te maximaliseren.

Hansen e.a.²⁵ laten zien dat het UCB1-algoritme op lange termijn tot collusieve uitkomsten leidt. Ze tonen verder aan dat alle algoritmes gebaseerd op exploratie door middel van een index²⁶ (criteriumwaarde) tot gecorrleerde prijzen leiden tussen de twee bedrijven waardoor coördinatie op supracompetitieve prijzen mogelijk is.

19. Daarbij speelt verdiscontering van toekomstige winsten een rol. Het mechanisme kan alleen werken als het belang van toekomstige winst voldoende is ten opzichte van de huidige winsten.

20. Het logitmodel wordt vaak gebruikt voor modellering van de vraag vanwege het lage aantal parameters, flexibiliteit van de functionele vorm en eenvoudige interpretatie op basis van een nutsfunctie. De foutterm van het logitmodel geeft de consumentenvoorkeuren weer die de differentiatie tussen beide producten veroorzaken. Onze specificatie van het logitmodel gebruikt twee parameters: de constante en een prijsgevoeligheidsparameter. De constante is gelijk aan 5 en de prijsparameter is gelijk aan -5 voor beide producten.

21. G.J. Werden, L.M. Froeb en T.J. Tardiff, ‘The use of the logit model in applied industrial organization’, *International Journal of the Economics of Business* 1996, nr. 1, p. 83-105.

22. P. Auer, N. Cesa-Bianchi en P. Fischer, ‘Finite-time analysis of the multi-armed bandit problem’, *Machine learning* 2002, nr. 2-3, p. 235-256.

23. Letterlijk *optimistic in face of uncertainty*.

24. Q_p kan ook worden gezien als gemiddelde winst plus exploratiebonus.

25. Hansen e.a. 2020.

26. J.C. Gittins, ‘Bandit processes and dynamic allocation indices’, *Journal of the Royal Statistical Society: Series B (Methodological)* 1979, nr. 2, p. 148-164.

Contextual bandit algoritmes

Alle algoritmes maken in principe een inschatting van de verwachte winst voor elke optie die zij hebben. In het geval van UCB-algoritmes is de verwachte winst simpelweg het gemiddelde (met een toevoeging van een bovengrens) van geobserveerde winsten. *Contextual bandit* algoritmes houden ook rekening met de omgeving, zoals Figuur 2 laat zien. Zoals te zien in Figuur 4 levert de prijs van 0.4 bijvoorbeeld een hogere winst wanneer de concurrent 0.8 kiest dan wanneer hij de prijs van 0.3 zet. Context maakt dus uit. In andere woorden, het is mogelijk om te reageren op wat de concurrent heeft gedaan en de optimale prijs kan veranderen op basis van het gedrag van de concurrent.

De algoritmes kunnen de omgeving in kaart brengen door een vraagmodel te schatten. Met het model zijn ze dan in staat om de verwachte winsten voor elke combinatie van eigen prijs en de prijs van de concurrent te berekenen. Zo vullen ze eigenlijk de tabel zoals in Figuur 4 in met eigen inschattingen van de winst. Wij maken gebruik van twee statistische modellen voor de vraag: een neurale netwerk²⁷ (een heel flexibel model) en een lineair vraagmodel (anders dan de ‘ware’ vraag²⁸). De bijhorende algoritmes noemen wij *deep contextual bandit* en *least squares contextual bandit*.

Omdat de bedrijven gelijktijdig prijzen moeten zetten reageren ze op de prijs van de concurrent in de vorige periode. In die zin volgen wij o.a. de aanpak van Calvano e.a.²⁹ Elke simulatie begint met een initialisatie van vijfhonderd periodes waarin prijzen toevallig worden gekozen. Op basis van deze prijzen en afzet wordt het model voor de vraag geschat. Op basis van het model en de geobserveerde prijs in de vorige periode kiezen beide bedrijven in elke periode tegelijk de prijs die volgens het model de hoogste winst oplevert.

In het geval van de *deep contextual bandit* worden vervolgens de parameters van het neurale netwerk in elke periode geüpdatet met één stap van een *backpropagation* algoritme op basis van een zogenoemde *minibatch* van honderd toevallig geselecteerde waarnemingen. Deze aanpak komt overeen met het trainen van *deep reinforcement learning* algoritmes.³⁰

In het geval van de *least squares contextual bandit* wordt het lineaire model één keer per vijfhonderd periodes opnieuw geschat met vijfhonderd toevallig geselecteerde waarnemingen. De reden voor het grotere aantal waarnemingen dan bij *deep contextual bandit* is dat het lineaire model volledig wordt geschat.

