Proving the Merger Paradox

Spanos Ioannis

This presentation is a Jupiter Notebook, so that we can interactively run python code. It is accompanied by a script writen in python, linked bellow.

This file also serves as documentation for the aforementioned script and is published under the MIT licence.

Data interpretation

To calculate the units that each company produces, based on their total costs and total units produced in the market, we can solve the following equation for q.

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} * \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \frac{a - MC_1}{2 * B} \\ \frac{a - MC_2}{2 * B} \\ \frac{a - MC_3}{2 * B} \end{bmatrix}$$

However, based on our generated date, we assume that 'Q' column, can be interpreted both as the total units of production, when we try to calculate the demand curve, as well as the production of each company individually, when trying to calculate their marginal cost curves.

With that assumption, we can calculate the marginal costs for each stage of production, by calculating the first differences of the total cost array, dividing it by the first differences of the production array. Then, we run a simple OLS regression for the company's production (independent variable) and their marginal cost (dependent variable).

The only purpose that this assumption serves, is compatibility with our original data source - the example spreadsheet.

Data problem

The issue that arises from our data however, is that we have panel data with the equilibrium combinations of prices and units of production. Thus, the variations over time are due to changes to the production costs of the companies, or lateral and vertical moves of the demand.

Defining our market

We assume the Inverse Demand Curve is linear,

$$P = \mathrm{A} - \mathrm{B} * Q$$

where P is the Price and Q is the total demand.

We also assume that the first derivatives of the Cost Curves of our companies are:

$$MC_i = K_i + M_i q_i$$

Where **i** is the number of the company. In our case, in the beginning there are 3.

Calculating production levels

Best Responses

Since the companies are in a **Cournot market game**, each one of them is going to maximize its profits, by adjusting its production, according the demand and their competitors' production.

$$max\Pi_i(q_i) => MR_i = MC_i \ (1)$$

$$\dots$$

$$for\ i = 1,\ (1)\ \ => \ \ (M_1 + 2 * B) * q_1 + B * q_2 + B * q_3 = A - K_1$$

$$for\ i = 2,\ (1)\ \ => \ \ B * q_1 + (M_2 + 2 * B) * q_2 + B * q_3 = A - K_2$$

$$for\ i = 3,\ (1)\ => \ B * q_1 + B * q_2 + (M_3 + 2 * B) * q_3 = A - K_3$$

$$Or$$

$$\begin{bmatrix} (M_1 + 2 * B) & \text{B} & \text{B} \\ \text{B} & (M_2 + 2B) & \text{B} \\ B & B & (M_3 + 2B) \end{bmatrix} * \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} A - K_1 \\ A - K_2 \\ A - K_3 \end{bmatrix}$$

Merger

After two companies, i and j, merge, we need to add the two marginal cost curves horizontally:

$$q_m = q_{i(MC_i)} + q_{j(MC_j)},$$

$$where\ q_{m(MC_m)}\ is\ the\ inverse\ marginal\ cost\ curve\ of\ the\ new\ company$$

The new company, has now the following marginal cost:

$$MC_m = \frac{M_j * K_i + M_i * K_j}{M_i + M_j} + \frac{M_i * M_j}{M_i + M_j} * q_m$$

$$Or$$

$$MC_m = K_m + M_m * q_m$$

Then, the new company, and the one that wasn't included in the merger, compete in quantities again.

$$max\Pi_i(q_i) => MR_i = MC_i\ (1)$$

$$\dots$$

$$for\ i = 1,\ (1)\ => \ (M_m + 2 * B) * q_m + B * q_2 = A - K_m$$

$$for\ i = 1,\ (1)\ => \ B * q_1 + (M_2 + 2B) * q_2 = A - K_2$$

$$Or$$

$$\begin{bmatrix} (M_m + 2 * B) & \text{B} \\ \text{B} & (M_2 + 2B) \end{bmatrix} * \begin{bmatrix} q_m \\ q_2 \end{bmatrix} = \begin{bmatrix} A - K_m \\ A - K_2 \end{bmatrix}$$

Edge Cases

To reduce the complexity of the python function, that calculates the marginal cost curve of the new company, I'm working under the assumption that the demand is relatively high. The script does not work in cases where the demand is so low that the company is better off using only one of the two facilities at its disposal.

