

Henryk J. Runka

Uniwersytet Ekonomiczny w Poznaniu, Wydział Informatyki i Gospodarki
Elektronicznej, Katedra Ekonomii Matematycznej, henryk.j.runka@ue.poznan.pl

PROBLEMY Z OGRANICZENIAMI KWADRATOWYMI I OPTYMALIZACJA NA STOŻKACH

Streszczenie: W artykule przedstawiamy problem z kwadratową funkcją celu i ograniczeniami kwadratowymi oraz jego transformacje w problem optymalizacji na stożku. Transformując problem optymalizacji nieliniowej poprzez aproksymację funkcji nieliniowych za pomocą funkcji kwadratowych i funkcji liniowych, otrzymujemy nowe problemy. Pierwszym z nich jest problem optymalizacji kwadratowej z ograniczeniami kwadratowymi (i ewentualnie liniowymi), który można z kolei transformować w problem optymalizacji na stożku. We wszystkich trzech przypadkach wymienionych problemów możemy stosować algorytmy punktów wewnętrznych.

Słowa kluczowe: ograniczenia kwadratowe, problem optymalizacji kwadratowej, optymalizacja nieliniowa, optymalizacja na stożku, algorytmy prymalno-dualne punktów wewnętrznych.

Klasyfikacja JEL: C61, C63.

QUADRATICALLY CONSTRAINED QUADRATIC PROBLEMS AND CONE OPTIMIZATION

Abstract: In this paper the problem of quadratic objective functions and quadratic constraints is presented, along with its transformation into a cone optimization problem. When transforming a nonlinear optimization problem by approximating nonlinear functions by using the quadratic and linear functions, new problems appear. The first is a quadratically constrained quadratic optimization problem with quadratic constraints (and optionally, linear), which may in turn be transformed into a cone optimization problem. In all three cases, these problems can be solved using internal point algorithms.

Keywords: quadratic constrains, quadratic optimization problem, nonlinear optimization, cone optimization, primal-dual interior point algorithms.

Wstęp

Rozwiązując instancje (zadania) problemu optymalizacji nieliniowej (problem NLO), stosuje się wiele algorytmów, w których dokonuje się aproksymacji funkcji nieliniowych funkcjami liniowymi (np. GRG), funkcji optymalizowanej funkcją kwadratową, a funkcje nieliniowe w ograniczeniach funkcjami liniowymi (np. SQP) lub rozwiązuje się układ prymalno-dualny za pomocą algorytmów punktów wewnętrznych (IPNLO). Tutaj rozważamy głównie przypadek, gdy funkcja celu oraz funkcje nieliniowe w ograniczeniach są funkcjami kwadratowymi (mogą być one aproksymacjami kwadratowymi funkcji nieliniowych).

1. Problem optymalizacji kwadratowej z ograniczeniami kwadratowymi

Problem optymalizacji kwadratowej z ograniczeniami kwadratowymi (QCQO problem – Quadratically Constraint Quadratic Optimization problem) jest definiowany następująco [Anstreicher 2004]:

$$\min | \max \frac{1}{2} x^T Q_0 x + c_0^T x + d_0,$$

przy ograniczeniach

$$\begin{aligned} \frac{1}{2} x^T Q_i x + c_i^T x + d_i &\leq b_i, \quad i \in \langle I \rangle, \\ \frac{1}{2} x^T Q_i x + c_i^T x + d_i &= b_i, \quad i \in \langle E \rangle, \\ l &\leq x \leq u, \end{aligned} \tag{1}$$

gdzie:

$\langle I \rangle \cup \langle E \rangle = \{1, \dots, m\}$, $Q_i \in \mathbb{R}^{n \times n}$, $Q_i = Q_i^T$ ($i = 0, \dots, m$), $b \in \mathbb{R}^m$ ($b_i \in \mathbb{R}$), $c_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$, $x \in \mathbb{R}^n$, $l, u \in \mathbb{R}^n$, $Q_i \succeq 0(\min)$, $Q_i \preceq 0(\max)$, $i = 0, \dots, m$, przy czym $\succeq$, $\preceq$ oznaczają odpowiednio dodatnią i ujemną półokreśloność symetrycznej macierzy kwadratowej.

