Cvičení 11

Jednokriteriální optimalizace

Jan Přikryl

ČVUT FD

20. dubna 2020

CVX

Stručný přehled

CVX je knihovna pro Matlab, podporující *disciplinované konvexní programování* – jde o širší škálu úloh, zahrnující lineární i kvadratické programování respektive všechny rozumně zadané *konvexní* úlohy.
Domovská stránka: `http://www.cvxr.com/cvx/`

Úkoly:

(a) Stáhněte si CVX z adresy `http://web.cvxr.com/cvx/cvx-w64.zip` a rozbalte archiv do libovolného vlastního adresáře `<cvx_root>` (nerozbalujte CVX rovnou mezi toolboxy Matlabu!)

(b) V Matlabu se přepněte do adresáře `<cvx_root>` a inicializujte CVX příkazem `cvx_setup`, například

```
cd C:\personal\cvx
cvx_setup
```

Příklad 1

Téměř původní Dantzigova úloha na lineární programování

Máme dvě konzervárny sardinek – v Seattlu a San Diegu – a tři hlavní odbytiště na území USA: New York, Chicago a Topeku. Kozervárny mají určitou kapacity produkce, odbytiště mají určitou poptávku.

	New York	Chicago	Topeka	Kapacita
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
Poptávka	325	300	275	

Vzdálenosti jsou v tisících mil, přepravné je 90 USD za bednu a 1000 mil.

Otázka

Jak uspokojit poptávku s nejnižšími přepravními náklady? (Předpokládáme, že nabídka je větší, než poptávka.)

Příklad 1

Pokračování

Analyzujeme:

- Máme celkem 6 přepravních aktivit:
  SEA $\to$ NYC, SEA $\to$ ORD, SEA $\to$ MCI, SAN $\to$ NYC, SAN $\to$ ORD, SAN $\to$ MCI
- Objem dodaných zásilek z SEA a SAN nesmí překročit počet beden na skladě.
- Je třeba uspokojit všechny odběratele.

Stavový vektor LP je vektor aktivit, $\mathbf{x} = (x_1, x_2, \ldots, x_6)^T$. Minimalizujeme cenu přepravy

$$\mathbf{x}_{\text{opt}} = \arg\min_{\mathbf{x}} \mathbf{c}\mathbf{x}$$

Kde vezmeme $\mathbf{c}$? Jaké budou omezující podmínky na $\mathbf{x}$?

Příklad 1

Pokračování

Vytvořte funkci `cvxtest2()` implementující výše uvedený problém v CVX. Po zadání parametrů `c`, `supply_sea`, `supply_san`, `demand_nyc`, `demand_ord` a `demand_mci` lze v CVX lineární program zapsat jako

```
cvx_begin
    variable x(6)
    minimize(c*x)
    subject to
        x >= 0 % nelze vozit zaporny pocet
        x(1)+x(2)+x(3) <= supply_sea
        x(4)+x(5)+x(6) <= supply_san
        x(1)+x(4) >= demand_nyc
        x(2)+x(5) >= demand_ord
        x(3)+x(6) >= demand_mci
cvx_end
```

Příklad 2

Parametry jako matice

Postup uvedený výše je z hlediska zápisu velmi složitý: Tabulka se stavem systému je transformována na vektor, a všechny podmínky jsou vypsané jednotlivě. Zkusme nyní celý proces upravit tak, že stav bude uložen v matici $\mathbf{X}$ o rozměrech (2×3) – sloupce matice odpovídají přepravě do jednotlivých odbytišť, řádky matic pak přepravě z jednotlivých továren.

Otázka

Jak se změní původní minimalizační kritérium? Pro vektory $\mathbf{c}$ (řádkový) a $\mathbf{x}$ (sloupcový) jsme minimalizovali $\mathbf{cx}$, jak to bude v případě matic $\mathbf{C}$ a $\mathbf{X}$? Potřebujeme, aby výsledek operace bylo jedno reálné číslo, odpovídající původnímu vektorovému zápisu.

Příklad 2

Parametry jako matice – matice vah **C**, poptávka a nabídka

Vytvořte funkci `cvxtest2m()` implementující výše uvedený problém v CVX. Stav **X** je nyní matice, proto i váhy (ceny přepravy) musí mít formu matice:

```
C = [ 2.5 1.7 1.8;
      2.5 1.8 1.4 ];
```

Pro jednodušší porovnání uložíme i nabídku a poptávku do vektorů-

```
supply = [ .... ];  % vektor nabídek
demand = [ .... ];  % vektor poptávek
```

Otázka

Jeden z vektorů `supply` a `demand` bude patrně muset být řádkový, jeden sloupcový. Který to bude a proč?

Příklad 2

Parametry jako matice – kód pro CVX

Máme zadáno `C`, `supply` a `demand`, nyní formulujeme optimalizační problém ve formátu pro CVX. Do následující šablony doplňte potřebné části kódu:

```
cvx_begin
  variable X(2,3)
  minimize(....)  % musí být nakonec skalární hodnota
  subject to
    .... >= 0  % počty nesmí být záporné
    sum(....,2) <= ....  % podmínky pro nabídku a poptávku
    sum(....,1) >= ....
cvx_end
```

Příklad 2

Parametry jako matice – řešení

CVX najde řešení po několika málo iteracích, minimální hodnota optimalizačního kritéria (cenové funkce) je

```
Optimal value (cvx_optval): +1707.5
```

Prvky matice řešení **X** odpovídají původním prvkům vektoru **x**:

```
x =
   25.0000  300.0000    0.0000
  300.0000    0.0000  275.0000
```

Co chybí

Nezmínili jsme se o tom

- jak místo vektoru **c** použít matici **C** a stav v maticové formě
- jak řešit úlohy kde máme možnost volby (třeba přepravního prostředku)
- jak optimalizovat plánovací úlohy (služby, stroje, soupravy)
- ...