

# Cvičení 14

## Jednokriteriální optimalizace

Jan Příklad

ČVUT FD

15. března 2018

# 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

	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.

Jak uspokojit poptávku s nejnižšími přepravními náklady?

# Příklad 1

## Pokračování

Analyzujeme:

- ▶ Máme celkem 6 přepravních aktivit:  
SEA → NYC, SEA → ORD, SEA → MCI, SAN → NYC,  
SAN → ORD, SAN → 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,  $\mathbf{x} = (x_1, x_2, \dots, x_6)^T$ .

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
```

# Co chybí

Nezmínili jsme se o tom

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