Optimisation Methods - Linear Programming

Mateusz Pełechaty

1 April 2024

1 Exercise (Sysło, Deo, Kowalik 1993)

One of the tests for precision and resistance of LP algorithms is the following problem:

$$\min \mathbf{c}^T\mathbf{x}$$

$$A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geqslant \mathbf{0}$$

$$A_{i,j} = \frac{1}{i+j-1}, \quad i,j = 1,\ldots,n$$

$$c_i = b_i = \sum_{j=1}^{n} \frac{1}{i+j-1}, \quad i = 1,\ldots,n$$

It's solution should be $\mathbf{x} = \mathbf{1}$. Problem is *ill-conditioned* due to A being Hilbert's Matrix.

1. Describe problem in `GNU MathProg` and solve it in `glpsol` ☑
2. Use `glpsol` to find n for which there is 2-digit precision. ☑
3. Print relative error $||x - \bar{x}||_2/||x||_2$ for every n. $\bar{x}$ is computed x ☑
4. Generalize the solution by separating model and data. Maximally parametrize model description ☑
5. * Solve it with all `clp`, `HiGHS`, `GLPK` ☒

1.1 Model

Parameters

- n — size of Matrix A
- $A_{i,j}$, c_i, b_i as described above

Decision Variables

- x — vector of size n

Constraints

- $Ax = b$ — $\forall i \in [n] \quad b_i = \sum_{j\in[n]} A_{i,j} \cdot x_j$
- $x_i \geqslant 0$

Objective Function

- $c^T \cdot x$ — $\sum_{i\in[n]} c_i \cdot x_i$

1.2 Sources

Solution in `GNU MathProg` can be found in *src/ex1/model.mod* and *src/ex1/data.dat* .

1.3 Precision for given n

Following table presents relative error of the problem solving depending on n As we can see, at $n = 8$, solution loses it's precision.

n	error	solution
2	1.05e-15	(1.00, 1.00)
3	3.67e-15	(1.00, 1.00, 1.00)
4	3.27e-13	(1.00, 1.00, 1.00, 1.00)
5	3.35e-12	(1.00, 1.00, 1.00, 1.00, 1.00)
6	6.83e-11	(1.00, 1.00, 1.00, 1.00, 1.00, 1.00)
7	1.67e-8	(1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)
8	0.51	(1.00, 0.99, 1.05, 0.74, 1.70, 0, 1.72, 0.80)
9	0.68	(1.00, 1.01, 0.93, 1.38, 0.00, 2.40, 0.00, 1.30, 0.99)
10	0.99	(1.00, 0.98, 1.18, 0.21, 2.56, 0.00, 0.00, 2.93, 0.00, 1.15)

2 Exercise (Camper rental)

Certain company rents campers in several cities and in two standards: `BASIC` and `VIP`. In this situation, company has surplus of campers in some places and shortage in other places. It can be seen here.

	Shortage		Surplus	
cities	`BASIC`	`VIP`	`BASIC`	`VIP`
Warszawa	—	4	14	—
Gdańsk	20	—	—	2
Szczecin	—	—	12	4
Wrocław	8	—	—	10
Kraków	—	8	10	—
Berlin	16	4	—	—
Rostok	2	—	—	4
Lipsk	3	—	—	10
Praga	—	4	10	—
Brno	9	—	—	2
Bratysława	4	—	—	8
Koszyce	4	—	—	4
Budapeszt	8	—	—	4
Total	74	20	46	48

Table 1: Shortage and Surplus of campers in certain places

Plan out camper relocation with following constraints:

- Cost of `BASIC` camper relocation is proportional to distance
- Cost of `VIP` is 15% higher than `BASIC`
- `BASIC` can be replaced by `VIP`

Do the following

1. Describe problem in `GNU MathProg` and solve it in `glpsol` ☑

2. Generalize the solution by separating model and data. Maximally parametrize model description ☑

3. Is explicitly writing in model *decision variables are integers* necessary? ☑
   No. There is no reason to. The only possible way to floating point variables to show up in solution, would be if from one city we would send to many different cities at the same time, which could be the case if cost of driving to these cities was the same. Although I could probably make some proof that greedy method of choosing is the best method, so we need to fill shortage in full. As shortage is integer, we do not create any floats. I checked it by looking through $s_{i,j,k}$.

2.1 Model

Parameters

- C — Cities that company is located at.
- $T = \{\mathcal{B}, \mathcal{V}\}$ — types of campers. $\mathcal{B}$ stands for BASIC and $\mathcal{V}$ stands for VIP
- $\delta_{i,j} \quad i \in C, j \in C$ — Distance between i and j.
- $u_{i,j} \quad i \in C, t \in T$ — Surplus of campers in city i and type t.
- $d_{i,j} \quad i \in C, t \in T$ — Shortage of campers in city i and type t.
- p — Cost of transport per km for BASIC.

