

Úvod do matematickej logiky

LP, ILP, B&B

Robert Lukotka

`lukotka@dcs.fmph.uniba.sk`

`www.dcs.fmph.uniba.sk/~lukotka`

M-255

Lineárne programovanie

- Množina premenných: $x_1, \dots, x_n$.
- Cieľová funkcia: $c_1x_1 + c_2x_2 + \dots + c_nx_n$
- Obmedzenia, nerovnosti: $a_{1i}x_1 + a_{2i}x_2 + \dots + a_{ni}x_n \leq b_i$, pre $i \in \{1, \dots, j\}$.
- Obmedzenia, rovnosti: $c_{1i}x_1 + c_{2i}x_2 + \dots + c_{ni}x_n = d_i$, pre $i \in \{1, \dots, k\}$.

V maticovom zápise

$$\max\{c^T x \mid Ax \leq b \wedge Cx = d\},$$

kde A a C sú matice, b , d , x sú vektory reálnych čísel.

Lineárne programovanie

Ak chceme vyriešiť problém LP, nemusíme pracovať s oboma typmi obmedzení:

- Na obmedzenia v tvare rovnosti možno použiť substitúciu a zbaviť sa rovníc.
 - Nahradenie rovnosti dvoma nerovnosťami je zlý nápad.
- Obmedzenia v tvare nerovnosti možno nahradiť obmedzeniami v tvare $x_i \geq 0$.

Lineárne programovanie - zložitosť

- Existuje slabo polynomiálny algoritmus.
- Pre mnohé triedy LP existuje silno polynomiálny algoritmus.
Napríklad
 - LP pre maximálny tok.
- Ako mnohé iné výpočty spoliehajúce sa na prácu s necelými číslami, numerická stabilita môže byť problémom.

Lineárne programovanie, algoritmy

- Simplexová metóda:
 - Hľadá optimálne riešenie prechádzaním po vrcholoch a hranách polytopu.
 - V najhoršom prípade exponenciálna zložitosť, v praxi však funguje dobre (na náhodných inštanciách kubický počet aritmetických operácií).
- Metódy vnútorných bodov:
 - Prechádzajú vnútom polytopu.
 - Zložitosť - rôzna, $O(n^3)$ aritmetických operácií, v niektorých prípadoch aj lepšia (často závisí od algoritmu na násobenie matíc).

Lineárne programovanie, príklady

Maximálny tok v sieti (pre neorientovaný graf).

(Poznámka: existujú špecializované algoritmy na tento problém)

- Premenná x_e pre každú hranu (orientácia náhodnoa).
- $-c_e \leq x_e \leq c_e$, pre každú hranu, c_e je kapacita e .
- Súčet vo vrchole 0.
- Maximalizujeme výtok zo zdroja.

Na problém maximálneho toku existujú silno polynomiálne algoritmy, ktoré nemusia používať floating point aritmetiku. Ak sa dá problém vyriešiť akjo problém maximálneho toku, je lepšie použiť solver na problém maximálneho toku.

LP - príklady

Racionálny knapsack s viacerými obmedzeniami (napr. hmotnosť, objem)

- Premenné reprezentujú pre každý produkt množstvo ktoré dame do batoha.
- Obmedzenia zabezpečujú aby boli splnené všetky obmedzenia problému.
- Hľadáme maximálnu cenu.

Duálny LP

Problém:

- Maximizuj $c^T x$ subject to $Ax \leq b, x \geq 0$;

Duál:

- Minimizuj $b^T y$ subject to $ATy \geq c, y \geq 0$;

Silná veta o dualite: Ak má primárny problém optimálne riešenie, potom aj duálny problém má optimálne riešenie a tieto optimá sa rovnajú.

- Duálny problém k maximálnemu toku: minimálny rez (lineárne relaxovaná verzia problému + je to trochu komplikovanejšie).
- Mimochodom, maximálny tok / minimálny rez možno riešiť efektívne aj v celých číslach.

Celočíselné lineárne programovanie

Lineárne programovanie, kde môžeme navyše požadovať, že premenné sú celočíselné. Ak sú iba niektoré premenné celočíselné, volá sa to mixed integer programming.

- NP-ťažký problém.
- V mnohých prípadoch však ILP solvre môžu byť dostačujúce pre danú aplikáciu.

ILP - príklad

Knapsack

- Každý objekt môžeme vložiť do batohu celočíselný počet krát
- Celková hmotnosť musí byť menšia ako kapacita batoha.

Podobne vieme riešiť aj knapsack s viacerými obmedzeniami (napr. hmotnosť a objem).

LP solvery - algoritmy

- V niektorých prípadoch (napr. maximálny tok s celočíselnými kapacitami, maximálne párovanie) je garantované, že nejaké optimálne riešenie LP je celočíselné.
- Branch and bound.
- ...

