Konspekt

Piotr Chołda

14 listopada 2017

1 Programowanie liniowe — podstawy teoretyczne

1.1 Optymalizacja z użyciem programowania liniowego LP (*linear programming*)

1. Programowanie liniowe: postać ogólna zadania.
2. Programowanie liniowe: postać standardowa zadania (*standard form*):
   - MIN:
     - ⋆ $z = \sum_{j=1,2,\ldots,n} c_j x_j$
   - S.t.:
     - ⋆ $\sum_{j=1,2,\ldots,n} a_{ij} x_j = b_i \quad i = 1, 2, \ldots, m;$
     - ⋆ $x_j \geq 0 \quad j = 1, 2, \ldots, n;$
     - ⋆ $x_j \in \mathbb{R} \quad j = 1, 2, \ldots, n.$

   i jej sformułowanie macierzowe[1]:

   - MIN:
     - ⋆ $z = \mathbf{c}^T\mathbf{x}$ (często pomija się oznaczanie transpozycji, ponieważ zazwyczaj wiadomo, o co chodzi; dlatego pisze się też: $z = \mathbf{cx}$).
   - S.t.:
     - ⋆ $\mathbf{Ax} = \mathbf{b}$;
     - ⋆ $\mathbf{x} \geq \mathbf{0}$.

   można to zapisać kompaktowo: $z = \max\{\mathbf{cx} : \mathbf{Ax} = \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}\}$.
3. Sprowadzanie dowolnego zadania programowania liniowego do postaci standardowej.
4. Programowanie liniowe: postać kanoniczna/normalna zadania (*canonical form*).
5. Zależność między funkcją celu i ograniczeniami dla programowania liniowego — różne możliwości nt. istnienia rozwiązań:

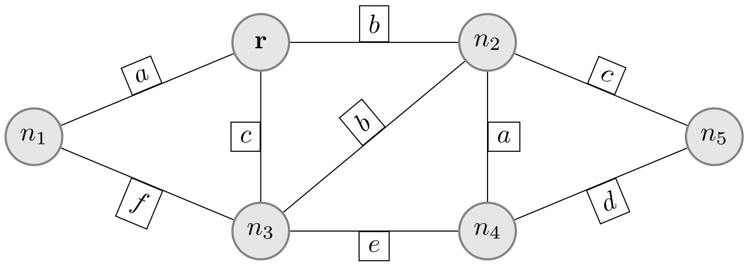
[1] W tym przypadku wektor oznacza dla nas „wektor kolumnowy", np. $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$.

- istnieje **jedno** rozwiązanie dopuszczalne, które oczywiście jest optymalne;
- istnieje **nieskończenie wiele** rozwiązań dopuszczalnych, w tym **jedno** jest optymalne;
- istnieje **nieskończenie wiele** rozwiązań dopuszczalnych, w tym **nieskończenie wiele** optymalnych;
- problem jest nierozwiązywalny (*infeasible*), tj. sprzeczny (założenia/ograniczenia są sprzeczne i zbiór rozwiązań dopuszczalnych jest pusty): **zero** rozwiązań dopuszczalnych;
- problem jest nieograniczony (*unbounded*): dla każdego rozwiązania dopuszczalnego można znaleźć inne rozwiązanie dopuszczalne o jeszcze lepszej wartości funkcji celu — **nieskończenie wiele** rozwiązań dopuszczalnych, ale **zero** rozwiązań optymalnych.

6. Prosta (*line*) i hiperpłaszczyzna (*hyperplane*). Półpłaszczyzna (*half-plane*) i półprzestrzeń (*half-space*).
7. Przestrzeń rozwiązań dopuszczalnych / zbiór dopuszczalny (*solution space*) zadania programowania liniowego. Wielokąt wypukły (*convex polyhedron*), powłoka/obwiednia/otoczka wypukła (*convex hull*), wielokomórka (*polytope*).
8. Wierzchołek wielokąta — punkt ekstremalny (*extreme point*).
9. Kombinacja wypukła (*convex combination*). Twierdzenie Caratheodory'ego o powłokach wypukłych podzbiorów przestrzeni euklidesowych.
10. Graficzny sposób rozwiązywania zadań programowania liniowego dla prostych problemów.
11. Podstawowy sposób rozwiązywania zadań programowania liniowego (George Dantzig 1947 — metoda sympleks lub sympleksów albo algorytm sympleksowy, *simplex algorithm*):
    - sympleks (*simplex*);
    - macierz bazowa (*basis matrix*), rozwiązanie bazowe (*basis solution*), dopuszczalne rozwiązanie bazowe (*feasible basis solution*);
    - zrewidowany algorytm sympleksowy (*revised simplex algorithm*);
    - właściwości rozwiązania optymalnego — liczba niezerowych zmiennych w rozwiązaniu optymalnym;
    - potencjalne trudności związane z działaniem algorytmu.
12. Inne metody rozwiązywania problemów programowania liniowego (metoda elipsoidalna Chaczjana 1979, algorytm projekcyjny Karmarkara 1984).