Decision Variables

- $s_{t,j,k} \quad t \in T, j \in C, k \in C$ — How many campers of type t should we send from city j to city k.

Constraints

- Any city c_1 , cannot send more than amount of campers of type t on surplus.

$$\sum_{c_2 \in C} s_{t,c_1,c_2} \leqslant u_{c_1,c_2}$$

- Shortage of VIP campers in every city c_1 must be satisfied

$$\sum_{c_2 \in C} s_{\mathcal{V},c_2,c_1} \geqslant d_{c_1}$$

- Shortage of BASIC campers in every city c_1 must be satisfied. Also, vips from the same city can be BASIC. They do it by sending itself to the same city they are in.

$$\sum_{c_2 \in C, t \in T} s_{t,c_2,c_1} \geqslant \sum_{t \in T} d_{t,c_1}$$

Objective Function

- Minimize cost of sending campers to different cities

$$\text{Minimize} \quad p \cdot \sum_{c_1 \in C, c_2 \in C} \delta_{c_1,c_2} \cdot (s_{\mathcal{B},c_1,c_2} + 1.15 s_{\mathcal{V},c_1,c_2})$$

2.2 Sources

Solution in `GNU MathProg` can be found in *src/ex2/model.mod* and *src/ex2/data.dat* .

2.3 Solution

Lowest possible cost of sending campers to different cities according to *glpsol* is 20595.8

3 Exercise (Factory)

Factory produces 4 type of products. P_1, P_2, P_3, P_4. To create them, it uses resources R_1, R_2, R_3. During creation of P_1, and P_2, some of R_i turns into $W_{i,j}$ - Waste from creation of P_j out of resource R_i. $W_{i,j}$ can be used in creation of P_3 and P_4, however if it won't be used, then factory will have to pay for utilization. Factory wants to know how to maximize profit.

Resource	Min	Max	Cost of 1kg
	(kg)	(kg)	($)
R_1	2000	6000	2.1
R_2	3000	5000	1.6
R_3	4000	7000	1.0

Table 2: Minimum and maximum values of resources to buy

Product	Value
P_1	3$/kg
P_2	2.5$/kg
P_3	0.6$/kg
P_4	0.5$/kg

Table 3: Value of products

Waste	Cost
$W_{1,1}$	0.1
$W_{1,2}$	0.2
$W_{2,1}$	0.2
$W_{2,2}$	0.2
$W_{3,1}$	0.4
$W_{3,2}$	0.5

Table 4: Disposal Costs

Product	Input	Variable Constraints	Output
P_1	r_1 kg R_1, r_2 kg R_2, r_3 kg R_3	$r_1 \geqslant 0.2 \cdot r$, $r_2 \geqslant 0.4 \cdot r$, $r_3 \leqslant 0.1 \cdot r$ $r = r_1 + r_2 + r_3$	$0.1 \cdot r_1$ kg W11, $0.2 \cdot r_2$ kg W21, $0.4 \cdot r_3$ kg W31 Other P1
P_2	r_1 kg R_1, r_2 kg R_2, r_3 kg R_3	$r_1 \geqslant 0.1 \cdot r$, $r_3 \leqslant 0.3 \cdot r$ $r = r_1 + r_2 + r_3$	$0.2 \cdot r_1$ kg W12, $0.2 \cdot r_2$ kg W22, $0.5 \cdot r_3$ kg W32, Other P2
P_3	$w_{1,1}$ kg $W_{1,1}$ $w_{2,1}$ kg $W_{2,1}$ $w_{3,1}$ kg $W_{3,1}$ r_1 kg R_1	$r_1 = 0.3 \cdot r$ $r = w_{1,1} + w_{2,1} + w_{3,1} + r_1$	r kg P3
P_4	$w_{1,2}$ kg $W_{1,2}$ $w_{2,2}$ kg $W_{2,2}$ $w_{3,2}$ kg $W_{3,2}$ r_2 kg R_2	$r_2 = 0.3 \cdot r$ $r = w_{1,2} + w_{2,2} + w_{3,3} + r_2$	r kg P4

Table 5: Possible Production Lines

Do the following

1. Describe problem in `GNU MathProg` and solve it in `glpsol` ☑
2. How many resources R_1, R_2, R_3 should factory buy? ☑
3. How many resources should be used to what product? ☑
4. How many waste should be utilized and how many should go towards production of C and D? ☑

