Úvod do matematickej logiky
LP, ILP, B&B

Robert Lukoťka
lukotka@dcs.fmph.uniba.sk
www.dcs.fmph.uniba.sk/~lukotka

M-255

Lineárne programovanie

- Množina premenných: $x_1, \dots x_n$.
- Cieľová funkcia: $c_1 x_1 + c_2 x_2 + \dots c_n x_n$
- Obmedzenia, nerovnosti: $a_{1i} x_1 + a_{2i} x_2 + \cdots + a_{ni} x_n \leq b_i$, pre $i \in \{1, \dots .j\}$.
- Obmedzenia, rovnosti : $c_{1i} x_1 + c_{2i} x_2 + \cdots + c_{ni} x_n = d_i$, pre $i \in \{1, \dots .k\}$.

V maticovom zápise

$$\max\{\mathrm{c}^T \mathrm{x} \mid A\mathrm{x} \leq \mathrm{b} \wedge C\mathrm{x} = \mathrm{d}\},$$

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 vnútrom polytopu.
  - Zložitosť - rôzna, $O(n^3)$ aritmetických operácií, v niektorých prípadoch aj lepšia (často závisí od algoritmu na násobenie matíc).

Lineárne programovanie, algoritmy

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.

LP - príklady

Chceme nakúpiť el. energiu. El. energia sa nakupuje po hodinách, buď každá hodina samostatne, alebo v balíkoch (napr. pracovné dni od 8:00 do 16:00). Každá hodina a každý balík má svoju jednotkovú cenu. Ako nakúpiť energiu čo najlacnejšie?

- Premenné reprezentujú pre každý produkt (hodinu, balíček hodín) množstvo nakúpeného produktu.
- V každej hodine musíme nakúpiť správne množstvo energie.
- Nemôžeme nakúpiť záporné množstvo produktu.
- Hľadáme minimálnu 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 reišenie, 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 komplikovaniejšie).
- Mimochodom, maximálny tok / minimálny rez možno riešiť efektívene aj v celých číslach.

Celočíselné lineárne programovanie

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

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

LP solvery - algoritmy

- V niektorých prípadoch (napr. maximálny tok s celočíselnými kapacitami, maximálne párovanie) je garantované, že nejáké 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 prípad minimalizácia) pre najlepšie možné riešenie v danej vetve.

Ak heuristika nájde rýchlo dobré riešenie, vervu 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 "dosť" dobré riešenie.
- Pri každom nasledujúcom vetvení vypočítame dolný odhad na dĺžku TSP.
- Ak tento dolný odhad je horší ako najlepší nájdený 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 $x' = 1$.

Celočíselné lineárne programovanie - príklady

- Knapsack
- Dominujúca množina

Vrcholové farbenie

- Pridáme binárnu premenú a_{ij} (celočíselná, $0 \leq x \leq 1$), 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 - nepolynomiálne

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

Resources I

- Wikipedia - Linear programming
- Wikipedia- Branch and bound