README for
Johnson, Rhodes, and Wildenbeest
"Platform Design when Sellers Use Pricing Algorithms"

Overview

The code in this replication package constructs the output tables and figures reported in Johnson, Rhodes, and Wildenbeest (2023): "Platform Design when Sellers Use Pricing Algorithms" from simulated data using FORTRAN.

Data Availability Statements

This paper does not involve analysis of external data, but only uses data that are generated by the authors via simulation in their code.

Description of Code

The file `qlearning.f90` contains the FORTRAN program that is needed to simulate the data that is required to reproduce the tables and figures in the paper. Before it can be used, it needs to be compiled. We have used GFortran to compile the code (GNU Fortran (GCC) 9.2.0).[1] To compile, use (in the command line):[2]

```
gfortran -o qlearning qlearning.f90
```

This creates a compiled file called `qlearning`, which can be used to run the simulations. The program uses parameter values as arguments that are supplied using the command line. The specific syntax needed to run the program is

```
./qlearning n mu advstate rndshow taupr ah showtwo rmax gridstep seq grid
```

where `./qlearning` calls the main program and the rest of the syntax are arguments that indicate specific parameter values. The arguments are:

`n`	number of firms n
`mu`	product differentiation parameter μ
`advstate`	indicator for whether dynamic "advantage" is a state variable as well
`rndshow`	indicator for whether buy-box is random
`taupr`	probability of losing advantage if other DPDP conditions are satisfied
`ah`	demand shock (set to non-zero in case of stochastic demand)
`showtwo`	indicator for whether two firms are shown in buy-box (only use in case of three firms)
`rmax`	number of repetitions
`gridstep`	number of steps in the grid (set to 0 if all are used)
`seq`	where to start in grid (in terms of gridsteps)
`grid`	name of the grid txt file

The grid file contains parameter values for variables that might change when constructing a specific table or figure. We use eight different grid files—each grid file contains up to 20,000 rows that contain the parameter values for the following variables:

`alpha`	learning rate α
`beta`	experimentation rate β
`gamma`	proportion γ of consumers shown $k < n$ products
`adv`	pricing advantage ADV
`delta`	discount factor δ

[1] See https://gcc.gnu.org/wiki/GFortranBinariesMacOS for instructions on how to install GFortran on a Mac.

[2] The code can also be compiled in parallel (using OpenMP) by adding the option `-fopenmp`

As an example, suppose there are $n = 2$ firms, a product differentiation parameter of $\mu = 0.25$, learning parameters $\alpha = 0.15$ and $\beta = 10^{-5}$, parameter $\gamma = 0.70$ (which corresponds to row 71 in `grid1.txt`), pricing advantage $ADV = 0$, and discount factor $\delta = 0.95$. To run this case a 1000 times (`rmax=1000`), use

```
./qlearning 2 0.25 0 0 0 0 0 1000 1 71 grid1
```

To run all the simulations that are needed to construct Figure 1 (i.e., all rows of `grid1.txt`, so $\gamma = 0.00$ to $\gamma = 1.00$ in increments of 0.01), use

```
./qlearning 2 0.25 0 0 0 0 0 1000 0 1 grid1
```

Note that the code uses the RAN2 function that is part of the replication code of Calvano, Calzolari, Denicolo, and Pastorello (2020) to generate random numbers, which itself is a modification of the RAN2 function in Press, Teukolsky, Vetterling, and Flannery (1992).

Guide to Figures and Tables

Table 1 gives an overview of how to generate the different output files that are needed to create the figures and tables. Each output file contains the following columns (reported values in the output files are averaged across repetitions, unless otherwise stated):