Furthermore, if one of the companies has a constant marginal cost, and the other one has a linear marginal cost, the script terminates. In cases like that, its likely that the new company manufactures only in the facilities with the constant marginal cost. Still, a very high constant marginal cost could be suboptimal, compared to a low, yet variable marginal cost.

These checks can't be done in the `merge_companies()` function without increasing the complexity of the `Company` object. Then, for every 2 companies that have merged, with that specific combination of marginal costs, our calculations would have to increase exponentially.

To manually find if the company is going to produce in both facilities or not, calculate q_m, solve for MC_m and run the function `set_cournot_production(demand, companies)` for all 3 possible MC_m curves. The resulting company with the highest profits, has the optimal MC parameters for the current market.

Cournot market game for N companies

We observe that the matrix that solves for the production units of each company, follows a clear pattern.

On the left side, the diagonal, is

$$(MC_i + 2 * B),\ where\ i = 1, 2, 3 \ldots N,\ for\ N\ companies$$

On the right side,

$$(A - K_i),\ where\ i = 1, 2, 3 \ldots N,\ for\ N\ companies$$

In order to create a function *(in Python)* to calculate the units of production for any number of companies we:

1. Create an N x N matrix, X, where every element is *B*, the slope of the inverse demand curve
2. Two N x 1 arrays that are composed of the elements mentioned above,

$$X : \begin{bmatrix} B\,B\,B\ldots B \\ B\,B\,B\ldots B \\ \ldots\ \ldots\ \ldots \\ B\,B\,B\ldots B \end{bmatrix},\ D : \begin{bmatrix} MC_1 + 2 * B \\ MC_2 + 2 * B \\ MC_3 + 2 * B \\ \ldots \\ MC_N + 2 * B \end{bmatrix}\ and\, U : \begin{bmatrix} (A - K_1) \\ (A - K_2) \\ (A - K_3) \\ \ldots \\ (A - K_N) \end{bmatrix}$$

3. Replace the diagonal of matrix X, with matrix D to create matrix H.

4. Finally, if we solve $H * q = U$ for q, we get the production units for every company competing in the market.

Simulations

Let's start with 3 companies with the following marginal cost curves:

Company 1: $MC_1\ =\ 2.71\ +\ 5.34 * q_1,$

Company 2: $MC_2 \;=\; 6.13 \;+\; 1.11 * q_2$,

Company 3: $MC_3 \;=\; 4.75 \;+\; 1.53 * q_3$

With an inverse demand curve : $P = 2221.08 - 15.81 * Q$

In [4]:

```
from cournot import *
D = (2221.08, 15.81)
companies: CompanyList = [Company(2.71, 5.34),
                          Company(6.13, 1.11),
                          Company(4.75, 1.53)]

companies = set_cournot_production(D, companies)
quantity = sum([comp.production for comp in companies])
price = calculate_price(quantity, D)

market_stats_dump(companies, quantity, price)
print(f"HHI:{hhi(companies)}")
```

```
Company 4 with Mc = 2.71 + 5.34 * q
        Produces 29.25  with €13529.37 profit.

Company 5 with Mc = 6.13 + 1.11 * q
        Produces 36.36  with €20906.58 profit.

Company 6 with Mc = 4.75 + 1.53 * q
        Produces 35.56  with €19995.47 profit.

Total production is 101.18 units @ €621.41.
HHI:3363
```