Om exploratie te faciliteren laten wij de algoritmes in elke periode een toevallige prijs kiezen met de kans van $\exp(-0.0002t)$. De kans is dus ongeveer 90 procent meteen na initialisatie en ongeveer 13 procent na 10.000

periodes. Deze aanpak is analoog aan die van Calvano e.a.³¹ en Klein.³²

Ook bij *contextual bandit* algoritmes speelt de toekomstige strategie geen expliciete rol en wordt daarom geen verdiscontering gebruikt.

Reinforcement learning algoritmes

Tot slot gebruiken wij *Q-learning*³³ algoritmes, die het volledige *reinforcement learning* probleem oplossen zoals weergegeven in Figuur 3. *Q-learning* algoritmes vormen een verwachting van de totale winst over het hele spel met een Bellman-vergelijking,³⁴ die het dynamische optimalisatieprobleem vertaalt naar een sequentie van simpelere oplosbare problemen. De Bellman-vergelijking is nauw gerelateerd aan achterwaartse inductie, een methode voor het bepalen van *subgame perfect equilibria* in een herhaald spel. Door middel van deze vergelijking wordt een tabel van Q-waarden (‘Q’ staat voor *quality*) ingevuld voor alle combinaties van mogelijke omgevingskenmerken en beschikbare keuzes. De Q-waarde geeft een inschatting van totale winst, die een bepaalde keuze oplevert. Daar zit het verschil ten opzichte van *contextual bandit* algoritmes, die alleen de winst in de huidige periode inschatten en gebruiken voor besluitvorming. De verwachting wordt gevormd door in elke periode de Q-waarde te updaten met de som van de gerealiseerde winst en maximale Q-waarde voor de omgevingskenmerken van de volgende periode.

De Bellman-vergelijking benadert in ons geval de Q-waarde als volgt:

$$Q(s, a) = \mathbb{E}[\pi|s, a] + \gamma \mathbb{E} \left[\max_a Q(s', a) | s, a \right]$$

waar π is de winst voor de staat van de omgeving (*state of the world*), s en de actie a , γ is de discontovoet en s is de staat van de omgeving in de komende periode.

Deze Q-waarde wordt geschat door iteratieve updates in elke periode t volgens:

$$Q'(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha \left(\pi_t + \gamma \max_a Q(s_{t+1}, a) \right)$$

waar α de snelheid van leren bepaalt – hoge α geeft meer gewicht aan

$$\pi_t + \gamma \max_a Q(s_{t+1}, a)$$

ten op zichte van de huidige waarde $Q(s_t, a_t)$.

Bij het ontwerp van de algoritmes volgen wij Calvano e.a.³⁵ en specificeren de geobserveerde omgevingskenmerken als de combinatie van eigen prijs en de prijs van

27. Wij gebruiken een neurale netwerk met één *layer*, twee *neurons*, logistische activatiefunctie en lineaire output.

28. De vraag in de simulatie is gespecificeerd als een logit.

29. Calvano e.a. 2019b.

30. V. Mnih e.a., ‘Human-level control through deep reinforcement learning’, *Nature* 2015, nr. 7540, p. 529-533.

31. Calvano e.a. 2019b.

32. Klein 2018.

33. C.J. Watkins en P. Dayan, ‘Q-learning’, *Machine learning* 1992, nr. 3-4, p. 279-292.

34. R. Bellman, *Dynamic programming*, Princeton, USA: Princeton University Press 1957

35. Calvano e.a. 2019b.

de concurrent in de periode voorafgaand aan de prijszetting. Omdat er tien mogelijke prijzen zijn, hebben wij dus honderd mogelijke *states of the world*, ofwel honderd situaties die door de algoritmes worden herkend. Voor elke van de honderd situaties schat het algoritme dan een verwachte Q-waarde voor de tien beschikbare prijzen. In totaal heeft de geschatte Q-tabel honderd rijen (situaties) en tien kolommen (acties). De betreffende Q-waarde wordt alleen bijgewerkt wanneer de bijhorende combinatie van situatie en actie wordt geobserveerd. Er is dus intensief experimenteren nodig om een betrouwbare inschatting te kunnen maken van de vele Q-waarden. De bedrijven kiezen een toevallige prijs met de kans van $\exp(-0.00001t)$. De kans is dus ongeveer 99 procent meteen na duizend periodes, ongeveer 13 procent na 200.000 periodes en ongeveer 2 procent na 400.000 periodes.