column	variable	description
A	`grid`	grid line number
B	`r`	number of repetitions (`rmax`)
C	`periods`	number of periods until convergence
D	`time`	time (in seconds) until convergence
E	`notconv`	number of repetitions that have not converged
F	`n`	number of firms n
G	`advstate`	indicator for whether dynamic “advantage” is a state variable as well
H	`rndshow`	indicator for whether buy-box is random
I	`showtwo`	indicator for whether two firms are shown in buy-box
J	`taupr`	probability of losing advantage if other DPDP conditions are satisfied
K	`ah`	demand shock
L	`alpha`	learning rate α
M	`beta`	experimentation rate β
N	`gamma`	proportion γ of consumers shown $k < n$ products
O	`mu`	product differentiation parameter μ
P	`adv`	pricing advantage ADV
Q	`delta`	discount factor δ
R	`price1`	price firm 1
S	`price2`	price firm 2
T	`price3`	price firm 3 (zero if $n = 2$)
U	`pricesw`	sales-weighted price
V	`revenue`	revenue
W	`sales`	sales
X	`cs`	consumer surplus
Y	`profits`	profits firms
Z	`cyclelength`	cycle length
AA	`prc1c`	percent outcomes with cycles of length 1 (constant prices)
AB	`prc2c`	percent outcomes with cycles of length 2
AC	`prcsymm1c`	percentage symmetric for outcomes with constant prices
AD	`indprof1c`	industry profit for outcomes with constant prices
AE	`highfirm1c`	high-profit firm’s share for outcomes with constant prices
AF	`cyclelengthproper`	cycle length for outcomes with a price cycle
AG	`indprof0c`	industry profit for outcomes with a price cycle
AH	`highfirm0c`	high-profit firm’s share for outcomes with a price cycle

Table 1: Overview of the different output files

file	n	mu	advstate	rndshow	taupr	ah	showtwo	grid	output file name	figures	tables
1	2	0.25	0	0	0	0	0	grid1	`output_2_25_0_0_0_0_0_1000_1_grid1.csv`	1, 4, 5, 7, 8, 11	I, II, III, IV
2	2	0.25	0	0	0	0	0	grid_heat1	`output_2_25_0_0_0_0_0_1000_1_grid_heat1.csv`	2, 12	
3	2	0.05	0	0	0	0	0	grid1	`output_2_05_0_0_0_0_0_1000_1_grid1.csv`	3	
4	2	0.25	0	1	0	0	0	grid1	`output_2_25_0_1_0_0_0_1000_1_grid1.csv`	4	
5	2	0.25	0	0	0	0	0	grid2	`output_2_25_0_0_0_0_0_1000_1_grid2.csv`	5, 7, 11	II, III, IV
6	2	0.25	0	0	0	0	0	grid_heat2	`output_2_25_0_0_0_0_0_1000_1_grid_heat2.csv`	6, 12	
7	2	0.25	1	0	0	0	0	grid2	`output_2_25_1_0_0_0_0_1000_1_grid2.csv`	7	II, IV
8	2	0.25	0	0	0	0	0	grid3	`output_2_25_0_0_0_0_0_1000_1_grid3.csv`	8	
9	2	0.25	1	0	0	0	0	grid3	`output_2_25_1_0_0_0_0_1000_1_grid3.csv`	8	
10	2	0.25	0	0	0	0	0	grid_heat3	`output_2_25_0_0_0_0_0_1000_1_grid_heat3.csv`	9	
11	2	0.25	0	0	0	0	0	grid_heat4	`output_2_25_0_0_0_0_0_1000_1_grid_heat4.csv`	10	
12	2	0.25	0	0	0	0.25	0	grid1	`output_2_25_0_0_0_25_0_1000_1_grid1.csv`		III
13	2	0.25	0	0	0	0.25	0	grid2	`output_2_25_0_0_0_25_0_1000_1_grid2.csv`		III
14	2	0.25	0	0	0	0.50	0	grid1	`output_2_25_0_0_0_50_0_1000_1_grid1.csv`		III
15	2	0.25	0	0	0	0.50	0	grid2	`output_2_25_0_0_0_50_0_1000_1_grid2.csv`		III
16	2	0.25	0	0	0.50	0	0	grid2	`output_2_25_0_0_50_0_0_1000_1_grid2.csv`	11	
17	2	0.25	0	0	0.10	0	0	grid_heat5	`output_2_25_0_0_10_0_0_1000_1_grid_heat5.csv`	12	
18	2	0.25	0	0	0.20	0	0	grid_heat5	`output_2_25_0_0_20_0_0_1000_1_grid_heat5.csv`	12	
19	2	0.25	0	0	0.30	0	0	grid_heat5	`output_2_25_0_0_30_0_0_1000_1_grid_heat5.csv`	12	
20	2	0.25	0	0	0.40	0	0	grid_heat5	`output_2_25_0_0_40_0_0_1000_1_grid_heat5.csv`	12	
21	2	0.25	0	0	0.50	0	0	grid_heat5	`output_2_25_0_0_50_0_0_1000_1_grid_heat5.csv`	12	
22	2	0.25	0	0	0.60	0	0	grid_heat5	`output_2_25_0_0_60_0_0_1000_1_grid_heat5.csv`	12	
23	2	0.25	0	0	0.70	0	0	grid_heat5	`output_2_25_0_0_70_0_0_1000_1_grid_heat5.csv`	12	
24	2	0.25	0	0	0.80	0	0	grid_heat5	`output_2_25_0_0_80_0_0_1000_1_grid_heat5.csv`	12	
25	2	0.25	0	0	0.90	0	0	grid_heat5	`output_2_25_0_0_90_0_0_1000_1_grid_heat5.csv`	12	
26	3	0.25	0	0	0	0	0	grid1	`output_3_25_0_0_0_0_0_1000_1_grid1.csv`		V
27	3	0.25	0	0	0	0	0	grid2	`output_3_25_0_0_0_0_0_1000_1_grid2.csv`		V
28	3	0.25	1	0	0	0	0	grid2	`output_3_25_1_0_0_0_0_1000_1_grid2.csv`		V
29	3	0.25	0	0	0	0	1	grid1	`output_3_25_0_0_0_0_1_1000_1_grid1.csv`		V
30	3	0.25	0	0	0	0	1	grid2	`output_3_25_0_0_0_0_1_1000_1_grid2.csv`		V
31	3	0.25	1	0	0	0	1	grid2	`output_3_25_1_0_0_0_1_1000_1_grid2.csv`		V

