1 Zrewidowany algorytm sympleks

Rozważmy zadanie programowania liniowego w postaci standardowej

$$\mathbb{X} = \left\{ \begin{array}{l} \min\ \boldsymbol{c}^T\boldsymbol{x} \\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{x} \geqslant \boldsymbol{0}, \end{array} \right. \tag{1}$$

gdzie $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m \leqslant n$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{b} \geqslant \boldsymbol{0}$ a $\boldsymbol{x} \in \mathbb{R}^n$ wektorem zmiennych decyzyjnych.

Przypomnijmy szkic (prymalnego) algorytmu sympleks. Niech $\boldsymbol{B}$ będzie bazą, $\boldsymbol{x}^{\boldsymbol{B}} = \boldsymbol{B}^{-1}\boldsymbol{b}$ a $\boldsymbol{x}^T = [\boldsymbol{x}^{\boldsymbol{B}}, \boldsymbol{0}]$ będzie rozwiązaniem bazowym zadania prymalnego (1). Takie rozwiązanie może być, np. wyznaczone za pomocą metody dwóch faz. Niech $J_{\boldsymbol{B}}$ będzie zbiorem indeksów kolumn (zmiennych) bazowych. Załóżmy, że $J_{\boldsymbol{B}}$ jest uporządkowany (zapewnienie, że algorytm sympleks skończy działanie - nie wpadnie w cykl, zob. twierdzenie 3, wykłady nr 4 i 5).

Krok 0 Oblicz $\boldsymbol{y}^j$ przedstawienia kolumn $\boldsymbol{A}_j$ w bazie $\boldsymbol{B}$, $\boldsymbol{y}^j = \boldsymbol{B}^{-1}\boldsymbol{A}_j$.

Krok 1 Sprawdź kryterium optymalności, tj. $(\forall j \notin J_{\boldsymbol{B}})(c_j - (\boldsymbol{c}^{\boldsymbol{B}})^T\boldsymbol{y}^j \geqslant 0)$. Jeśli kryterium jest spełnione, to STOP $\boldsymbol{x}$ jest rozwiązaniem optymalnym.

W przeciwnym przypadku wybierz kolumnę $\boldsymbol{A}_j$ taką, że $k \notin J_{\boldsymbol{B}}$ i $c_k - (\boldsymbol{c}^{\boldsymbol{B}})^T\boldsymbol{y}^k < 0$, która wchodzi i do bazy (lub zmienna x_k).

Krok 2 Wybierz kolumnę $\boldsymbol{B}_{i^*}$ (zmienną x_{i^*}) wychodzącą z bazy $\boldsymbol{B}$,

$$\frac{x_{i^*}^{\boldsymbol{B}}}{y_{i^*}^k} = \min\left\{ \frac{x_i^{\boldsymbol{B}}}{y_i^k} \ :\ y_i^k > 0, i \in J_{\boldsymbol{B}} \right\}.$$

Jeśli $\boldsymbol{y}^k \leqslant \boldsymbol{0}$, wówczas nie ma rozwiązania optymalnego skończonego, STOP.

Krok 3 Uaktualnij: $J_{\boldsymbol{B}}$, rozwiązanie bazowe $\overline{\boldsymbol{x}}$, bazę $\overline{\boldsymbol{B}}$ i $\overline{\boldsymbol{y}}^j$ przedstawienia kolumn macierzy w bazie $\overline{\boldsymbol{B}}$

$\boldsymbol{B} := \overline{\boldsymbol{B}}$, $\boldsymbol{x} := \overline{\boldsymbol{x}}$, $\boldsymbol{y}^j := \overline{\boldsymbol{y}}^j$. Idź do Krok 1.

W powyższym algorytmie wymagane jest uaktualnienie $(m+1)\times(n+1)$ elementów tablicy sympleksowej. Główny wysiłek obliczeniowy jest w Kroku 1 i Kroku 3. Okazuje się, że

- możemy uaktualniać tylko $(m+1)\times(m+1)$ elementów tablicy sympleksowej - kluczem jest posiadanie informacji o $\boldsymbol{B}^{-1}$. Mając informacje o odwrotności macierzy bazowej możemy wyznaczyć $\boldsymbol{x}^{\boldsymbol{B}}$, $\boldsymbol{y}^j$ przedstawienia kolumn macierzy $\boldsymbol{A}$ w bazie $\boldsymbol{B}$, wskaźniki optymalności:

$$c_j - z_j = c_j - (\boldsymbol{c}^{\boldsymbol{B}})^T\boldsymbol{y}^j = c_j - (\boldsymbol{c}^{\boldsymbol{B}})^T\boldsymbol{B}^{-1}\boldsymbol{A}_j = c_j - \boldsymbol{\pi}^T\boldsymbol{A}_j;$$

- możemy wyznaczać wskaźniki optymalności (Krok 1) tak długo aż znajdziemy pierwszą kolumnę $k \notin J_{\boldsymbol{B}}$, dla której $c_j - (\boldsymbol{c}^{\boldsymbol{B}})^T \boldsymbol{y}^j < 0$ i przerwać przeszukiwanie - wiemy wtedy, że aktualne rozwiązanie nie jest optymalne.