In [5]:

```
for combination in [(0, 1), (0, 2), (1, 2)]:
    #print(("*" * 60 + "\n"))
    post_merge = merge_two(D, companies, combination)
    new_quantity = sum([comp.production for comp in post_merge])
    new_price = calculate_price(new_quantity, D)

    old_profits=sum([companies[combination[0]].profits(price),
                     companies[combination[1]].profits(price)])

    print(f"The sum of the profits, of companies {companies[combination[
          f"and {companies[combination[1]].name}\n \t before the merger,
          f"€{round(old_profits,2)}\n")
    market_stats_dump(post_merge, quantity, new_price)
    print(f"HHI:{hhi(post_merge)}")
    print(f"The new price is {round(((new_price-price)*100)/price)}% hig
    print(("\n" + "*" * 60 + "\n"))
```

```
The sum of the profits, of companies 4 and 5
         before the merger, where: €34435.96

Company 4&5 with Mc = 5.54 + 0.92 * q
        Produces 46.34  with €33954.67 profit.

Company 6 with Mc = 4.75 + 1.53 * q
        Produces 44.76  with €31668.42 profit.

Total production is 101.18 units @ €780.81.
```

```
HHI:5002
The new price is 26.0% higher.

************************************************************

The sum of the profits, of companies 4 and 6
         before the merger, where: €38063.87

Company 4&6 with Mc = 4.3 + 1.19 * q
        Produces 45.56  with €32817.94 profit.

Company 5 with Mc = 6.13 + 1.11 * q
        Produces 45.67  with €32969.39 profit.

Total production is 101.18 units @ €778.79.
HHI:5000
The new price is 25.0% higher.

************************************************************

The sum of the profits, of companies 5 and 6
         before the merger, where: €50317.11

Company 5&6 with Mc = 5.55 + 0.64 * q
        Produces 49.67  with €39004.7 profit.

Company 4 with Mc = 2.71 + 5.34 * q
        Produces 38.77  with €23769.22 profit.

Total production is 101.18 units @ €822.78.
HHI:5076
The new price is 32.0% higher.

************************************************************
```

Merger Paradox

As we simulate the mergers of two companies, by adding their $q_{(mc)}$ horizontally, we observe that the resulting companies, produce fewer units. The competing companies are now fewer, thus the HHI index increases after the merger. The new equilibrium is closer to the equilibrium in a monopoly. However, the profits of the newly created company are less than the sum of the profits of the companies that merged.

The company that benefits from the merger, is the one that did not take part in it. This happens because both its market share, and the market price, increase.

The conclusion is that neither the consumers, nor the companies that took part in the merger, benefit from the merger. The only beneficiary is the company that did not take part in the merger.

Extras

Consecutive mergers - Simulation #2

Now, lets try a market composed of more companies, since we are able to add as many companies as we want in the simulation.

In [6]:

```
reset_names()
D = (2221.08, 15.81)
C: CompanyList = [Company(2.71, 5.34),
                  Company(6.13, 1.11),
                  Company(4.75, 1.53),
                  Company(1, 3.4),
                  Company(4, 2),
                  Company(5, 1.6),
                  Company(4, 2.2)]
companies = set_cournot_production(D, C)

quantity = sum([comp.production for comp in companies])
price = calculate_price(quantity, D)

market_stats_dump(companies, quantity, price)
print(f"HHI:{hhi(companies)}")
```

```
Company 1 with Mc = 2.71 + 5.34 * q
        Produces 14.85  with €3484.77 profit.

Company 2 with Mc = 6.13 + 1.11 * q
        Produces 18.36  with €5326.99 profit.

Company 3 with Mc = 4.75 + 1.53 * q
        Produces 17.99  with €5117.23 profit.

Company 4 with Mc = 1.0 + 3.4 * q
        Produces 16.43  with €4270.29 profit.

Company 5 with Mc = 4.0 + 2.0 * q
        Produces 17.56  with €4874.07 profit.

Company 6 with Mc = 5.0 + 1.6 * q
        Produces 17.9  with €5068.04 profit.

Company 7 with Mc = 4.0 + 2.2 * q
        Produces 17.36  with €4766.41 profit.

Total production is 120.45 units @ €316.71.
HHI:1435
```

In [7]:

```
new_price, post_merge = consecutive_merger(price,companies,(0, 1),D)
```

```
The sum of the profits, of companies 1 and 2
        before the merger, were: €8811.76

Company 1&2 with Mc = 5.54 + 0.92 * q
        Produces 20.77  with €6822.04 profit.

Company 3 with Mc = 4.75 + 1.53 * q
        Produces 20.09  with €6378.68 profit.

Company 4 with Mc = 1.0 + 3.4 * q
        Produces 18.33  with €5309.77 profit.
```

```
Company 5 with Mc = 4.0 + 2.0 * q
        Produces 19.6  with €6072.53 profit.

Company 6 with Mc = 5.0 + 1.6 * q
        Produces 19.99  with €6318.41 profit.

Company 7 with Mc = 4.0 + 2.2 * q
        Produces 19.38  with €5938.4 profit.

Total production is 118.16 units @ €353.05.
HHI:1669
The new price is 11.0% higher.

************************************************************
```