Wanneer de algoritmes niet exploreren, kiezen ze de prijs met de hoogste Q-waarde aan de hand van de geobserveerde prijzen in de vorige periode. Ook hier maken de bedrijven hun keuzes simultaan en updaten de Q-waarde op basis van de verdiende winst. Bij het bepalen van de waarden van de parameters volgen wij Calvano e.a.³⁶ en zetten de snelheid van leren $\alpha = 0.01$ en de discontovoet op $\gamma = 0.95$. De initiële Q-tabel komt overeen met de werkelijke winsten (Figuur 4). De bijhorende optimale reacties zouden dus in beginsel tot competitieve uitkomsten leiden. Afwijkingen van deze uitkomst kunnen wij aan de manier van updaten van de Q-tabel toekennen.

Uitkomsten van de simulaties

Multi-armed bandit algoritmes

Wij beginnen met de simpele UCB1-tuned algoritmes die we hiervoor beschreven hebben. Wij draaien duizend simulaties en laten de algoritmes concurreren voor 10.000 periodes. Met deze algoritmes toetsen wij of enkel exploratie de algoritmes al in staat stelt om bovencompetitieve prijzen te realiseren. De algoritmes obser-

veren alleen eigen winst en eigen prijs voor elke voorafgaande periode.

Figuur 5 laat de resultaten zien van de simulaties. Het linker paneel geeft het verloop van de prijzen weer. De resultaten van alle duizend experimenten zijn samengevat door het opnemen van een gemiddelde en 10 procent en 90 procent kwantielen. De 10 procent-kwantiel lijn betekent voor een bepaalde periode dat in 10 procent van experimenten een prijs op of onder deze lijn wordt gehanteerd. Het rechter paneel bevat een histogram van de verschillende prijzen over alle experimenten. De groene gestreepte lijnen geven de competitieve prijs aan en de rode gestreepte lijnen de collusieve prijs.

De algoritmes convergeren binnen duizend periodes richting de theoretisch optimale collusieprijs van 0.9 en hebben beide een bijna identiek verloop, waardoor de lijnen in het linker paneel overlappen. Vanwege doorgaande exploratie wijken de bedrijven in een deel van de experimenten echter later rond periode 2.500 af. In deze simulaties convergeren de bedrijven naar prijzen rond 0.5.³⁷ De exploratie kan dus ook tegen een stabiel bovencompetitief evenwicht gaan.³⁸ Het histogram, waarin staat weergegeven hoe vaak (procentueel) bepaalde prijzen voorkomen, laat wel zien dat in de meeste gevallen heel hoge prijzen worden gehanteerd.

Het is opvallend dat de algoritmes zonder enige kennis over de omgeving in staat zijn om (in de meeste gevallen) op de optimale collusieprijs te komen. Resultaten zijn soortgelijk ook met drie concurrenten.³⁹ Het duurt wel zodanig lang dat het waarschijnlijk moeilijk is om zo'n scenario in veel markten in de praktijk aan te treffen. Wij kunnen het optimalisatieprobleem echter eenvoudiger maken door de keuzeset van beschikbare prijzen te beperken. Dan gaat naar verwachting de convergentie naar hogere prijzen sneller.⁴⁰

De resultaten laten zien dat exploratie op zich al tot bovencompetitieve prijzen kan leiden. Een beetje gechargeerd kan elke exploratie gezien worden als een poging om te colluderen, soortgelijk aan prijssignalering. Hansen e.a.⁴¹ laten wel zien dat de manier van exploreren van belang is – alleen gecorreleerde experimenten leiden tot hogere prijzen. Het feit dat de algoritmes geen prijzen van de concurrent observeren, zorgt ervoor dat ze de optimale reactie zoals weergegeven in Figuur 4 niet kunnen uitvoeren en ze zoeken dus naar het optimum op een ongestructureerde manier, dat het mechanisme dat tot competitieve prijzen leidt, omzeilt. Omdat er geen informatie over de omgeving nodig is, kan dit mechanisme zelfs in offline markten werken waar informatie over concurrenten niet of moeilijk beschikbaar is, bijvoorbeeld in markten waar door

36. Calvano e.a. 2019b.

37. Dit is ook zichtbaar op de histogram in Figuur 5.

38. Als wij de mate van exploratie met tijd laten dalen kunnen wij deze afwijkingen voorkomen.

39. De uitkomsten van deze simulatie zijn beschikbaar in de online appendix.

40. Dit hebben wij geverifieerd met een simulatie waar de keuzeset bestaat uit 0.3, 0.4 en 0.5. Zoals verwacht convergeren de algoritmes relatief snel naar 0.5. De uitkomsten van deze simulatie zijn beschikbaar in de online appendix.

