1DR2243:A - Teoria i metody optymalizacji, ćw.
Minimalizacja w kierunku. Metoda *backtracking line search*

(opracował: Maciej Twardy, ostatnia modyfikacja: 27 listopada 2018)

1 Preliminaria

Wiele algorytmów optymalizacji opartych jest na zależności rekurencyjnej [1, 2, 3]

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s\boldsymbol{v}_k,$$

gdzie wektor $\boldsymbol{v}_k \in \mathbb{R}^n$ nazywamy kierunkiem poszukiwań (ang. *search direction*) lub kierunkiem aktualizacji (ang. *update direction*), zaś $s > 0$ jest długością kroku (ang. *step size*).
Po wyznaczeniu kierunku poszukiwań, można wyznaczyć długość kroku (ang. *stepsize selection*). W tym celu rozpatrujemy obcięcie dziedziny funkcji f_0 do prostej wyznaczonej przez wektor $\boldsymbol{v}_k$ zaczepiony w punkcie $\boldsymbol{x}_k$

$$\phi(s) = f_0(\boldsymbol{x}_k + s\boldsymbol{v}_k), \quad s > 0$$

Oczywiście ϕ jest funkcją zmiennej skalarnej s, $\phi : \mathbb{R} \to \mathbb{R}$, zachodzi również

$$\phi(0) = f_0(\boldsymbol{x}_k)$$

Wybór odpowiedniej długości kroku sprowadza się do takiego wyboru wartości $s > 0$ aby

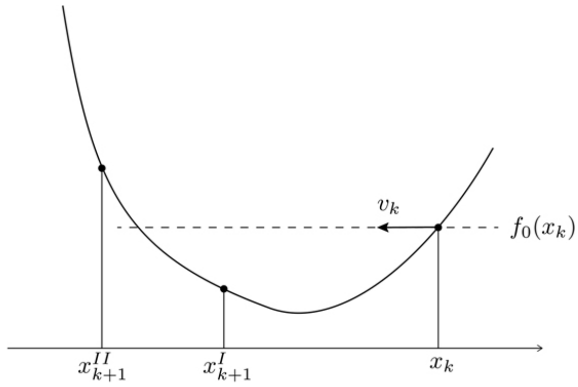
$$\phi(s) < \phi(0)$$

Można poszukiwać minimum funkcji ϕ, innymi słowy rozwiązać zadanie

$$s^* = \arg\min_{s>0} \phi(s)$$

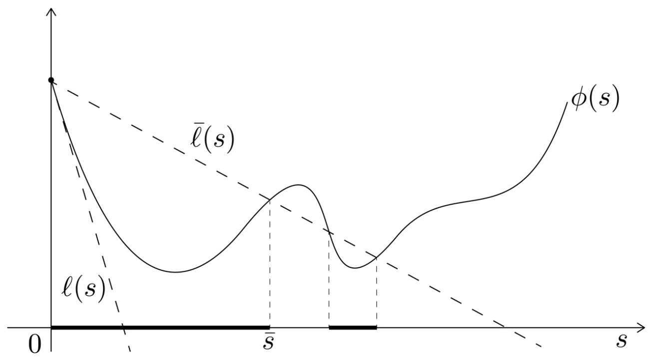
Metodę tę nazywamy dokładnym poszukiwaniem w kierunku (ang. *exact line search*), dostarcza ona wartość s^* zapewniającą największy spadek wartości funkcji celu f_0 w kierunku wyznaczonym przez wektor $\boldsymbol{v}_k$, jednak obliczenie s^* może być samo w sobie trudnym zadaniem i z tego powodu w praktyce rzadko używa się tej metody.