3.1 Model

Parameters

- $\mathcal{R} = \{1, 2, 3\}$— Indexes of resources
- $\mathcal{P} = \{1, 2, 3, 4\}$ — Indexes of products, although $\mathcal{P}^1$ and $\mathcal{P}^2$ will mean first and second class products
- $L_i \quad i \in \mathcal{R}$ — lower bound on buying resource i.
- $U_i \quad i \in \mathcal{R}$ — upper bound on buying resource i.
- $C_i \quad i \in \mathcal{R}$ — cost of buying resource i.
- $P_i \quad i \in \mathcal{P}$ — price of product i
- $Wc_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}^1$ — Cost of utilizing wastes from resource i created in production of j
- $W_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}^1$ — Factor of which wastes from resource i will be created in production line j
- $M^{\geqslant}_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}$ — Constraint: *Amount of resource i should be not less than $M_{i,j}$ during production of j.*
- $M^{\leqslant}_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}$ — Constraint: *Amount of resource i should be not more than $M_{i,j}$ during production of j.*

Decision Variables

- $b_i \quad i \in \mathcal{R}$ — How many resources of type i should company buy.
- $r_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}$ — How many of resource i went into production line j.
- $m_i \quad i \in \mathcal{P}$ — Ingredient mass given on input of production line i.
- $w_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}^1$ — wastes created from resource i on production line j
- $\hat{w}_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}^1$ — wastes destinated to utilization.
- $\overline{w}_{i,j} \quad i \in \mathcal{R}, j \in \mathcal{P}^1$ — wastes destinated to use in creation of second class products
- $p_i \quad i \in \mathcal{P}$ — weight of product i

Constraints

- Limit on minimum and maximum of each resource i to buy.
$$U_i \geqslant b_i \geqslant L_i$$
- Division of resource i into his production lines destinations.
$$b_i = \sum_{j \in \mathcal{P}} r_{i,j}$$
- Mixing of ingredients $i \in \mathcal{P}^1$ in first class production lines
$$m_i = \sum_{j \in \mathcal{R}} r_{j,i}$$
- Mixing of ingredients $i \in \mathcal{P}^2$ in second class production lines
$$m_i = r_{i-2,i} + \sum_{j \in \mathcal{R}} \overline{w}_{j,i-2}$$
- Division of wastes from resource i during production j to utilization and later usage.
$$w_{i,j} = \hat{w}_{i,j} + \overline{w}_{i,j}$$
- Usage of resource i in production line j is not smaller in mix than certain value from task.
$$r_{i,j} \geqslant M^{\geqslant}_{i,j} \cdot m_j$$
- Usage of resource i in production line j is not bigger in mix than certain value from task.
$$r_{i,j} \leqslant M^{\leqslant}_{i,j} \cdot m_j$$
- Production of first class products $i \in \mathcal{P}^1$ produces product and waste
$$m_i = p_i + \sum_{j \in \mathcal{R}} w_{j,i}$$
- Factor of waste produced by resource i during production of first class products $j \in \mathcal{P}^1$ is given by $W_{i,j}$.
$$w_{i,j} = r_{i,j} \cdot W_{i,j}$$
- During creation second class products $i \in \mathcal{P}^2$, mix is equivalent to product.
$$p_i = m_i$$

Objective Function

- sell := $\sum_{i \in \mathcal{P}} p_i \cdot P_i$
- product cost := $\sum_{i \in \mathcal{R}} b_i \cdot C_i$
- utilization := $\sum_{i \in \mathcal{R}, j \in \mathcal{P}^1} \hat{w} \cdot Wc_{i,j}$

$$\text{Maximize profit:} \quad \text{sell} - \text{product cost} - \text{utilization}$$

3.2 Sources

Solution in `GNU MathProg` can be found in *src/ex3/model.mod* and *src/ex3/data.dat* .

3.3 Answers to questions

How many resources R_1, R_2, R_3 should factory buy?

According to *glpsol*, factory should buy

Resource 1	6000kg
Resource 2	5000kg
Resource 3	4000kg

How many resources should be used to what product in kilograms?

	Product 1	Product 2	Product 3	Product 4
Resource 1	1175	4725	99	0
Resource 2	940	4060	0	0
Resource 3	235	3765	0	0

How many waste should be utilized and how many should go towards production of C and D?

	Product 1	Product 2
Resource 1	117	0
Resource 2	188	0
Resource 3	94	0

Table 6: Wastes that were used in creation of products 3 and 4

	Product 1	Product 2
Resource 1	0	945
Resource 2	0	812
Resource 3	0	1882

Table 7: Wastes that were utilized

4 Exercise (Student Schedule)

Student needs to choose his schedule. He needs to pick 5 subjects (A, B, C, D, E). For every subject there are 4 groups and he needs to pick one of them. Groups are following:

	A	B	C	D	E
I	Pn. 13-15	Pn. 13-15	Wt. 8-11	Pn. 8-10	Pn. 9-10:30
II	Wt. 10-12	Wt. 10-12	Wt. 10-13	Pn. 8-10	Pn. 10:30-12
III	Śr. 10-12	Śr. 11-13	Cz. 15-18	Cz. 13-15	Pt. 11-12:30
IV	Śr. 11-13	Cz. 8-10	Cz. 17-20	Pt. 13-15	Pt. 13-14:30

