Міністерство освіти та науки України
Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

Звіт до лабораторної роботи №4 на тему:
"Виробництво. Виробнича функція і вартість виробництва.
Максимізація прибутку і продукції. Мінімізація витрат"

Виконав студент групи ОМ-3
Скибицький Нікіта

Київ, 2019

Зміст

1 Теоретичні відомості

1.1 Виробнича функція

За заданою таблицею значень обсягу виробництва q_i в залежності від витрат капіталу k_i і витрат праці l_i, $i = \overline{1, n}$ знаходиться аналітичний вигляд функції $Q(K, L)$.

Можна обмежитися певною сім'єю параметризованих функцій. Часто зустрічаються наступні функції:

$$
\begin{aligned}
&F(K, L) = A \cdot K^a \cdot L^{1-a}, \quad A, a > 0, \\
&F(K, L) = a \cdot K + b \cdot L, \quad a, b > 0, \\
&F(K, L) = a \cdot K \cdot L - b \cdot K^2 - c \cdot L^2, \quad a, b, c > 0, \\
&F(K, L) = a \cdot (b \cdot K + c \cdot L)^d, \quad a, b, c, d > 0, \\
&F(K, L) = \sqrt[a]{(K^a + L^a)} = (K^a + L^a)^{1/a}, \quad a \geq -1.
\end{aligned}
$$

Таке обмеження дозволяє поставити скінченно-вимірну оптимізаційну задачу

$$
\mathcal{J}(F) = \sum_{i=1}^{n} (F(k_i, l_i) - q_i)^2 \to \min,
$$

розв'язок знаходиться за допомогою ітераційних чисельних або навіть аналітичних методів (у випадку найпростіших моделей).

Рекомендується не одразу обмежуватися лише одним класом залежностей, а спробувати усі найпоширеніші, знайти оптимальну функцію з кожного класу, обчислити для них середньоквадратичні відхилення і обирати той клас, на якому досягається мінімум середньоквадратичного відхилення.

1.2 Вартість виробництва

Окрім обсягу виробництва від витрат капіталу і робочої сили залежить також вартість виробництва. Якщо для виробничої функції відомо багато різних у тому числі нелінійних варіантів які гарно працюють у тих чи інших практичних задачах, то вартість виробництва (майже) завжди (майже) лінійно залежить від витрат x_i різного роду ресурсів, а саме

$$\text{cost} = \sum_{i=1}^{m} w_i \cdot x_i = \langle w, x \rangle,$$

де w_i — вартість одиниці i-го ресурсу, а m — кількість різних ресурсів. У нашій роботі $m = 2$ а відповідні ресурси — капітал і робоча сила, але бувають і складніші ситуації.

1.3 Оптимізаційні задачі

Можна ставити багато різноманітних оптимізаційних задач в залежності від потреб виробника. Розглянемо основні три з них:

1.3.1 Максимізація прибутку

За обмеженої $\text{cost} \leq \text{TC}$ вартості виробництва максимізувати прибуток.

Оскільки прибуток визначається як

$$\text{TR} = p \cdot Q - \text{cost} = p \cdot Q(K, L) - w_1 \cdot K - w_2 \cdot L,$$

де p — ціна одиниці продукції, то задача його максимізації це цілком класична задача обмеженої оптимізації і її можна розв'язати методом множників Лагранжа. Справді, записуємо функцію Лагранжа:

$$\mathcal{L}(K, L, \lambda) = p \cdot Q(K, L) - w_1 \cdot K - w_2 \cdot L + \lambda(\text{TC} - w_1 \cdot K - w_2 \cdot L),$$

а метод множників Лагранжа також передбачає виконання наступних (не)рівностей:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial K} &= p \cdot \frac{\partial Q(K, L)}{\partial K} - (1 + \lambda) w_1 \leq 0, \\
\frac{\partial \mathcal{L}}{\partial L} &= p \cdot \frac{\partial Q(K, L)}{\partial L} - (1 + \lambda) w_2 \leq 0, \\
\frac{\partial \mathcal{L}}{\partial \lambda} &= \text{TC} - w_1 \cdot K - w_2 \cdot L \geq 0.
\end{aligned}$$

Практика показує що (майже) завжди реальні виробничі функції мають такий вигляд, що максимальний прибуток досягається на верхній межі вартості виробництва, тоді останню нерівність можна замінити рівністю.

1.3.2 Максимізація обсягів виробництва

За обмеженої $\text{cost} \leq \text{TC}$ вартості виробництва максимізувати обсяги виробництва.

Ця задача не сильно відрізняється від минулої. Зокрема, якщо виробнича функція однорідна то розв'язки цієї задачі і попередньої однакові.

1.3.3 Мінімізація вартості виробництва

За фіксованих обсягів виробництва $Q(K, L) = q_0$ мінімізувати вартість виробництва.