41. Hansen e.a. 2020.

Figuur 5 Resultaten van simulaties met twee UCB1-tuned algoritmes

Figuur 6 Vergelijking van geschatte winst door deep contextual bandit algoritmes en merkelijke winst

22

middel van aanbestedingen wordt geconcurrereerd. Soortgelijke algoritmes worden ook gebruikt bijvoorbeeld in *real-time bidding* voor online advertenties.⁴²

Voor dit mechanisme is het wel cruciaal dat de vraag stabiel blijft. Als afnemers hun aankoopgedrag aanpassen aan het experimenteren van bedrijven en het kopen uitstellen in afwachting van een periode met lagere prijzen, zullen de algoritmes niet in staat zijn om te leren dat de hogere prijzen optimaal zijn. De mogelijkheid voor uitstel van aankoop en de acceptatie van de variatie in prijzen in het algemeen zal verschillen per markt. Daardoor zijn de risico's die gepaard gaan met dit mechanisme afhankelijk van het type markt. Het gebruikte algoritme bevat geen mechanisme dat stabiliteit bevordert en kan in principe uitgebuit worden door concurrenten. Voor duurzame coördinatie op hogere prijzen is dus meer nodig dan alleen exploratie. Een besef van bedrijven dat een eventuele prijsverlaging slechts tijdelijke winst oplevert, kan wel al voldoende zijn om een stabiele uitkomst te waarborgen.

Contextual bandit algoritmes

Vervolgens krijgen de algoritmes extra informatie over prijzen van concurrenten. Ze weten dus iets meer over de context waarin ze hun prijzen zetten. De gegevens

over prijzen en winsten stelt ze in staat om een model voor de vraag te schatten en de optimale prijs op basis daarvan te zetten. Wij draaien honderd experimenten van 20.000 periodes en zoals hiervoor genoemd gebruiken wij een neurale netwerk en een lineair model als twee manieren om de vraag te schatten. De twee algoritmes noemen wij respectievelijk *deep contextual bandit* en *least squares contextual bandit*.

Het neurale netwerk slaagt in het inschatten van de werkelijke relatie tussen prijs en winst. Figuur 6 vergelijkt de gemiddelde verwachting van het algoritme aan het einde van het experiment met waarden gebaseerd op de ware vraag. Wij zien nauwelijks verschil en ook de optimale reacties zijn dezelfde als voorgeschreven door de economische theorie op basis van het eenmalige spel. Dit is op zich niet verbazingwekkend omdat het doel van een model is om de werkelijkheid zo goed mogelijk in kaart te brengen. Exploratie heeft blijkbaar niet tot patronen geleid die afwijken van de verwachtingen.

De resultaten van de simulaties zijn te zien in Figuur 7. Zoals verwacht op basis van de reactiefuncties in Figuur 6 convergeren de algoritmes naar het competitieve evenwicht van 0.4.

Het is niet verbazingwekkend dat extra informatie tot een competitieve uitkomst leidt als de algoritmes geen rekening houden met de toekomst bij hun besluitvorming. De theoretische resultaten zijn namelijk afgeleid onder perfecte informatie en een simultaan eenmalig prijszettingsspel. De resultaten laten zien dat de

42. G. Jauvion, N. Grislain, P. Dkengne Sielenou, A. Garivier en S. Gerchinovitz, 'Optimization of a SSP's header bidding strategy using Thompson sampling', in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 2018, p. 425-432.

Figuur 7 Resultaten van simulaties met deep contextual bandit algoritmes

Figuur 8 Geschatte winst door least squares contextual bandit algoritmes

exploratie niet voldoende is om de schattingen zodanig te beïnvloeden dat er afgeweken wordt van het competitieve evenwicht. In het geval van de simpele *multi-armed bandit* algoritmes werkte exploratie wel anti-competitief omdat de geschatte winst voor elke beschikbare actie apart wordt bepaald. Het model legt een structuur op aan de verwachte winsten en het lijkt dat de structuur moeizaam te veranderen is als de exploratie geen overduidelijke patronen genereert, die afwijken van het geschatte verband.