Bardziej praktycznym podejściem jest szukanie wartości s która zagawarantuje nam wystarczający spadek wartości funkcji ϕ (ang. *sufficient rate of decrease*).
Długość kroku należy dobrać ostrożnie, aby otrzymać odpowiedni spadek wartości funkcji. Na Rys. 1 kierunek spadku wartości funkcji jest „w lewo”. Jeśli $s_k > 0$ jest dostatecznie małe otrzymujemy $\boldsymbol{x}^I_{k+1}$ tak, że $f_0(\boldsymbol{x}^I_{k+1}) < f_0(\boldsymbol{x}_k)$, jeśli jednak $s_k > 0$ będzie zbyt duże to otrzymamy $\boldsymbol{x}^{II}_{k+1}$ tak, że $f_0(\boldsymbol{x}^{II}_{k+1}) > f_0(\boldsymbol{x}_k)$.



Rys. 1 Dobór długości kroku (Źródło: [3]).

Weźmy pod uwagę prostą styczną do wykresu funkcji ϕ w punkcie $(0, \phi(0))$, tak jak to przedstawiono na Rys. 2.



Rys. 2 Warunek Armijo (Źródło: [3].)

Dla $\phi(s) = f_0(\boldsymbol{x}_k + s\boldsymbol{v}_k)$ mamy

$$\phi(s) \approx \ell(s) = \phi(0) + s\phi'(0),$$

gdzie

$$\phi'(0) = \nabla f_0(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{v}_k < 0, \qquad s > 0.$$

Funkcja ℓ jest funkcją afiniczną o współczynniku nachylenia $\phi'(0)$. Dla $\alpha \in (0,\, 1)$ prosta

$$\bar{\ell}(s) = \phi(0) + s\alpha\phi'(0), \quad s > 0 \tag{1}$$

leży powyżej wykresu funkcji $\ell(s)$, co więcej, dla małych wartości $s > 0$, prosta (1) leży powyżej wykresu funkcji $\phi(s)$.
Funkcja ϕ jest ograniczona z dołu, natomiast funkcja $\bar{\ell}$ jest nieograniczona z dołu, zatem musi istnieć punkt przecięcia się ich wykresów, niech $\bar{s}$ będzie najmniejszym z takich punktów. Wszystkie wartości s dla których $\phi(s) \leqslant \bar{\ell}(s)$ zapewniają wystarczający spadek, dany przez nachylenie $\alpha\phi'(0)$ prostej $\bar{\ell}$. Zgodnie z tym warunkiem (wystarczającego spadku wartości funkcji celu), w literaturze zwanym warunkiem Armijo (ang. *Armijo condition*), dla wybranej (ustalonej) wartości $\alpha \in (0,\, 1)$, dopuszczalna długość s kroku musi spełniać nierówność

$$\phi(s) \leqslant \phi(0) + s\alpha\phi'(0), \tag{2}$$

lub, równoważnie

$$f_0(\boldsymbol{x}_k + s\boldsymbol{v}_k) \leqslant f_0(\boldsymbol{x}_k) + s\alpha\nabla f_0(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{v}_k. \tag{3}$$

Do znalezienia odpowiedniej długości s kroku często używa się tzw. metody *backtracking search*, w której początkowa wartość s jest równa pewnej ustalonej wartości s_{initial} (najczęściej $s_{\text{initial}} = 1$), następnie wartość s jest zmniejszana iteracyjnie z pewną ustaloną prędkością $\beta \in (0, 1)$, dopóki nie jest spełniony warunek Armijo (2) lub (3).

Podsumowując, metoda *backtracking search* jest postaci:

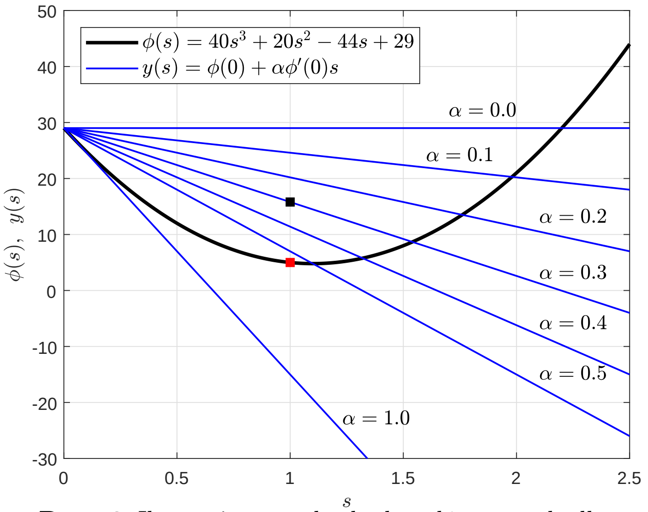
Dane: różniczkowalna funkcja $\phi(s)$, parametr $\alpha \in (0, 1)$, parametr $\beta \in (0, 1)$, początkowa wartość s_{initial} (najczęściej $s_{\text{initial}} = 1$).