  Zatem możemy wyznaczać wskaźniki optymalności i $\boldsymbol{y}^j$ tylko wtedy kiedy potrzeba.

Biorąc pod uwagę powyższe spostrzeżenia nowy, *zrewidowany*, algorytm sympleks w każdej iteracji będzie tylko pamiętał $\boldsymbol{B}^{-1}$, $\boldsymbol{x}^{\boldsymbol{B}}$, $\boldsymbol{\pi}$ i wartość funkcji celu $(\boldsymbol{c}^{\boldsymbol{B}})^T \boldsymbol{x}^{\boldsymbol{B}}$.

Załóżmy, że pierwsza tablica simpleksowa zawiera macierz jednostkową $\mathbf{I}$, która znajduje się po lewej stronie (w pierwszych m kolumnach). Zatem mamy początkową bazę $\boldsymbol{B} = \mathbf{I}$ i $\boldsymbol{x}^{\boldsymbol{B}} = \boldsymbol{b}$. Ponadto zakładamy, że współczynniki funkcji celu zmiennych odpowiadającym kolumnom macierzy jednostkowej są równe zero. W dalszej części tego wykładu pokarzemy jak spełnić ten warunek. Tę lewą cześć tablicy sympleksowej, z macierzą $\mathbf{I}$, nazwiemy $\texttt{CARRY}^{(0)}$ (początkowa tablica), która zawiera $(m+1) \times (m+1)$ elementów (zob. tabela 1). W kolejnych iteracjach zrewidowany algorytm będzie uaktualniał tylko tablicę $\texttt{CARRY}$. Tablica w iteracji r, $\texttt{CARRY}^{(r)}$, jest przedstawiona w tabeli 1.

Tabela 1: Początkowa tablica $\texttt{CARRY}^{(0)}$ i tablica $\texttt{CARRY}^{(r)}$ w iteracji r algorytmu zrewidowanego.

$\texttt{CARRY}^{(0)}$

$\boldsymbol{x}^{\boldsymbol{B}}$	$\mathbf{I}$
$(\boldsymbol{c}^{\boldsymbol{B}})^T \boldsymbol{x}^{\boldsymbol{B}}$	$0 \quad 0 \quad \cdots \quad 0 \quad 0$

$\longrightarrow$

$\texttt{CARRY}^{(r)}$

$\boldsymbol{x}^{\boldsymbol{B}}$	$\boldsymbol{B}^{-1}$
$(\boldsymbol{c}^{\boldsymbol{B}})^T \boldsymbol{x}^{\boldsymbol{B}}$	$-\boldsymbol{\pi}$

Szkic

Przedstawimy teraz kluczowy fragment zrewidowanego algorytmu sympleks. Okazuje się, że zrewidowany algorytm potrzebuję następujących informacji: tablicę zawierającą oryginalną macierz $\boldsymbol{A}$, informację o kolumnach bazowych $J_{\boldsymbol{B}}$ oraz aktualną tablicę $\texttt{CARRY}^{(r)}$.

Krok 1 (Pricing operation) Obliczaj wskaźniki optymalności $c_j - \boldsymbol{\pi}^T \boldsymbol{A}_j$ dla kolumn niebazowych $j \notin J_{\boldsymbol{B}}$ aż znajdzie się kolumna $k \notin J_{\boldsymbol{B}}$, dla której $c_k - \boldsymbol{\pi}^T \boldsymbol{A}_k < 0$ (kolumna $\boldsymbol{A}_k$ wchodzi do bazy, rozwiązanie nie jest optymalne). Jeśli $(\forall j \notin J_{\boldsymbol{B}})(c_j - (\boldsymbol{c}^{\boldsymbol{B}})^T \boldsymbol{y}^j \geqslant 0)$, to aktualne rozwiązanie jest optymalne, STOP.

Krok 2 (Column generation) Oblicz przestawienie $\boldsymbol{y}^k = \boldsymbol{B}^{-1}\boldsymbol{A}_k$ kolumny $\boldsymbol{A}_k$ w bazie. Jeśli $k \in [m]$, to przedstawienie jest już obliczone i jest zapamiętane w $\texttt{CARRY}^{(r)}$.