He also has preferences and they are following:

	A	B	C	D	E
I	5	4	3	10	0
II	4	4	5	10	5
III	10	5	7	7	3
IV	5	6	8	5	4

He wants to sign up for classes and choose schedule that maximizes sum of preference points. There are also 3 constraints that he wants to meet:

- Do not sign up for more than 4 hours a day
- Everyday, have 1 hour lunch break between 12 and 14.
- Practice sport at least once a week. Trainings are Mon. 13-15, Wed. 11-13, Wed. 13-15

Do the following

1. Describe problem in `GNU MathProg` and solve it in `glpsol` □
2. Is there a schedule where all groups are only in Mon, Tue, Thu, and all preference points are not less than 5? □

4.1 Model

Names

- S — Subjects
- GSI — Group indexes for subjects
- GAI — Group indexes for activities
- A — Activities such as Lunches at every day and sport.
- SH — Schedule Hours (0..48)
- D — Days

Parameters

- $\mathrm{GSH}_{g\in\mathrm{GSI},s\in\mathrm{S}}$ - Group Start Hours for given group index g and subject s
- $\mathrm{GSD}_{g\in\mathrm{GSI},s\in\mathrm{S}}$ - Group Start Days for given group index g and subject s
- $\mathrm{ASH}_{g\in\mathrm{GAI},a\in\mathrm{A}}$ - Activity Start Hours for given group index g and activity a
- $\mathrm{ASD}_{g\in\mathrm{GAI},a\in\mathrm{A}}$ - Activity Start Days for given group index g and activity a
- $\mathrm{P}_{g\in\mathrm{GSI},s\in\mathrm{S}}$ - Preference of picking group index g and subject s
- $\mathrm{TA}_{a\in\mathrm{A}}$ — Time needed for activity a
- $\mathrm{TS}_{s\in\mathrm{S}}$ — Time needed for subject s

Decision Variables

- $\mathrm{GP}_{g \in \mathrm{GSI}, s \in \mathrm{S}}$ — Group Subjects picked by student
- $\mathrm{AP}_{g \in \mathrm{GAI}, a \in \mathrm{A}}$ — Group Activities picked by student

Constraints

- Student have to take one group from every activity i

$$\sum_{j \in \mathrm{GAI}} \mathrm{AP}_{j,i} = 1$$

- Student have to take one group from every subject i

$$\sum_{j \in \mathrm{GSI}} \mathrm{GP}_{j,i} = 1$$

- Do not allow multitasking by calculating for every day d and hour$h$ and check if overlaps happen

$$\mathrm{activity}_{d,h} := \sum_{g \in \mathrm{GAI}, a \in \mathrm{A}} [\mathrm{ASD}_{g,a} = d] \cdot [\mathrm{ASH}_{g,a} \leqslant h] \cdot [h \leq \mathrm{ASH}_{g,a} + \mathrm{AT}_a] \cdot [\mathrm{AP} = 1]$$

$$\mathrm{subject}_{d,h} := \sum_{g \in \mathrm{GSI}, a \in \mathrm{S}} [\mathrm{GSD}_{g,a} = d] \cdot [\mathrm{GSH}_{g,a} \leqslant h] \cdot [h \leq \mathrm{GSH}_{g,a} + \mathrm{GT}_a] \cdot [\mathrm{GP} = 1]$$

$$\mathrm{activity}_{d,h} + \mathrm{subject}_{d,h} \leqslant 1$$

- Do not overload on certain day d

$$\sum_{g \in \mathrm{GSI}, s \in \mathrm{S}} [\mathrm{GST}_{g,s} = d] \cdot [\mathrm{GP}_[g, s] = 1] \cdot \mathrm{GT}_s \leqslant 8$$

Objective Function

Sum of preferences of picked groups

$$\sum_{g \in \mathrm{GSI}, s \in \mathrm{S}} [\mathrm{GP}_{g,s} \cdot P_{g,s}]$$

4.2 Sources

Solution in `GNU MathProg` can be found in *src/ex4/model.mod* and *src/ex4/data.dat* .

4.3 Solution Basic

Student can acquire 37 preference points at most and should sign up for following subjects:

	A	B	C	D	E
I	0	0	0	1	0
II	0	1	0	0	1
III	1	0	0	0	0
IV	0	0	1	0	0

4.4 Solution Strict

Student can acquire 28 preference points at most and should sign up for following subjects:

	A	B	C	D	E
I	1	0	0	0	0
II	0	0	1	0	1
III	0	0	0	1	0
IV	0	1	0	0	0