Ця задача трохи відрізняється від минулих, але також може бути розв'язана методом множників Лагранжа.

2 Чисельне моделювання

Було використано мову програмування Python і модуль scipy.

2.1 Код

2.2 Аналітичні апроксимації

Частина коду яка за даними підбирає параметри для виробничої функції і визначає необхідні для подальшої оптимізації функції:

```python
#!/usr/bin/env python
import numpy as np
import pandas as pd
from scipy.optimize import curve_fit, minimize


df = pd.DataFrame({
        'capital': [2860, 2740, 2950, 2880, 2510, 2690, 2990, 2850, 3000, 3070],
        'labor': [10680, 10310, 10680, 10800, 10540, 10420, 10940, 10710, 9900, 9930],
        'production': [49920, 47750, 50550, 50570, 47820, 47900, 51900, 45970, 48030, 48100]
})

df[[0, 1]] = df[['capital', 'labor']]


class ProductionFunction(object):
        def __init__(self, df: pd.DataFrame):
                self.A, self.a = curve_fit(
                        lambda cl, A, a: A * cl['capital']**a * cl['labor']**(1 - a),
                        df[['capital','labor']], df['production']
                )[0]

        def __call__(self, cl) -> float:
                # cl[0] = cl['capital'], cl[1] = cl['labor']
                return self.A * cl[0]**self.a * cl[1]**(1 - self.a)

        def __repr__(self):
                return f'{self.A:.2f} * K^{self.a:.2f} * L^{1 - self.a:.2f}'


production_function = ProductionFunction(df)

print(production_function)

p, w_1, w_2 = 5, 2, 3


def cost(cl) -> float:
        # cl[0] = cl['capital'], cl[1] = cl['labor']
        return w_1 * cl[0] + w_2 * cl[1]
```

```python
def profit(cl) -> float:
        return p * production_function(cl) - cost(cl)


def minus_profit(cl) -> float:
        return -profit(cl)


def minus_production_function(cl) -> float:
        return -production_function(cl)
```

Було отримано наступну апроксимацію виробничої функції:

$$Q(K, L) = 7.01 \cdot K^{0.31} \cdot L^{0.69}.$$

2.2.1 Максимізація прибутку

```python
TC = 100_000

cl_star_1 = minimize(
        minus_profit,
        x0=(10_000, 10_000),
        constraints=(
                {'type': 'ineq', 'fun': lambda cl: cost(cl)},
                {'type': 'ineq', 'fun': lambda cl: TC - cost(cl)},
        )
)['x']

print(
        'limited cost, maximal profit:\n'
        f'                  cl_star   = {cl_star_1}\n'
        f'             cost(cl_star) = {cost(cl_star_1)}\n'
        f'          product(cl_star) = {production_function(cl_star_1)}\n'
        f'           profit(cl_star) = {profit(cl_star_1)}\n'
)
```

Було отримано наступний розв'язок: $K^\star \approx 15674$, $L^\star \approx 22883$. За таких витрат ресурсів обсяги виробництва $Q^\star \approx 142402$, а прибуток ≈ 612009.

2.2.2 Максимізація обсягів виробництва

```python
cl_star_2 = minimize(
        minus_production_function,
        x0=(10_000, 10_000),
        constraints=(
                {'type': 'ineq', 'fun': lambda cl: cost(cl)},
                {'type': 'ineq', 'fun': lambda cl: TC - cost(cl)},
        )
)['x']
```

```python
print(
        'limited cost, maximal production:\n'
        f'                cl_star   = {cl_star_2}\n'
        f'           cost(cl_star) = {cost(cl_star_2)}\n'
        f'        product(cl_star) = {production_function(cl_star_2)}\n'
        f'         profit(cl_star) = {profit(cl_star_2)}\n'
)
```

Як вже зазначалося, для однорідних виробничих функцій розв'язки цієї і попередньої задач збігаються, що і було отримано у коді.

2.2.3 Мінімізація вартості виробництва

```python
q_0 = 55_000

cl_star_3 = minimize(
        cost,
        x0=(10_000, 10_000),
        constraints=(
                {'type': 'eq', 'fun': lambda cl: production_function(cl) - q_0},
        )
)['x']

print(
        'fixed production, minimal cost:\n'
        f'                cl_star   = {cl_star_3}\n'
        f'           cost(cl_star) = {cost(cl_star_3)}\n'
        f'        product(cl_star) = {production_function(cl_star_3)}\n'
        f'         profit(cl_star) = {profit(cl_star_3)}\n'
)
```

Було отримано наступний розв'язок: $K^\star \approx 6054$, $L^\star \approx 8838$. За таких витрат ресурсів вартість виробництва ≈ 38623, а прибуток ≈ 236377.