In [8]:

```
new_price, post_merge = consecutive_merger(new_price,post_merge,(0, 1),[
```

```
The sum of the profits, of companies 1&2 and 3
        before the merger, were: €13200.72

Company 1&2&3 with Mc = 5.24 + 0.57 * q
        Produces 24.7  with €9642.67 profit.

Company 4 with Mc = 1.0 + 3.4 * q
        Produces 21.28  with €7162.31 profit.

Company 5 with Mc = 4.0 + 2.0 * q
        Produces 22.79  with €8210.76 profit.

Company 6 with Mc = 5.0 + 1.6 * q
        Produces 23.26  with €8550.1 profit.

Company 7 with Mc = 4.0 + 2.2 * q
        Produces 22.54  with €8029.42 profit.

Total production is 114.56 units @ €409.87.
HHI:2005
The new price is 16.0% higher.

************************************************************
```

In [9]:

```
new_price, post_merge = consecutive_merger(new_price,post_merge,(0, 1),[
```

```
The sum of the profits, of companies 1&2&3 and 4
        before the merger, were: €16804.98

Company 1&2&3&4 with Mc = 4.63 + 0.49 * q
        Produces 29.27  with €13545.94 profit.

Company 5 with Mc = 4.0 + 2.0 * q
        Produces 26.83  with €11378.06 profit.

Company 6 with Mc = 5.0 + 1.6 * q
        Produces 27.39  with €11857.11 profit.

Company 7 with Mc = 4.0 + 2.2 * q
        Produces 26.53  with €11126.76 profit.

Total production is 110.01 units @ €481.78.
HHI:2504
```

```
The new price is 18.0% higher.

*************************************************************
```

In [10]:

```
new_price, post_merge = consecutive_merger(new_price,post_merge,(0, 1),[
```

```
The sum of the profits, of companies 1&2&3&4 and 5
        before the merger, were: €24924.01

Company 1&2&3&4&5 with Mc = 4.51 + 0.39 * q
        Produces 36.36  with €20906.78 profit.

Company 6 with Mc = 5.0 + 1.6 * q
        Produces 33.82  with €18081.11 profit.

Company 7 with Mc = 4.0 + 2.2 * q
        Produces 32.75  with €16953.88 profit.

Total production is 102.93 units @ €593.77.
HHI:3340
The new price is 23.0% higher.

*************************************************************
```

In [11]:

```
new_price, post_merge = consecutive_merger(new_price,post_merge,(0, 1),[
```

```
The sum of the profits, of companies 1&2&3&4&5 and 6
        before the merger, were: €38987.89

Company 1&2&3&4&5&6 with Mc = 4.6 + 0.32 * q
        Produces 48.08  with €36541.23 profit.

Company 7 with Mc = 4.0 + 2.2 * q
        Produces 43.08  with €29343.08 profit.

Total production is 91.16 units @ €779.89.
HHI:5015
The new price is 31.0% higher.

*************************************************************
```

Table of the total profits

Companies	Not merged	Merged
1, 2	8811.76	6822.04
1, 2, 3	13928.99	9642.67
1, 2, 3, 4	18199.28	13545.94
1, 2, ..., 5	23073.35	20906.78
1, 2, ..., 6	28141.39	36541.23

Only after 5 consecutive mergers did we see an increase in the profitability compared to the pre-merge conditions. The new company, named " `1&2&3&4&5&6` " in the above code-block, has a €36541.23 profit. This happens because we have much less competition, with an HHI index that's three times higher than before. The price is more than two times higher, and the production is 25% lower.

Furthermore, the mergers that include less than 6 companies, are not profitable, hence the companies would rather compete than merge.

In a real market however, a merger like that would create a huge dead weight loss, and such a price increase, that no committee would ever allow such a merger to take place.