W szczególnym przypadku, gdy $Q_i \equiv I$, elipsoida przechodzi w kulę w $\mathbb{R}^n$ przesuniętą ze względu na $c_i^T x + d_i$.

Ponieważ ograniczenia

$$\frac{1}{2} x^T Q_i x + c_i^T x = b_i, \quad i \in \langle E \rangle$$

nie generują zbioru wypukłego, albowiem punkty dopuszczalne leżą tylko na brzegu elipsoidy (kuli), więc QCQO jest formułowany również w wersji z liniowymi ograniczeniami równościowymi [Boyd i Vandenberghe 2009, s. 152], zastąpienie układu równań

$$\frac{1}{2} x^T Q_i x + c_i^T x + d_i = b_i, \quad i \in \langle E \rangle$$

układem afinicznym (układem równań liniowych, generującym zbiór wypukły, o ile nie jest sprzeczny), postaci

$$\alpha_i^T x = b_i, \quad \alpha_i \in \mathbb{R}^n, \quad b_i \in \mathbb{R}, \quad i \in \langle E \rangle, \quad Ax = b, \quad A = [\alpha_i^T]_{i \in \langle E \rangle},$$

prowadzi do problemu

$$\min \left| \max \frac{1}{2} x^T Q_0 x + c_0^T x + d_0 \right|$$

przy ograniczeniach

$$\begin{aligned} \frac{1}{2} x^T Q_i x + c_i^T x + d_i &\leq b_i, \quad i \in \langle I \rangle, \\ \alpha_i^T x &= b_i, \quad i \in \langle E \rangle, \\ l_j &\leq x_j \leq u_j, \quad j = 1, \dots, n. \end{aligned} \tag{2}$$

Problem NLO możemy transformować w problem QCQO, zastępując nieliniowe ograniczenia nierównościowe kwadratowymi ograniczeniami nierównościowymi, a równania nieliniowe równaniami liniowymi. W dalszej części artykułu rozważamy przypadek minimalizacji funkcji celu.

Rozważmy problem optymalizacji nieliniowej postaci [Runka 2009, s. 205]:

$$\min f(x)$$

przy ograniczeniach:

$$\begin{aligned} g_i(x) &\leq 0, \quad i = 1, \dots, m_{\langle I \rangle}, \\ h_i(x) &= 0, \quad i = 1, \dots, m_{\langle E \rangle}, \\ l_j &\leq x_j \leq u_j, \quad j = 1, \dots, n, \end{aligned} \tag{3}$$

które można zapisać równoważnie w postaci wektorowo-macierzowej:

$$\begin{aligned} g(x) &\leq 0, \\ h(x) &= 0, \\ -x + l &\leq 0, \\ x - u &\leq 0. \end{aligned} \tag{4}$$

Stosując algorytmy punktów wewnętrznych, funkcję $f(x)$ zastępuje się funkcją barierową, która zawiera również zmienne swobodne postaci:

$$\begin{aligned} \min \Phi(x, y, v, w; \mu) &= \\ &= f(x) - \mu \sum_{i=1}^{m_{\langle I \rangle}} \ln(y_i) - \mu \sum_{j=1}^n \ln(v_j) - \mu \sum_{j=1}^n \ln(w_j), \quad \mu \in \mathbb{R}_{++}. \end{aligned} \tag{5}$$

Zmienne swobodne w (5) występują w ograniczeniach nierównościowych i w ograniczeniach dolnych i górnych:

$$\begin{aligned} g_i(x) + y_i &= 0, \quad i = 1, \dots, m_{\langle I \rangle}, \\ h_i(x) &= 0, \quad i = 1, \dots, m_{\langle E \rangle}, \\ -x_j + v_j + l_j &= 0 \quad (x_j - v_j = l_j), \quad j = 1, \dots, n, \\ x_j + w_j - u_j &= 0 \quad (x_j + w_j = u_j), \quad j = 1, \dots, n. \end{aligned} \tag{6}$$

Zatem $y \in \mathbb{R}^{m_{\langle I \rangle}}$, $v, w \in \mathbb{R}^n$.