Wij hebben soortgelijke experimenten ook uitgevoerd met een minder flexibel lineair model. Dit model wijkt dus af van de werkelijkheid en het is mogelijk dat het foute inschattingen oplevert. Figuur 8 laat zien dat de geschatte winsten behoorlijk afwijken van de werkelijke winsten en ook de optimale reacties verschuiven richting hogere prijzen. De getoonde reactiefuncties kunnen tot twee evenwichten leiden, 0.5 of 0.6.⁴³ Een mogelijke verklaring voor de verschuiving richting hogere prijzen is dat de functionele vorm van het model minder flexibel is dan de werkelijke vraagfunctie waardoor de verschillen tussen verschillende prijzen per definitie kleiner zijn. In Figuur 7 zien wij bijvoorbeeld in elke rij grotere spreiding in winsten dan in Figuur 8.

Figuur 9 presenteert de resultaten van de simulaties. Zoals wij kunnen verwachten op basis van Figuur 8,

convergeren de algoritmes naar bovencompetitieve prijzen van 0.5 of 0.6. Hoewel het niet om optimale collusieprijsen gaat, leveren deze prijzen wel minder consumentensurplus op. De prijzen van 0.5 en 0.6 leveren voor de bedrijven respectievelijk ongeveer 70 en 81 procent van monopoliewinsten.

De uitkomsten van deze subparagraaf laten voornamelijk zien dat informatie over prijzen van de concurrent de afwijkmogelijkheden voor bedrijven zichtbaar maakt, wat tot competitieve prijzen leidt. Het voorbeeld van *least squares contextual bandit* algoritmes toont echter aan dat bepaalde modellen met foute specificatie een neiging kunnen hebben om hogere of lagere prijzen te hanteren. De richting van deze afwijking is vooraf niet in te schatten maar men kan zich voorstellen dat er een risico kan ontstaan als concurrenten technieken gebruiken met dezelfde neiging. De coördinatie op technieken kan tot stand komen door publieke uitspraken over *best practices* in publicaties en blogs of op conferenties, door advies van consultants of het aanschaffen van dezelfde software.

De resultaten kunnen natuurlijk sterk veranderen als de algoritmes tegen concurrenten met andere strategieën spelen. Omdat er talloze denkbare strategieën zijn, laten we deze in dit artikel buiten beschouwing. Combinaties van lerende algoritmes met bepaalde vuistregels zoals volggedrag kan een onderwerp zijn voor verder onderzoek. Algoritmes die een ingebouwde neiging hebben om collusiemogelijkheden op te zoeken, vallen ook

43. Deze zijn een evenwicht omdat er op basis van de geschatte winsten niet van wordt afgeweken.

Figuur 9 Resultaten van simulaties met least squares contextual bandit algoritmes

Figuur 10 Resultaten van simulaties met Q-learning algoritmes

onder deze categorie. Wij richten ons nu echter op algoritmes die van tevoren als ‘onschuldig’ kunnen worden aangemerkt.

Reinforcement learning algoritmes

Tot slot draaien we simulaties met Q-learning algoritmes. Wij gebruiken de theoretische winsten uit Figuur 4 als een startpunt voor een Q-tabel. Dat betekent dat ze beginnen met reactiefuncties, die tot een competitief evenwicht leiden. Exploratie en de vooruitkijkende aard van Q-learning algoritmes kan deze uitkomst echter veranderen.

De resultaten zijn vergelijkbaar met die van Calvano e.a. en Klein⁴⁴ met prijzen ongeveer in het midden tussen het competitieve evenwicht (0.4) en de monopolieprijs (0.8). De resultaten zijn weergegeven in Figuur 10. De bedrijven convergeren het vaakst naar de prijs van 0.6 maar de prijzen zijn vrijwel altijd boven het competitieve niveau zoals wij kunnen zien aan de kwantilen en het histogram. Calvano e.a.⁴⁵ vinden bovendien dat coördinatie op hogere prijzen ook plaatsvindt met meerdere spelers en verschillende waarden voor de discontovoet en de snelheid van leren.