Non symmetrical costs - Simulation #3

In [12]:

```
reset_names()
D = (2221.08, 15.81)
C: CompanyList = [Company(26.71, 8.34),
                  Company(4, 2),
                  Company(4.1, 2.2)]
companies = set_cournot_production(D, C)
quantity = sum([comp.production for comp in companies])
price = calculate_price(quantity, D)

market_stats_dump(companies, quantity, price)
print(f"HHI:{hhi(companies)}")
```

```
Company 1 with Mc = 26.71 + 8.34 * q
        Produces 26.08  with €10755.49 profit.

Company 2 with Mc = 4.0 + 2.0 * q
        Produces 36.64  with €21227.61 profit.

Company 3 with Mc = 4.1 + 2.2 * q
        Produces 36.23  with €20752.41 profit.

Total production is 98.95 units @ €656.6.
HHI:3406
```

In [13]:

```
for combination in [(0, 1), (0, 2), (1, 2)]:
    post_merge = merge_two(D, companies, combination)
    new_quantity = sum([comp.production for comp in post_merge])
    new_price = calculate_price(new_quantity, D)

    i, j = combination[0], combination[1]
    old_profits=sum([companies[i].profits(price),
                     companies[j].profits(price)])

    print(f"The sum of the profits, of companies {companies[i].name}",
          f"and {companies[j].name}\n \t before the merger, where:",
          f"€{round(old_profits,2)}\n")
    market_stats_dump(post_merge, quantity, new_price)
    print(f"HHI:{hhi(post_merge)}")
    print(f"The new price is {round(((new_price-price)*100)/price)}% hig
    print(("\n" + "*" * 60 + "\n"))
```

```
The sum of the profits, of companies 1 and 2
         before the merger, where: €31983.1
```

```
Company 1&2 with Mc = 8.39 + 1.61 * q
        Produces 45.52  with €32757.21 profit.

Company 3 with Mc = 4.1 + 2.2 * q
        Produces 44.27  with €30990.0 profit.

Total production is 98.95 units @ €801.47.
HHI:5001
The new price is 22.0% higher.

************************************************************

The sum of the profits, of companies 1 and 3
         before the merger, where: €35331.78

Company 1&3 with Mc = 8.82 + 1.74 * q
        Produces 45.12  with €32179.76 profit.

Company 2 with Mc = 4.0 + 2.0 * q
        Produces 44.73  with €31631.46 profit.

Total production is 98.95 units @ €800.63.
HHI:5000
The new price is 22.0% higher.

************************************************************

The sum of the profits, of companies 2 and 3
         before the merger, where: €49765.39

Company 2&3 with Mc = 4.05 + 1.05 * q
        Produces 51.07  with €41232.02 profit.

Company 1 with Mc = 26.71 + 8.34 * q
        Produces 34.71  with €19046.77 profit.

Total production is 98.95 units @ €864.94.
HHI:5182
The new price is 32.0% higher.

************************************************************
```

If companies 1 and 2 merge, the merger is profitable. However, the constant part of $MC_{1\&2}$ is almost double that of MC_2, so we have to make sure that the merged company is actually producing only in the facilities of company 2. So we use $MC_2 = MC_{1\&2}$

In [14]:

```
reset_names()
D = (2221.08, 15.81)
C: CompanyList = [Company(4, 2, '2'),
                  Company(4.1, 2.2, '3')]
companies = set_cournot_production(D, C)
quantity = sum([comp.production for comp in companies])
price = calculate_price(quantity, D)

market_stats_dump(companies, quantity, price)
print(f"HHI:{hhi(companies)}")
```

```
Company 2 with Mc = 4.0 + 2.0 * q
        Produces 45.01  with €32036.01 profit.
```

```
Company 3 with Mc = 4.1 + 2.2 * q
        Produces 44.51  with €31320.63 profit.

Total production is 89.52 units @ €805.71.
HHI:5000
```

Company "1&2" produces in both facilities, because the profits are now lower than before.

The reason that the merger of companies 1 and 2 is profitable is the asymmetry of the marginal costs. The price increased by 22%, and the third company is almost 50% more profitable. The total production is exactly the same. All in all, the merger is probably going to be prevented by the competition committee.