Wybierz kolumnę $\boldsymbol{B}_{i^*}$ (zmienną x_{i^*}) wychodzącą z bazy $\boldsymbol{B}$,

$$\frac{x^{\boldsymbol{B}}_{i^*}}{y^k_{i^*}} = \min\left\{\frac{x^{\boldsymbol{B}}_i}{y^k_i} \;:\; y^k_i > 0, i \in J_{\boldsymbol{B}}\right\}.$$

Jeśli $\boldsymbol{y}^k \leqslant \mathbf{0}$, wówczas nie ma rozwiązania optymalnego skończonego, STOP.

Krok 3 Uaktualnij $\texttt{CARRY}^{(r)}$ aby otrzymać $\texttt{CARRY}^{(r+1)}$. Uaktualnienie dokonujemy tak jak dla algorytmu sympleks korzystając z policzonego już przedstawienia $\boldsymbol{y}^k$.

Krok 4 (Update basis) Uaktualnij indeksy kolumn należących do bazy $J_{\boldsymbol{B}}$ (indeksy są uporządkowane).

Kluczowym krokiem jest Krok 2: mając $\boldsymbol{B}^{-1}$ mamy wystarczającą ilość informacji, aby wygenerować każdy element tablicy sympleksowej.

Główne zalety zrewidowanej wersji algorytmu

- Nie obliczamy wszystkich wskaźników optymalności odpowiadającym kolumnom niebazowym. Wybieramy pierwszą, dla której wskaźnik jest ujemny. Ponieważ $J_{\boldsymbol{B}}$ jest uporządkowany spełniamy założenie twierdzenia 3 (zob. wykłady nr 4 i 5), które zapewnia, że algorytm sympleks wykonuje skończoną liczbę kroków.
- Krok 1, czyli obliczanie wskaźników, korzysta z oryginalnych kolumn macierzy $\boldsymbol{A}$. W wielu przypadkach macierz ograniczeń jest macierzą rzadką (zawiera dużo elementów zerowych), np. w zagadnieniach sieciowych (problem najkrótszej ścieżki, problem maksymalnego przepływu). W tym przypadku możliwa jest redukcja złożoności obliczeniowej i pamięciowej.

Metoda dwóch faz dla zrewidowanego algorytmu

Jak już wiemy metoda dwóch faz służy do wyznaczenia początkowego rozwiązania bazowego dopuszczalnego i polega na rozwiązaniu w pierwszej fazie, za pomocą algorytmu sympleks, następującego problemu programowania liniowego:

$$\begin{aligned} &\min \sum_{i\in[m]} 1\, x^A_i \\ &\begin{cases} \mathbf{I}\boldsymbol{x}^A + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{x} \geqslant \mathbf{0} \\ \boldsymbol{x}^A \geqslant \mathbf{0}, \end{cases} \end{aligned} \tag{2}$$

gdzie $\boldsymbol{x}^A \in \mathbb{R}^m$ jest wektorem sztucznych zmiennych. Aby zastosować zrewidowany algorytm sympleks do zadania (2) musimy zmodyfikować to zadanie i skonstruować równoważne. Zauważ, że współczynniki funkcji celu odpowiadające kolumnom macierzy jednostkowej (przy zmiennych sztucznych) są równe 1, a powinny być równe 0. Wystarczy od funkcji celu (wiersza zerowego) odjąć wszystkie wiersze macierzy $\mathbf{I}\,\boldsymbol{x}^A + \boldsymbol{A}\boldsymbol{x}$. Otrzymujemy następujące zadanie programowania liniowego:

$$
\begin{aligned}
&\min \sum_{i\in[m]} 0\,x_i^A - \sum_{i\in[m]} a_{i1}x_1 - \cdots - \sum_{i\in[m]} a_{in}x_n \\
&\begin{cases}
\mathbf{I}\,\boldsymbol{x}^A + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\
\boldsymbol{x} \geqslant \boldsymbol{0} \\
\boldsymbol{x}^A \geqslant \boldsymbol{0},
\end{cases}
\end{aligned}
\tag{3}
$$

gdzie $d_j = -\sum_{i\in[m]} a_{ij}$ są współczynnikami funkcji celu oryginalnych zmiennych decyzyjnych x_j, $j \in [n]$, rozwiązywanego w pierwszej fazie zadania (3) za pomocą zrewidowanego algorytmu sympleks a wskaźniki optymalności odpowiadające kolumnom oryginalnych zmiennych są postaci: $d_j - \boldsymbol{\pi}^T\boldsymbol{A}_j$, $j \in [n]$.

W drugiej fazie wracamy do oryginalnych współczynników c_j zmiennych x_j oraz generujemy $-\boldsymbol{\pi}^T = -(\boldsymbol{c}^{\boldsymbol{B}})^T\boldsymbol{B}^{-1}$ (współczynniki funkcji celu zmiennych sztucznych są zerowe). Zauważ, że $-\boldsymbol{\pi}^T$ jest wskaźnikiem optymalności kolumn odpowiadających macierzy jednostkowej (zmiennych sztucznych), $-\pi_j = 0 - (\boldsymbol{c}^{\boldsymbol{B}})^T\boldsymbol{B}^{-1}\boldsymbol{e}_j$.

