Implementace simplexové metody

Tomáš Werner

Cíl

Chceme vyřešit úlohu lineárního programování

$$\min\{\, \mathbf{c}^T\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n,\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}. \tag{1}$$

Předpokládáme, že $\mathbf{A} \in \mathbb{R}^{m\times n}$ má hodnost m. Nepředpokládáme ale, že $\mathbf{b} \geq \mathbf{0}$.

Implementujte dvoufázovou simplexovou metodu na řešení úlohy (1) jako matlabskou funkci

```
x = simplex(c,A,b)
```

kde `c` je matice $n \times 1$, `A` je matice $m \times n$ a `b` je matice $m \times 1$. Hodnota `x` je následující:

- Pokud má úloha optimální řešení, je `x` matice $n \times 1$ s libovolným optimálním řešením.
- Pokud je úloha nepřípustná nebo neomezená, `x` bude prázdná matice.

Doporučujeme nejdříve implementovat v samostatné matlabské funkci základní simplexovou metodu, která řeší úlohu (1) za předpokladu, že $\mathbf{A}$ obsahuje standardní bázi a $\mathbf{b} \geq \mathbf{0}$. Pomocí této funkce pak implementujete požadovanou funkci `simplex`.

Úkoly

Nejprve implementujte popsanou funkci `simplex`. Výstupem tohoto úkolu nebude žádný text ve zprávě, pouze odevzdaný kód.

Poté pomocí této funkce vyřešte následující úlohy. Výstupem každé úlohy bude argument a hodnota optimálního řešení:

1. Úloha

$$\begin{aligned} \min \quad & x_1 - x_2 + 2x_3 \\ \text{za podmínek} \quad & -3x_1 + x_2 + x_3 = 4 \\ & x_1 - x_2 + x_3 = 3 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

2. Úloha

$$\begin{aligned} \min \quad & -x_1 - 3x_3 + x_4 \\ \text{za podmínek} \quad & x_1 + 2x_2 + 3x_3 = 15 \\ & 2x_1 + x_2 + 5x_3 = 20 \\ & x_1 + 2x_2 + x_3 + x_4 = 10 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

3. Úloha *Jistá výhra* z předchozího cvičení.
4. Úloha *Minimaxní prokládání lineární funkce* z předchozího cvičení.

Konečnost algoritmu: doplňkové úkoly pro dobrovolníky

Vyřešit tyto úkoly není povinné a nebude bodováno. Jsou určeny pro ty, kteří si chtějí se simplexovou metodou více pohrát.

U základní simplexové metody není bez dodatečné péče zaručena její konečnost. V případě degenerované úlohy může algoritmus cyklit mezi několika degenerovanými bázemi příslušnými stejnému bázovému řešení. Hodnota účelové funkce se v tom případě nezlepšuje a algoritmus se nikdy nezastaví.

Zamyslete se nad následujícímu úkoly:

1. Proveďte modifikaci algoritmu, která zajistí, že se pro žádnou vstupní úlohu nezacyklí. Pomocí internetu prostudujte běžně používaná anticyklící pivotová pravidla a jedno z nich implementujte.
2. Naleněte příklad úlohy LP a pivotové pravidlo, které povedou k cyklení.