1.2 Zadania

- Pewien problem optymalizacyjny zadano w następujący sposób:
    - ⋆ funkcja celu: $\min z = 2x_1 + 2x_2$



Rysunek 1: Przykładowy graf ważony

⋆ ograniczenia:

  * $x_1 \le 5$
  * $x_2 \le 5$
  * $Ax_1 + Bx_2 \le C$

Proszę podać takie przykładowe wartości A, B i C ($A, B, C \in \mathbb{R} \setminus \{0\}$), że ten problem nie będzie miał rozwiązania.

- Na rys. 1 dany jest graf prosty ważony (wartości wag w prostokącikach należy zastąpić wartościami reprezentowanymi przez cyfry numeru własnego indeksu, gdzie numer indekstu to $fedcba$). Proszę znaleźć drzewo najkrótszych ścieżek z korzenia **r**, a następnie sformułować w postaci zadania programowania liniowego LP problem poszukiwania tego drzewa (indeksy, stałe, zmienne, funkcję celu i ograniczenia proszę dobrać dla tego konkretnego grafu). Jeśli to potrzebne, każdą krawędź grafu prostego można zastąpić parą łuków przeciwnie skierowanych o tej samej wadze.

- Na rys. 1 dany jest graf prosty ważony (wartości wag w prostokącikach należy zastąpić wartościami reprezentowanymi przez cyfry numeru własnego indeksu, gdzie numer indekstu to $fedcba$). Proszę znaleźć minimalne rozcięcie między n_1 i n_5, a następnie sformułować w postaci zadania programowania liniowego LP problem poszukiwania tego minimalnego rozcięcia (indeksy, stałe, zmienne, funkcję celu i ograniczenia proszę dobrać dla tego konkretnego grafu). Jeśli to potrzebne, każdą krawędź grafu prostego można zastąpić parą łuków przeciwnie skierowanych o tej samej wadze.

- Niech x_{ij} przyjmuje wartość 1, jeśli ruter i jest fizyczne połączony z ruterem j (w przeciwnym przypadku x_{ij} przyjmuje wartość 0). Niech w naszej sieci będzie N ruterów. Jak można zinterpretować poniższe ograniczenia, tłumacząc je inżynierowi zajmującemu się konfiguracją ruterów?

$$\begin{array}{ll} x_{ii} = 0 & i = 1, 2, \ldots, N \\ \sum_{j=1}^{N} x_{ij} \le 5 & i = 1, 2, \ldots, N \\ \sum_{j=1}^{N} x_{ij} \ge 1 & i = 1, 2, \ldots, N \end{array}$$

- W każdym z N interesujących nasze przedsiębiorstwo miast instalujemy rutery. Niech x_{ij} przyjmuje wartość 1, jeśli nasze przedsiębiorstwo instaluje rutery produkowane przez firmę oznaczoną indeksem i w mieście indeksowanym za pomocą j (w przeciwnym przypadku x_{ij} przyjmuje wartość 0). Na rynku mamy P firm produkujących rutery. Jak można zinterpretować poniższe ograniczenia, tłumacząc je inżynierowi zajmującem się instalacją ruterów?

$$\sum_{i=1}^{P} x_{ij} \geq 2 \qquad j = 1, 2, \ldots, N$$

- W każdym z P interesujących nas miast nasze przedsiębiorstwo instaluje koncentratory. Niech x_{ij} oznacza liczbę klientów przyłączonych do koncentratora w mieście indeksowanym za pomocą j, ale tylko takich klientów, którzy wykupili klasę obsługi QoS oznaczaną indeksem i. Nasze przedsiębiorstwo obsługuje N typów klas obsługi. Jak można zinterpretować poniższe ograniczenia, tłumacząc je inżynierowi instalującemu koncentratory?

$$\sum_{i=2}^{N} x_{ij} = x_{1j} \qquad j = 1, 2, \ldots, P$$

1.3 Lektury

1.3.1 Materiał wykładu

Zagadnienia omówione w ramach tego wykładu są w dużym stopniu opisane w następującej książce:

- Michał Pióro and Deepankar Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers—Elsevier, San Francisco, CA, 2004: appendix C.3.