W przypadku problemu (3) funkcja Lagrange'a jest postaci

$$\begin{aligned} L(x; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}) &= \\ &= f(x) + \sum_{i=1}^{m_{\langle l \rangle}} \pi_i g_i(x) + \sum_{i=1}^{m_{\langle E \rangle}} \rho_i h_i(x) + \sum_{j=1}^n s_{\langle l \rangle j} (-x_j + l_j) + \sum_{j=1}^n s_{\langle u \rangle j} (x_j - u_j), \end{aligned} \quad (7)$$

a jej gradient

$$\begin{aligned} \nabla_x L(x; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}) &= \\ &= \nabla f(x) + \sum_{i=1}^{m_{\langle l \rangle}} \pi_i \nabla g_i(x) + \sum_{i=1}^{m_{\langle E \rangle}} \rho_i \nabla h_i(x) - \sum_{j=1}^n s_{\langle l \rangle j} e_j + \sum_{j=1}^n s_{\langle u \rangle j} e_j. \end{aligned} \quad (8)$$

Natomiast w problemach (5) i (6) przyjmie ona postać

$$\begin{aligned} \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) &= \\ &= \Phi(x, y, v, w; \mu) + \pi^T (g(x) + y) + \rho^T h(x) + \\ &\quad + s_{\langle l \rangle}^T (-x + v + l) + s_{\langle u \rangle}^T (x + w - u) = \\ &= f(x) - \mu \sum_{i=1}^{m_{\langle l \rangle}} \ln(y_i) - \mu \sum_{j=1}^n \ln(v_j) - \mu \sum_{j=1}^n \ln(w_j) + \\ &\quad + \pi^T (g(x) + y) + \rho^T h(x) + s_{\langle l \rangle}^T (-x + v + l) + s_{\langle u \rangle}^T (x + w - u). \end{aligned} \quad (9)$$

Jej gradient jest następujący:

$$\nabla \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) = \begin{bmatrix} \nabla_x \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) \\ \nabla_y \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) \\ \nabla_v \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) \\ \nabla_w \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) \end{bmatrix}, \quad (9)$$

gdzie:

$$\begin{aligned}\nabla_x \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) &= \nabla f(x) + J_{\langle g \rangle}^T(x)\pi + J_{\langle h \rangle}^T(x)\rho - s_{\langle l \rangle} + s_{\langle u \rangle}, \\ \nabla_y \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) &= -\mu Y^{-1}e + \pi, \\ \nabla_v \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) &= -\mu V^{-1}e + s_{\langle l \rangle}, \\ \nabla_w \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) &= -\mu W^{-1}e + s_{\langle u \rangle}.\end{aligned}$$

Zgodnie z warunkami optymalności pierwszego rzędu w punkcie

$$(x, y, v, w) \in X \times Y \times V \times W \subseteq R^{3n+m_{\langle l \rangle}}$$

mamy

$$\nabla \tilde{L}(x, y, v, w; \pi, \rho, s_{\langle l \rangle}, s_{\langle u \rangle}; \mu) = 0. \quad (11)$$

Z (3)–(11) wynika następujący układ prymalno-dualny [Runka 2009, s. 218]:

$$\begin{bmatrix} g(x) & +y \\ h(x) \\ -x & & +v \\ x & & & +w \\ J_{\langle g \rangle}^T(x)\pi + J_{\langle h \rangle}^T(x)\rho & & & -s_{\langle l \rangle} + s_{\langle u \rangle} \\ & Y\Pi e \\ & & VS_{\langle l \rangle}e \\ & & & WS_{\langle u \rangle}e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -l \\ u \\ -\nabla_x f(x) \\ \mu e \\ \mu e \\ \mu e \end{bmatrix}, \quad (12)$$

gdzie:

$$\begin{aligned}Y &= \text{diag}(y), \quad V = \text{diag}(v), \quad W = \text{diag}(w), \quad \Pi = \text{diag}(\pi), \\ S_{\langle l \rangle} &= \text{diag}(s_{\langle l \rangle}), \quad S_{\langle u \rangle} = \text{diag}(s_{\langle u \rangle}).\end{aligned}$$