De Q-tabel bevat duizend elementen en daarom zijn heel veel iteraties nodig voor betrouwbare exploratie. Het lijkt aannemelijk dat bedrijven in veel markten het

zich niet zouden kunnen veroorloven om 500.000 periodes te experimenteren. Met een andere opzet van algoritmes (bijv. kleinere keuzeset⁴⁶) of een andere exploratiestrategie zou de convergentie sneller kunnen zijn, maar de coördinatie lijkt in ieder geval niet makkelijk te bereiken. Relevant is ook dat de lengte van een periode in de praktijk niet alleen wordt bepaald door het vermogen om prijzen te veranderen (dat kan tegenwoordig in veel markten snel) maar ook door de kans om binnen een periode te leren over de reactie van de vraag. Ook als bedrijven hun prijzen elke seconde kunnen veranderen, is het vaak onrealistisch dat ze elke seconde ook kunnen zien hoe afnemers op de prijsveranderingen reageren.

Om bovenstaande redenen suggereren Calvano e.a.⁴⁷ dat de algoritmes in een simulatie getraind zouden kunnen worden en pas daarna worden ingezet voor ‘live’ prijszetting in reële markten. Het algoritme zou dan gaan prijzen op basis van een Q-tabel die is geschat in een simulatie. Wij vinden echter dat de strategieën afkomstig van de simulaties niet eenvoudig te gebruiken zijn voor echte prijszetting. De potentiële strategieën die logisch lijken en dus zouden kunnen worden ingezet leiden dan niet tot bovencompetitieve prijzen.⁴⁸ Hieron-

44. Calvano e.a. 2019b en Klein 2018.

45. Calvano e.a. 2019b.

46. Met een kleinere keuzeset is de convergentie inderdaad sneller. Zie de online appendix.

47. Calvano e.a. 2019b.

48. Denk bijvoorbeeld aan strategieën die volgedrag voorschrijven.

der gaan wij in op de redenen waarom wij vinden dat de kans dat bovenstaande uitkomst in reële markten voorkomt, beperkt wordt door enkele praktische beperkingen. De volgende paragraaf geeft iets meer intuïtie over het potentieel gebruik van *reinforcement learning* algoritmes voor prijszetting in de praktijk.

Manier van toepassing van reinforcement learning algoritmes in de praktijk

Algoritmes leren verschillende strategieën

In elke simulatie bepalen beide algoritmes hun Q-tabel, die de strategie bepaalt. Wij kunnen de Q-tabel observeren en de geïmpliceerde strategieën analyseren. Een belangrijke bevinding is dat de twee algoritmes in dezelfde simulatie tot verschillende inschattingen van de Q-tabel komen. Dat betekent dat de bedrijven verschillende strategieën voeren en daarom is het onduidelijk welke van de twee strategieën in de praktijk zou moeten worden gebruikt. De gemiddelde correlatie tussen optimale reacties in de honderd simulaties is 0.17.

Dit verschil in strategieën maakt het moeilijk of zelfs onmogelijk om van de simulatieresultaten een zinvolle strategie af te leiden voor de prijszetting in de reële markten. Het feit dat de algoritmes toch convergeren naar gezamenlijk evenwicht betekent dat de strategieën in bepaald opzicht complementair aan elkaar zijn. Het ontwikkelen van zulke strategieën is alleen mogelijk als beide algoritmes tegelijkertijd leren en elkaars acties gedurende het hele proces in de strategievorming meenemen. Zoals hiervoor genoemd kan dit in de praktijk moeilijk zijn vanwege het grote aantal benodigde iteraties.

Strategieën die collusievorming bevorderen alleen rationeel met complementaire strategie van de concurrent

Wij kunnen ons voorstellen dat met een andere simulatieomgeving of andere algoritmes het mogelijk is dat er consistente collusieve Q-tabellen ontstaan. Onder een

aanname dat vóór de introductie van de algoritmes er competitief wordt geprijsd is het voor het bereiken van hogere prijzen nodig dat de geleerde strategie een prijsverhoging bevat als optimale reactie op competitief evenwicht. Zo'n strategie is echter alleen rationeel als het bedrijf gelooft dat de concurrent meegaat. De beslissing over het al dan niet gebruiken van de geschatte Q-tabel zal door mensen worden gemaakt. Als deze menselijke beslismakers vinden dat de concurrent gaat volgen, kunnen zij de prijzen winstgevend verhogen ook zonder een algoritme. Als ze deze verwachting niet hebben, lijkt het onzinnig om een algoritme met de betreffende prijsstrategie gaan inzetten. Dit wordt ook opgemerkt door Deng.⁴⁹