Notes: For all files, rmax = 1000, gridstep = 0, and seq = 1.

Table 2 shows the output files needed for each figure and table. Note that there is an additional file with the name `benchmarks.nb` which computes Bertrand-Nash and fully collusive outcomes in Mathematica. The remainder of this section gives more detailed instructions on how to generate the figures and tables.

Table 2: Mapping of tables and figures to output files

figure/table	output files
figure 1	1
figure 2	2
figure 3	3
figure 4	1, 4
figure 5	1, 5
figure 6	6
figure 7	1, 5, 7
figure 8	1, 8, 9
figure 9	10
figure 10	11
figure 11	1, 5, 16
figure 12	2, 6, 17-25
table I	1
table II	1, 5, 7
table III	1, 5, 12-15
table IV	1, 5, 7
table V	26-31

Figure 1 uses rows 2-102 along with columns N and U (left panel), columns N and X (middle panel) and columns N and W (right panel) of output file 1.

Figure 2 uses rows 2-20001 along with columns L, M and X (left panel) and columns L, M and W (right panel) of output file 2.

Figure 3 uses rows 2-102 along with columns N and U (left panel), columns N and X (middle panel) and columns N and W (right panel) of output file 3.

Figure 4 uses rows 2-102 along with columns N and U (left panel), columns N and X (middle panel) and columns N and W (right panel) of output file 1 ("PDP" curve) and output file 4 ("Random Display" curve).

Figure 5 uses rows 2-102 along with columns N and U (left panel), columns N and X (middle panel) and columns N and W (right panel) of output file 1 ("PDP" curve) and output file 5 ("DPDP" curve).

Figure 6 uses rows 2-20001 along with columns L, M and X (left panel) and columns L, M and W (right panel) of output file 6.

Figure 7 uses rows 2-102 along with columns N and U (left panel), columns N and X (middle panel) and columns N and W (right panel) of output file 1 ("PDP" curve), output file 5 ("DPDP" curve) and output file 7 ("Smarter AI" curve).

Figure 8 uses row 2 and column X of output file 1 for the "No PDP" line. It also uses rows 3-102 along with columns P and X of output file 8 ("DPDP" curve) and output file 9 ("Smarter AI" curve).

Figure 9 uses rows 2-10101 and columns N, Q and X of output file 10. The left panel depicts the raw data. The right panel depicts, for each given (γ, δ) pair, the difference in column X between (γ, δ) and $(0, \delta)$, divided by the level of column X at $(0, \delta)$.

Figure 10 uses rows 2-10101 and columns N, Q and X of output file 11. The left panel depicts the raw data. The right panel depicts, for each given (γ, δ) pair, the difference in column X between (γ, δ) and $(0, \delta)$, divided by the level of column X at $(0, \delta)$.

Figure 11 uses rows 2-102 along with columns N and U (left panel), columns N and X (middle panel) and columns N and W (right panel) of output file 1 ("0 (PDP)" curve), output file 5 ("1 (DPDP)" curve) and output file 16 ("0.5" curve).