Postać iloczynowa macierzy odwrotnej $\boldsymbol{B}^{-1}$

Wspomnimy jeszcze o jednym udoskonaleniu algorytmu sympleks, czyli sposobie pamiętania informacji o macierzy $\boldsymbol{B}^{-1}$ w kolejnych iteracjach. Zauważmy, że macierz bazowa $\boldsymbol{B}$ różni się tylko jedną kolumną od macierzy bazowej w kolejnej iteracji $\overline{\boldsymbol{B}}$. Zatem niewielkie modyfikacje macierzy odwrotnych w kolejnych iteracjach algorytmu wykonuje się za pomocą przekształceń elementarnych zastosowanych do $\boldsymbol{B}^{-1}$. Mianowicie, jeśli $\boldsymbol{B}^{-1}$ jest aktualną odwrotnością macierzy bazowej, to odwrotność $\overline{\boldsymbol{B}}^{-1}$ w kolejnej iteracji wyznaczana jest następująco:

$$\overline{\boldsymbol{B}}^{-1} = \boldsymbol{E}\boldsymbol{B}^{-1},$$

gdzie $\boldsymbol{E}$ jest macierzą przekształcenia (kolumna $\boldsymbol{A}_k$ wchodzi do bazy, kolumna $\boldsymbol{A}_{i*}$ wychodzi z bazy) jest postaci:

$$\boldsymbol{E} = \begin{bmatrix} 1 & & \eta_1 & 0 & \cdots & 0 \\ & \ddots & & & & \\ & & \eta_{i^*} & 0 & & \\ & & \eta_{i^*+1} & 1 & & \\ & & \vdots & & \ddots & \\ 0 & \cdots & \eta_m & 0 & & 1 \end{bmatrix}$$

a

$$\boldsymbol{\eta}^T = \left[-\frac{y_1^k}{y_{i*}^k}, \cdots, -\frac{y_{i*-1}^k}{y_{i*}^k}, \frac{1}{y_{i*}^k}, -\frac{y_{i*+1}^k}{y_{i*}^k}, \cdots, -\frac{y_m^k}{y_{i*}^k} \right].$$

Zatem informacja o przejściu z $\boldsymbol{B}^{-1}$ do $\overline{\boldsymbol{B}}^{-1}$ pamiętana jest za pomocą $\boldsymbol{E}$ lub oszczędniej za pomocą wektora $\boldsymbol{\eta}$. Jeśli założymy, że macierz bazowa w pierwszej iteracji jest równa $\mathbf{I}$, wówczas macierz $\boldsymbol{B}^{-1}$ po r-tej iteracji możemy obliczyć

$$\boldsymbol{B}^{-1} = \boldsymbol{E}_r \boldsymbol{E}_{r-1} \cdots \boldsymbol{E}_1.$$

Jest to *postać iloczynowa macierzy odwrotnej* $\boldsymbol{B}^{-1}$. Podobnie obliczamy

$$\begin{aligned} \boldsymbol{\pi}^T &= (\cdots(((\boldsymbol{c}^{\boldsymbol{B}})^T \boldsymbol{E}_r)\boldsymbol{E}_{r-1})\cdots)\boldsymbol{E}_1, \\ \boldsymbol{y}^k &= \boldsymbol{E}_r(\cdots(\boldsymbol{E}_2(\boldsymbol{E}_1 \boldsymbol{A}_k))\cdots). \end{aligned}$$

Powyższa metoda ma zastosowanie do dużych problemów z macierzami rzadkimi.

Od czasu do czasu ciąg przekształceń elementarnych może być zbyt długi, wtedy dokonuje się *reinwersji*, jest to podyktowane względami pamięciowymi i własnościami numerycznymi (błędy numeryczne), czyli oblicza się macierz $\boldsymbol{B}^{-1}$ (również $\boldsymbol{x}^{\boldsymbol{B}}$) korzystając z oryginalnych kolumn macierzy $\boldsymbol{A}$. Następnie zastępuje się aktualny dłuższy ciąg przekształceń elementarnych krótszym, który prowadzi do macierzy $\boldsymbol{B}^{-1}$.

Uwagi na temat treści wykładu

Część wykładu została przygotowana na podstawie książek [2, 1].

Literatura

[1] Narsingh Deo, Janusz S. Kowalik, Maciej M. Sysło. *Algorytmy optymalizacji dyskretnej z programami w języku PASCAL*. Wydawnictwa Naukowego PWN, 1999.

[2] Christos H. Papadimitriou, Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications Inc., 1998.