1. Kładziemy $s = s_{\text{initial}}$.
2. Jeśli $\phi(s) \leqslant \phi(0) + s\alpha\phi'(0)$ to s jest poszukiwaną wartością.
3. Jeśli $\phi(s) > \phi(0) + s\alpha\phi'(0)$ to kładziemy $s \leftarrow \beta s$ i przechodzimy do punku 2 .

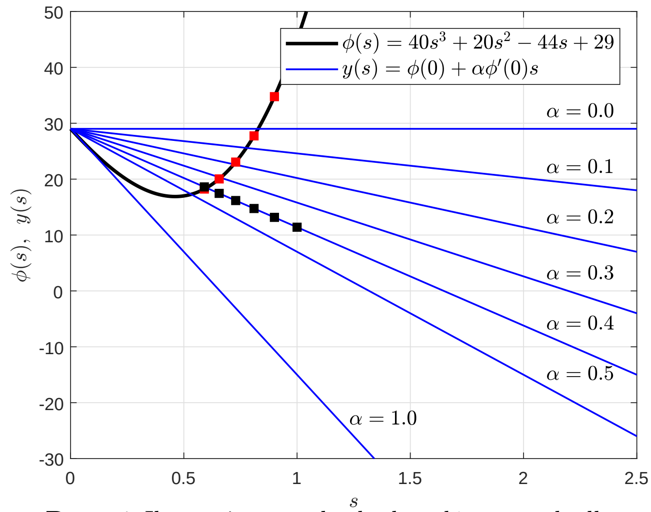
Co można zapisać w pseudokodzie:

$s \leftarrow 1$
while $\phi(s) \geqslant \phi(0) + \alpha\phi'(0)s$
    $s \leftarrow \beta s$
end

Na Rys. 3 oraz Rys. 4 przedstawiono przykładowe użycie metody *backtracking search*.



Rys. 3 Ilustracja metody *backtracking search* dla $\phi(s) = 20s^2 - 44s + 29$, $\ell(s) = \phi(0) + \phi'(0)s = 29 - 44s$, $\bar{\ell}(s) = \phi(0) + \alpha\phi'(0)s = 29 - \alpha 44s$, $\alpha = 0.3$.



Rys. 4 Ilustracja metody *backtracking search* dla $\phi(s) = 40s^3 + 20s^2 - 44s + 29$, $\ell(s) = \phi(0) + \phi'(0)s = 29 - 44s$, $\bar{\ell}(s) = \phi(0) + \alpha\phi'(0)s = 29 - \alpha 44s$, $\alpha = 0.4$, $\beta = 0.9$.

2 Zadania

Zadanie 1 Napisać skrypt w środowisku Matlab do wyznaczania, metodą *backtracking search*, długości kroku dla ustalonej funkcji jednej zmiennej. Wygenerować wykresy przedstawione na Rys. 3. [$\phi(s) = 20s^2 - 44s + 29$] i Rys. 4. [$\phi(s) = 40s^3 + 20s^2 - 44s + 29$].

Literatura

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. `http://web.stanford.edu/~boyd/cvxbook/` [Online; accessed 19.02.2016].

[2] Stephen Boyd and Lieven Vandenberghe. *Additional Exercises for Convex Optimization*. 2004. `http://web.stanford.edu/~boyd/cvxbook/` [Online; accessed 19.02.2016].

[3] G.C. Calafiore and L. El Ghaoui. *Optimization Models*. Control systems and optimization series. Cambridge University Press, October 2014.