Om te illustreren hoe de complementaire strategieën eruit kunnen zien, laten wij in Figuur 11 delen van Q-tabellen zien van één concrete simulatie. In dit experiment convergeren de algoritmes naar een prijs van 0.6. Wij laten dus delen van de tabel zien die tussen de competitieve prijs (0.4) en de uitkomst (0.6) zitten. Wij zien dat de geschatte winsten en bijhorende optimale reacties behoorlijk verschillen tussen de twee bedrijven. De optimale reactie op competitief evenwicht van Bedrijf 1 is bijvoorbeeld 0.4 terwijl Bedrijf 2 de prijs naar 0.5 verhoogt. Het is dus niet duidelijk welke van de twee Q-tabellen gebruikt zouden moeten worden in de prijszetting. De prijsverhoging gehanteerd door Bedrijf 2 is alleen winstgevend als Bedrijf 1 volgt. Dat is het geval in de simulatie maar in de reële markt zou dit alleen onder heel specifieke omstandigheden werken.

De Q-tabellen uit Figuur 11 geven aan hoe de uitkomst is bereikt. Het resulterende pad naar het evenwicht is weergegeven in Figuur 12. Wij kunnen zien dat de strategieën elementen vertonen die vorming of stabiliteit van stilzwijgende collusie zouden kunnen bevorderen, zoals het hanteren van prijsverhogingen als optimale reacties (vorming) of volgen van prijsverhogingen door de concurrent (stabiliteit). Wij kunnen echter niet argumenteren dat er een bovencompetitieve uitkomst zou resulteren als een van de algoritmes tegen een onbekende concurrent zou gaan spelen.

Combinatie van volggedrag en exploratie kan tot hogere prijzen leiden

Zoals net besproken kent het gebruik van strategieën die prijsverhogingen voorschrijven op basis van simulaties, belangrijke beperkingen. Algoritmes kunnen echter ook leren om te volgen en zo'n strategie kan ook in de praktijk worden ingezet. Dit kan op zich al de stabiliteit van een bovencompetitief evenwicht bevorderen. Het bereiken van hoge prijzen wordt echter door volggedrag niet in gang gezet. Als er echter ook tijdens de prijszetting in de echte markt wordt geëxploreerd is het voor een lerend algoritme mogelijk om het volggedrag van de concurrent te herkennen en de eigen strategie op basis daarvan aan te passen. Het is cruciaal dat een

49. A. Deng, 'How Concerned Should We Be About Algorithmic Tacit Collusion?', Comments on Calvano et al., 11 oktober 2019.

Figuur 11 Voorbeeld van (deel van) geschatte Q -tabellen

Figuur 12 Voorbeeld van strategieën gebaseerd op geschatte Q -tabellen

ontwikkelde prijsverhogende strategie gebaseerd is op het gedrag van de concurrent in de reële markt en niet op simulaties.

Conclusies

We hebben drie mechanismes geïdentificeerd, die het voor zelflerende algoritmes mogelijk maken om hoge prijzen te genereren. Deze mechanismes zijn onderdeel van algemene ‘onschuldige’ zelflerende algoritmes en zijn dus niet vooraf ingebouwd om collusie te faciliteren. Ten eerste leren alle zelflerende algoritmes door middel van exploratie. Zonder exploratie kunnen ze niet als (volledig) autonoom worden gezien. De exploratie is willekeurig en is vooral een technisch hulpmiddel om een optimalisatieprobleem op te lossen. Zo kunnen echter sommige economische mechanismes worden omzeild. Wij laten zien dat simpele UCB1-tuned algoritmes in staat zijn om de optimale collusieprijs te vinden.

Het tweede mechanisme maakt gebruik van extra informatie over de omgeving. Onze simulaties met *contextual bandit* algoritmes hebben geleid tot competitieve uitkomsten wanneer de vraag juist wordt geschat en optimale reacties overeenkomen met de economische theorie. Incorrecte schatting van de vraag kan echter tot

hogere of lagere prijzen leiden. Een potentiële coördinatie op gebruik van technieken met neiging tot hogere prijzen kan schadelijk zijn voor de concurrentie. Zulke coördinatie kan ontstaan door het afnemen van dezelfde software, advies van consultants aan meerdere concurrenten of communicatie over *best practices* in technische publicaties, blogs of presentaties. Het is ook denkbaar dat de informatie over prijzen van de concurrent gebruikt wordt voor het ontwikkelen van volgstrategieën die stabiliteit van collusie bevorderen. Deze gegevens kunnen ook nuttig zijn voor het ontwikkelen van prijsalgoritmes met een ingebouwde neiging tot collusie. Zulke algoritmes kunnen een interessant onderwerp zijn voor verder onderzoek.