Ze względu na ograniczenia nieliniowe w (12) przekształca go się w następujący układ Newtona (aproxymacja liniowa):

$$F\left(\xi^{(k)} + \Delta\xi\right) \approx F\left(\xi^{(k)}\right) + J_{\langle F \rangle}\left(\xi^{(k)}\right)\Delta\xi, \quad (13)$$

gdzie:
 $\xi^{(k)} = \begin{bmatrix} x^{(k)} & y^{(k)} & v^{(k)} & w^{(k)} & \pi^{(k)} & \rho^{(k)} & s_{\langle l \rangle}^{(k)} & s_{\langle u \rangle}^{(k)} \end{bmatrix}^T$ (rozwiązanie w k -tej iteracji),

$\Delta\xi = \begin{bmatrix} \Delta x & \Delta y & \Delta v & \Delta w & \Delta\pi & \Delta\rho & \Delta s_{\langle l \rangle} & \Delta s_{\langle u \rangle} \end{bmatrix}^T$ (kierunek zmiany rozwiązania),

$F\left(\xi^{(k)}\right)$ – funkcja wektorowa,

$J_{\langle F \rangle}\left(\xi^{(k)}\right)$ – macierz Jacobianu funkcji wektorowej $F\left(\xi^{(k)}\right)$.

Zmianę rozwiązania otrzymujemy, rozwiązując iteracyjnie układ równań liniowych

$$J_{\langle F \rangle}\left(\xi^{(k)}\right)\Delta\xi = -F\left(\xi^{(k)}\right), \quad (14)$$

gdzie:

$$F\left(\xi^{(k)}\right) = \begin{bmatrix} g\left(x^{(k)}\right) + y^{(k)} \\ h\left(x^{(k)}\right) \\ l - x^{(k)} + v^{(k)} \\ -u + x^{(k)} + w^{(k)} \\ \nabla_x f(x) + J_{\langle g \rangle}^T\left(x^{(k)}\right)\pi^{(k)} + J_{\langle h \rangle}^T\left(x^{(k)}\right)\rho^{(k)} - s_{\langle l \rangle}^{(k)} + s_{\langle u \rangle}^{(k)} \\ -\mu e + Y_{(k)}\pi^{(k)} \\ -\mu e + V_{(k)}s_{\langle l \rangle}^{(k)} \\ -\mu e + W_{(k)}s_{\langle u \rangle}^{(k)} \end{bmatrix}, \quad (15)$$

$$\begin{aligned}
J_{\langle F \rangle}(\xi^{(k)}) &= \\
&= \begin{bmatrix} J_{\langle g \rangle}(x^{(k)}) & I & & & & & & & \\ J_{\langle h \rangle}(x^{(k)}) & & & & & & & & \\ -I & & I & & & & & & \\ I & & & I & & & & & \\ J_{\langle g \rangle}^T(x^{(k)}) & J_{\langle h \rangle}^T(x^{(k)}) & \nabla_{xx}^2 \tilde{L} & & & & -I & I \\ Y_{(k)} & & & \Pi_{(k)} & & & & & \\ & & & & S_{\langle l \rangle(k)} & & V_{(k)} & & \\ & & & & & S_{\langle u \rangle(k)} & & W_{(k)} & \end{bmatrix}, \quad (16)
\end{aligned}$$

$$J_{\langle g \rangle}(x^{(k)}) = \begin{bmatrix} \nabla g_1(x^{(k)})^T \\ \vdots \\ \nabla g_{m_1}(x^{(k)})^T \end{bmatrix}, \quad J_{\langle h \rangle}(x^{(k)}) = \begin{bmatrix} \nabla h_1(x^{(k)})^T \\ \vdots \\ \nabla h_{m_2}(x^{(k)})^T \end{bmatrix},$$

$$\begin{aligned}
\nabla_{xx}^2 \tilde{L} &\equiv \nabla_{xx}^2 \tilde{L}(x^{(k)}, y^{(k)}, \nu^{(k)}, w^{(k)}; \pi^{(k)}, \rho^{(k)}, s_{\langle l \rangle}^{(k)}, s_{\langle u \rangle}^{(k)}; \mu_k) = \\
&= \nabla_{xx}^2 f(x^{(k)}) + \sum_{i=1}^{m_{\langle l \rangle}} \pi_i \nabla_{xx}^2 g(x^{(k)}) + \sum_{i=1}^{m_{\langle E \rangle}} \rho_i \nabla_{xx}^2 h(x^{(k)}),
\end{aligned}$$

$$Y_{(k)} = \text{diag}(y^{(k)}), \Pi_{(k)} = \text{diag}(\pi^{(k)}), S_{\langle l \rangle(k)} = \text{diag}(s_{\langle l \rangle}^{(k)}), S_{\langle u \rangle(k)} = \text{diag}(s_{\langle u \rangle}^{(k)}).$$