Figure 12 uses rows 2-20001 along with columns L, M and X (left panel) and columns L, M and W (right panel) of output files 2,6 and 17-25. Each output file corresponds to a different value of `taupr`. To illustrate, consider the left panel. For each (α, β) pair, we find the corresponding row in the output files, and then select the `taupr` value that gives the highest value for column X. Figure 12 then depicts 1-`taupr`. [Note that `taupr` is the probability of exogenously losing the advantage, whereas Figure 12 depicts the probability of keeping the advantage (subject to other DPDP conditions being met).]

Table I uses file benchmarks.nb to compute the numbers in column 1 ("Bertrand-Nash Price") and column 2 ("Collusive Price"). It also uses row 2 and column U of output file 1 for column 3 ("AI Price").

The columns in Table II use, respectively, row 2 of output file 1 ("No Intervention" column), row 72 of output file 1 ("PDP" column), row 72 of output file 5 ("DPDP" column) and row 72 of output file 7 ("Smarter AI" column). The rows in Table II then report, respectively, columns AA, AC, AD, AE, AF, AG, AH within those output files.

Table III uses output file 1 ("No Uncertainty" column, "PDP" rows), output file 5 ("No Uncertainty" column, "Dynamic PDP" rows), output file 12 ("Low Uncertainty" column, "PDP" rows), output file 13 ("Low Uncertainty" column, "Dynamic PDP" rows), output file 14 ("High Uncertainty" column, "PDP" rows), and output file 15 ("High Uncertainty" column, "Dynamic PDP" rows). The "Prices" panel computes the percentage change in moving from row 2 to row 102 in column U of the appropriate output file; the "Consumer Surplus" panel does the same but for column X; the "Total Output" panel does the same but for column W.

Table IV has two distinct parts. Start with the first two columns ("Percentage change in revenue"). The six entries compute the percentage change in moving from row 2 and column V of output file 1, to: row 72 and column V of output file 1 ("$\gamma = 0.7$" column, "PDP" row), row 102 and column V of output file 1 ("$\gamma = 1$" column, "PDP" row), row 72 and column V of output file 5 ("$\gamma = 0.7$" column, "Dynamic PDP" row), row 102 and column V of output file 5 ("$\gamma = 1$" column, "Dynamic PDP" row), row 72 and column V of output file 7 ("$\gamma = 0.7$" column, "Smarter AI" row), row 102 and column V of output file 7 ("$\gamma = 1$" column, "Smarter AI" row).

In Table IV, the final two columns ("Critical ω for higher platform payoff") use the fact that the platform's payoff (given a 20% commission) is higher due to an intervention if and only if

$$0.2\omega R^{\emptyset} + (1-\omega)\,U^{\emptyset} < 0.2\omega R + (1-\omega)\,U, \tag{1}$$

where $R^{\emptyset}$ and $U^{\emptyset}$ are respectively revenue and consumer surplus absent intervention, and R and U are respectively revenue and consumer surplus with the intervention. Row 2 and columns V and X of output file 1 give $R^{\emptyset}$ and $U^{\emptyset}$, respectively. Rows 72 and 102, and columns V and X, for output files 1, 5 and 7, give R and U for PDP, Dynamic PDP and Smarter AI for $\gamma = 0.7$ and $\gamma = 1$. It is straightforward to check that (1) cannot hold for any $\omega \in [0, 1]$ under PDP, but can hold for some ω values under Dynamic PDP and Smarter AI. In the latter two cases, rearrange (1) to find that the platform's payoff is higher with the intervention if and only if

$$\omega < \frac{U - U^{\emptyset}}{0.2(R^{\emptyset} - R) + U - U^{\emptyset}}. \tag{2}$$

The final two columns of Table IV then compute this critical ω for the different interventions, using the values for $R^{\emptyset}, U^{\emptyset}, R$ and U described above.

Table V uses row 72 along with column U ("Prices" column), column X ("Consumer Surplus" column) and column W ("Total Output" column) of various output files. In particular, the first row in Table V ("PDP") computes the percentage change in moving from the relevant cell in output file 26 to the same cell in output file 29. Similarly, the second row in Table V ("Dynamic PDP") does the same when moving from the relevant cell in output file 27 to the same cell in output file 30, while the third row in Table V ("Smarter AI") also does the same when moving from the relevant cell in output file 28 to the same cell in output file 31.

References

Calvano, E., G. Calzolari, V. Denicolò, and S. Pastorello (2020): "Artificial Intelligence, Algorithmic Pricing and Collusion," *American Economic Review*, 110(10), 3267–97.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery (1992): *Numerical Recipes in Fortran 77* (2nd edition), Cambridge University Press.