Het derde mechanisme betreft de vooruitkijkende aard van de strategieën ontwikkeld door *reinforcement learning* algoritmes. Deze algoritmes houden expliciet rekening met gevolgen van hun acties voor toekomstige prijzen en winsten. Zo kunnen ze leren om de korte termijn optimale reactie voorgeschreven door de *best-response* te ontwijken als ze kunnen observeren dat de afwijkingen van hogere prijzen alleen maar tot prijsoorlogen leiden, die op lange termijn niet winstgevender zijn dan het blijven in het collusieve evenwicht. Exploratie blijft cruciaal zodat de verschillende prijsscenario's met elkaar kunnen worden vergeleken. Wij vinden in onze simulaties dat simpele *Q-learning* algoritmes inderdaad in staat zijn om naar bovencompetitieve prijzen te convergeren.

Het begrijpen van de mechanismes stelt ons in staat om te beoordelen hoe waarschijnlijk het is dat deze mechanismes zich voordoen in de praktijk. De *Q-learning* algoritmes hebben bijvoorbeeld honderdduizenden periodes nodig voor convergentie. Dat lijkt in vele markten onrealistisch. Wij laten bovendien zien dat de geleerde strategieën zich niet lenen voor gebruik in reële markten. Strategieën die volgedrag voorschrijven, kunnen de stabiliteit van een potentieel evenwicht met hogere prijzen bevorderen. De vorming van collusie wordt echter alleen bereikt door synchroon leren en het ontwikkelen van complementaire strategieën.

Exploratie op kleinere schaal (bijvoorbeeld) met een beperkte set beschikbare prijzen zou wel tot hogere prijzen kunnen leiden. Wel is het noodzakelijk dat beide concurrenten op een soortgelijke manier exploreren of dat een van de concurrenten een strategie volgt die stabiliteit van collusie bevordert (bijvoorbeeld *price matching*). Experimenteren kan kostbaar zijn en negatieve reputatie-effecten creëren wanneer afnemers een afkeur hebben voor dergelijke frequente prijsveranderingen. Tot slot, als afnemers de exploratie herkennen voor wat het is, kunnen ze hun aankoop uitstellen bij hogere prijzen waardoor de algoritmes niet kunnen leren dat deze prijzen tot hogere winsten leiden.

Gelet op het bovenstaande concluderen wij dat lerende algoritmes theoretisch in staat zijn om bovencompetitieve evenwichten te bereiken, maar dat er aanzienlijke praktische belemmeringen bestaan die de risico's in reële markten aanzienlijk kleiner maken. Collusie door 'onschuldige' zelflerende algoritmes lijkt alleen mogelijk onder specifieke omstandigheden. Daar komt nog bij dat een andere vraagspecificatie of groter aantal concurrenten het bereiken van hogere prijzen verder kan bemoeilijken. Het toenemende gebruik van prijsalgoritmes vergroot wel het risico op directe of indirecte informatie-uitwisseling over technieken voor prijszetting omdat de mededingingsbeperkende strekking van zulke informatie-uitwisseling minder duidelijk is dan in geval van communicatie over de prijsstrategie zelf. De technieken kunnen ook kenbaar worden gemaakt door middel van *whitepapers* of optredens op conferenties.⁵⁰ Bedrijven kunnen ook prijsalgoritmes afnemen van consultants of softwareontwikkelaars. Met coördinatie op het gebruik van bepaalde technieken is de kans op collusieve uitkomsten substantieel groter. Het experimenteren met de prijszetting maakt het vinden van bovencompetitieve evenwichten ook mogelijk.

Hoewel wij aanzienlijke praktische belemmeringen hebben gevonden voor het ontstaan van stilzwijgende algoritmische collusie, is het van belang dat toezichthouders inzicht hebben in de werking van de algoritmes en daarin ingebouwde mechanismes, die tot onwenselijke uitkomsten kunnen leiden. Aan deze kennisontwikkeling willen wij met dit artikel bijdragen. De inzichten

kunnen ook in een andere context dan prijszetting nuttig zijn. De *Q-learning* algoritmes kunnen bijvoorbeeld dankzij hun nauwe samenhang met speltheorie gebruikt worden om inzicht te krijgen in uitkomsten van complexe situaties, die analytisch niet oplosbaar zijn.

50. Illustratief voorbeeld: A. Derakhshan, F. Hammer en Y. Demazeau, 'PriceCast Fuel: Agent Based Fuel Pricing', in: *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Cham: Springer 2016, p. 247-250.