Пән: Бизнесте басқару шешімдерін қабылдауды қолдау жүйелері

Басқарушылық шешімдерді оңтайландыру әдістері және Python құралдарын қолдану

Дәріс 14

Шаяхметова А.С., PhD, қауымдастырылған профессор
Жасанды интеллект және Big Data кафедрасы
2026-2027 оқу жылы

Дәрістің мақсаты

Басқарушылық шешімдерді оңтайландыру есебінің мәнін түсіндіру. Сызықтық, бүтін сандық, сызықтық емес және эвристикалық әдістерді қарастыру. Python кітапханаларымен (SciPy, PuLP, OR-Tools) таныстыру. Практикалық мысалдар арқылы оңтайлы шешімді табуды көрсету.

1. **Оңтайландыру есебі**
2. **Әдістер**
3. **Python**

Дәріс жоспары

1. Оңтайландыру есептері мен әдістері
2. Сызықтық бағдарламалау
3. Бүтін сандық, сызықтық емес және эвристикалық әдістер
4. Python құралдары және практикалық мысалдар

Оңтайландыру есебінің құрылымы

$$\max \text{ (немесе } \min) \ f(x)$$
$$g_i(x) \le b_i, \quad i = 1, \ldots, m$$
$$x \ge 0$$

x	шешім айнымалылары (өндіріс көлемі, бюджет бөлінісі)
f(x)	мақсат функциясы (пайда, шығын, уақыт)
$g_i(x) \le b_i$	шектеулер (ресурс, бюджет, сұраныс)
Рұқсат аймақ	барлық шектеуді қанағаттандыратын шешімдер жиыны
Оңтайлы	мақсат функциясына ең жақсы мән беретін нүкте

Оңтайландыру үдерісі

1. **Мәселені қою**
2. **Математикалық модель құру**
3. **Шешу (солвер)**
4. **Талдау және енгізу**

Мысалдар:

- Өндіріс жоспары: ресурс шектеуінде пайданы арттыру
- Жеткізу маршруттары: көлік шығынын азайту
- Бюджетті жобалар арасында бөлу
- Қызметкерлердің жұмыс кестесін жасау

Оңтайландыру әдістерінің жіктелуі

Әдіс	Ерекшелігі	Бизнес мысалы
Сызықтық (LP)	Мақсат пен шектеулер сызықтық	Өндіріс жоспары
Бүтін сандық (MILP)	Айнымалылар бүтін немесе 0/1	Жоба таңдау, кесте
Сызықтық емес (NLP)	Мақсат немесе шектеу сызықтық емес	Баға оңтайландыру
Эвристикалық	Жуық шешім, күрделі есептер	Маршруттау, орналастыру

Сызықтық бағдарламалау моделі

$$\max Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$
$$a_{11}x_1 + \ldots + a_{1n}x_n \le b_1$$
$$\ldots$$
$$a_{m1}x_1 + \ldots + a_{mn}x_n \le b_m , \quad x_j \ge 0$$

c_j	бір өнімнің пайдасы (мақсат функциясының коэффициенті)
a_{ij} , b_i	ресурс шығыны және ресурс көлемі
Оңтайлы	рұқсат етілген аймақтың төбелерінің бірінде жатады
Әдістер	симплекс-әдіс, ішкі нүктелер әдісі

Мысал: өндіріс жоспарын оңтайландыру

Көрсеткіш	А өнімі	В өнімі	Ресурс
Пайда, мың тг	40	30	—
Станок сағаты	2	1	100
Еңбек сағаты	1	1	80

Модель: $\max Z = 40x_1 + 30x_2$

$2x_1 + x_2 \le 100, \quad x_1 + x_2 \le 80, \quad 0 \le x_1 \le 40, \quad x_2 \ge 0$

x_1, x_2 — А және В өнімдерінің көлемі; $x_1 \le 40$ — А өніміне сұраныс шегі

Мысал бойынша шешім

Төбе (x_1; x_2)	$Z = 40x_1 + 30x_2$
(0; 0)	0
(40; 0)	1600
(40; 20)	2200
(20; 60)	**2600 ← оңтайлы**
(0; 80)	2400

Оңтайлы жоспар: 20 бірлік А және 60 бірлік В, пайда 2600 мың тг
Екі ресурс та толық пайдаланылады. Көлеңкелі баға: +1 станок сағаты → +10, +1 еңбек сағаты → +20 мың тг

Бүтін сандық бағдарламалау (MILP)

1. Айнымалылар бүтін ($x_i \in \mathbb{Z}$) немесе бинарлық ($x_i \in \{0; 1\}$) болады
2. Бинарлық айнымалы: жобаны таңдау (1) немесе таңдамау (0) шешімі
3. Мысалдар: жобалар портфелі, қызметкерлер кестесі, қойма орналастыру
4. Шешу әдісі: тармақтар мен шектер әдісі (branch and bound)
5. Бүтін шешім сызықтық шешімді дөңгелектеуден әрдайым жақсы емес