Branch and bound

Prístup k riešeniu výpočtovo ťažkých optimalizačných problémov. V podstate ide o backtracking s prehľadávaním do hĺbky. Obsahuje:

- Heuristiku - zabezpečí, že sa rýchlo dostaneme k “dobrým” riešeniam.
- Dolný odhad (v prípade minimalizácia) pre najlepšie možné riešenie v danej vetve.

Ak heuristika nájde rýchlo dobré riešenie, vetvu môžeme zahodiť ak dolný odhad pre optimálne riešenie v nej je väčší ako riešenie, ktoré sme už našli

- Pri ILP - odhad - lineárna relaxácia problému.

Branch and bound prístup ale môžeme použiť aj mimo ILP.

Branch and bound - príklad TSP

Problém obchodného cestujúceho (nejdeme riešiť ILP).

- Na vstupe máme graf, ktorého hrany sú ohodnotené kladnými celými číslami - tieto hodnoty budeme považovať za dĺžku hrany.
- Cieľ je nájsť uzavretý sled minimálnej celkovej dĺžky.

Heuristika môže napríklad rozširovať ťah o najkratšie spojnice s nejakým vrcholom.

Branch and bound - príklad TSP

Odhad - použijeme LP relaxáciu TSP. Graf doplníme na kompletný (vzdialenosti jednoducho vypočítame). Zaujímajú nás iba Hamiltonovské kružnice.

- Premenná pre každú hranu.
- Súčet okolo každého vrchola je presne dva.
- Súčet na každom reze je aspoň dva.
- Hodnoty premenných sú medzi 0 a 1.

Ak by hodnoty premenných boli celé čísla, optimalizujeme cez hamiltonovské kružnice.

- LP má exponenciálne veľa obmedzení, problém je však riešiteľný v P ((delayed column generation)).
- Riešenie je dolným odhadom na dĺžku TSP.

Branch and bound - príklad TSP

- Backtrackom rozširujeme optimálnu cestu (pričom medzi viacerymi možnosťami ako pokračovať si vyberáme podľa heuristiky).
- Ak vyberieme vhodnú heuristiku, nájdeme pomerne rýchlo “dost” dobré riešenie.
- Pri každom nasledujúcom vetvení vypočítame dolný odhad na dĺžku TSP.
- Ak tento dolný odhad je horší ako najlepší nájdenny sled, nemusíme danú vetvu prehľadávať.

Ako formulovať problémy.

Podmienka $x \in [a, b] \cup [c, d]$, $a \leq b < c \leq d$.

- Nová premenná $x' \in \{0, 1\}$.
- $a \leq x \leq d$, $:$).
- $x \leq b + x'(d - b)$. Podmienka je relevantná práve keď $x' = 0$.
- $x \geq c + (1 - x')(a - c)$. Podmienka je relevantná práve keď $x' = 1$.

Ak niektoré z a, b, c, d sú premenné, nahradíme $(a - c)$ a $(d - b)$ vhodnými konštantami.

Celočíselné lineárne programovanie - príklady

- Knapsack
- Dominujúca množina

Vrcholové farbenie

- Pridáme binárnu premenú a_{ij} (celočíselná, $0 \leq a_{ij} \leq 1$), ktorá pre každú hranu určuje, ktorý vrchol má väčšiu farbu.
- $a_{ij} = 1 - a_{ji}$, $x_i - x_j \geq 1 - 2na_{ij}$ (n je počet vrcholov, nerovnosť je v prípade $a_{ij} = 1$ irelevantná).

Celočíselné lineárne programovanie - príklady

Už sme videli príklad na TSP (ten ale bol exponenciálny).

Uvažujme kompletný graf.

- Premenná - x_{ij} sled pokračuje z i do j .
- Do každého vrcholu vchádzame, z každého vychádzame práve raz.
- Premenné počítajúce koľko vrcholov sme už navštívili,
 $t_j \leq t_i + 1 + n(1 - x_{ij})$ (ak $j = 0$, použijeme $t'(0)$ miesto $t(0)$).
- $t_0 = 0$, $t'_0 = n$.

Iné solvery, ktoré používame na riešenie problémov - polynomiálne

Rozšírenia lineárneho programovania:

- Kvadratické programovanie, ak kvadratické členy tvoria kladne semidefinitnú maticu
- Semidefinitné programovanie.
- ...

Iné solvery, ktoré používame na riešenie problémov - nepolynomialne

- Constraint satisfaction problem.
- Hamiltonovskosť.
- ...

Resources |

- [Wikipedia - Linear programming](#)
- [Wikipedia- Branch and bound](#)