Сызықтық емес оңтайландыру: баға таңдау

Сұраныс: $q = 1000 - 20p$, шығын: $c = 10$
Пайда: $\pi(p) = (p - 10)\cdot(1000 - 20p)$
$d\pi/dp = 0 \rightarrow p^* = (50 + 10) / 2 = 30$

Оңтайлы баға	Сатылым	Пайда
$p^* = 30$	$q = 400$	$\pi = 8000$

Күрделі есептерде градиенттік әдістер (SLSQP, BFGS) қолданылады.
Жергілікті және ғаламдық оңтайлы шешім ерекшеленеді.

Эвристикалық және метаэвристикалық әдістер

Мақсаты	Дәл шешуге уақыт жетпейтін күрделі есептерде жуық шешім табу
Генетикалық	Шешімдер «популяциясы» іріктеу мен мутация арқылы жақсарады
Имитациялық күйдіру	Нашар шешімді уақытша қабылдап, жергілікті минимумнан шығады
PSO, құмырсқа	Бөлшектер тобы әдісі, құмырсқа алгоритмі – маршруттау есептері
Шектеуі	Оңтайлылыққа кепілдік жоқ, бірақ іс жүзінде жақсы шешім береді

Python кітапханалары

Кітапхана	Мақсаты
NumPy, pandas	Деректерді өңдеу, матрицалық есептеу
SciPy (optimize)	linprog, milp, minimize – LP, MILP, NLP
PuLP, Pyomo	Модельді алгебралық түрде жазу, солверлерге беру
OR-Tools	Маршруттау, кесте, тағайындау есептері
statsmodels, scikit-learn	Болжам моделі: оңтайландыруға кіріс деректер
matplotlib	Нәтижелерді визуализациялау

Python: сызықтық бағдарламалау (linprog)

```
from scipy.optimize import linprog

c = [-40, -30]          # max Z → min (-Z)
A = [[2, 1], [1, 1]]
b = [100, 80]
res = linprog(c, A_ub=A, b_ub=b,
              bounds=[(0, 40), (0, None)])
print(res.x, -res.fun)  # [20. 60.]  2600.0
```

Нәтиже графикалық шешіммен сәйкес: $x_1 = 20$, $x_2 = 60$, $Z = 2600$.

Python: бағаны оңтайландыру (minimize)

```
from scipy.optimize import minimize

# пайда = (p - 10) * (1000 - 20p);  max → min
profit = lambda p: -(p[0] - 10) * (1000 - 20 * p[0])
res = minimize(profit, x0=[20])
print(res.x, -res.fun)   # [30.]  8000.0
```

Сандық нәтиже аналитикалық шешімге
(p* = 30, π = 8000) сәйкес.

Python: жобаларды таңдау (milp)

```
import numpy as np
from scipy.optimize import milp, LinearConstraint, Bounds

profit = np.array([50, 35, 60, 40])   # пайда
cost   = np.array([30, 20, 40, 25])   # шығын
res = milp(-profit, integrality=np.ones(4),
           bounds=Bounds(0, 1),
           constraints=LinearConstraint(cost, ub=80))
print(res.x, -res.fun)   # [1. 1. 0. 1.]  125.0
```

Бюджет 80 болғанда 1, 2 және 4-жобалар таңдалады, пайда 125.

Қорытынды

1. Оңтайландыру есебі мақсат функциясымен, айнымалылармен және шектеулермен сипатталады
2. Әдіс есептің түріне қарай таңдалады: LP, MILP, NLP, эвристика
3. Графикалық және симплекс әдісі сызықтық есептің оңтайлы шешімін табады
4. Python (SciPy, PuLP, OR-Tools) модельді тез құрып, шешуге мүмкіндік береді
5. Нәтижені көлеңкелі баға және сезімталдық талдауымен бағалау керек

Әдебиеттер

1. Трофимова Л.А., Трофимов В.В. Методы принятия управленческих решений. — М.: Юрайт, 2024.

2. Бусов В.И., Лябах Н.Н., Саткалиева Т.С., Таспенова Г.А. Теория и практика принятия управленческих решений. — М.: Юрайт, 2024.

3. Liu S., Zaraté P. et al. (eds.). Decision Support Systems XIII. — Cham: Springer, 2023.

4. James G., Witten D., Hastie T., Tibshirani R., Taylor J. An Introduction to Statistical Learning: with Applications in Python. — Springer, 2023.

Назарларыңызға рахмет!

Басқарушылық шешімдерді оңтайландыру
әдістері және Python құралдарын қолдану

Дәріс 14