Dla uproszczenia notacji przyjmujemy, że:

$$\begin{aligned}
J_{\langle g \rangle} &\equiv J_{\langle g \rangle}(x^{(k)}), \quad J_{\langle h \rangle} \equiv J_{\langle h \rangle}(x^{(k)}), \\
\nabla_x \tilde{L} &\equiv \nabla_x \tilde{L}(x^{(k)}, y^{(k)}, \nu^{(k)}, w^{(k)}; \pi^{(k)}, \rho^{(k)}, s_{\langle l \rangle}^{(k)}, s_{\langle u \rangle}^{(k)}; \mu_k)
\end{aligned}$$

oraz $\nabla_{xx}^2 \tilde{L}$ jak wyżej.

Korzystając z (13)–(16), rozwiązujemy iteracyjnie układ

$$\begin{bmatrix}
 & J_{\langle g \rangle} & I & & & & & & \\
 & J_{\langle h \rangle} & & & & & & & \\
 & -I & & I & & & & & \\
 & I & & & I & & & & \\
 J_{\langle g \rangle}^T & J_{\langle h \rangle}^T & \nabla_{xx}^2 \tilde{L} & & & & -I & I & \\
 Y_{(k)} & & \Pi_{(k)} & & & & & & \\
 & & & S_{\langle l \rangle(k)} & & V_{(k)} & & & \\
 & & & & S_{\langle u \rangle(k)} & & W_{(k)} & &
 \end{bmatrix}
 \begin{bmatrix}
 \Delta\pi \\
 \Delta\rho \\
 \Delta x \\
 \Delta y \\
 \Delta v \\
 \Delta w \\
 \Delta s_{\langle l \rangle} \\
 \Delta s_{\langle u \rangle}
 \end{bmatrix} =
 \begin{bmatrix}
 g(x^{(k)}) + y^{(k)} \\
 h(x^{(k)}) \\
 l - x^{(k)} + v^{(k)} \\
 -u + x^{(k)} + w^{(k)} \\
 \nabla_x \tilde{L} \\
 Y_{(k)} \pi^{(k)} - \mu_k e \\
 V_{(k)} s_{\langle l \rangle}^{(k)} - \mu_k e \\
 W_{(k)} s_{\langle u \rangle}^{(k)} - \mu_k e
 \end{bmatrix}. \quad (17)$$

Układ (17) jest już układem ograniczeń liniowych. Kolejne rozwiązania otrzymujemy, wyznaczając

$$\xi^{(k+1)} = \xi^{(k)} + \Delta\xi.$$

Praktycznie proces wyznaczania zmiany rozwiązania (12) na podstawie (17) jest bardziej skomplikowany [Nocedal i Wright 2006, s. 564–594; Runka 2011, s. 194–200].

Niezależnie od metody rozwiązywania układu (17), zmiana rozwiązania układu (12) jest następująca:

$$\begin{aligned} & \left[x^{(k+1)} \ y^{(k+1)} \ v^{(k+1)} \ w^{(k+1)} \ \pi^{(k+1)} \ \rho^{(k+1)} \ s_{\langle l \rangle}^{(k+1)} \ s_{\langle u \rangle}^{(k+1)} \right]^T = \\ & = \left[x^{(k)} \ y^{(k)} \ v^{(k)} \ w^{(k)} \ \pi^{(k)} \ \rho^{(k)} \ s_{\langle l \rangle}^{(k)} \ s_{\langle u \rangle}^{(k)} \right]^T + \\ & + \alpha_k \left[\Delta x \ \Delta y \ \Delta v \ \Delta w \ \Delta \pi \ \Delta \rho \ \Delta s_{\langle l \rangle} \ \Delta s_{\langle u \rangle} \right]^T, \end{aligned} \quad (18)$$

gdzie α_k jest długością kroku. Szczegóły związane z wyznaczeniem długości kroku zawierają prace: [Nocedal i Wright 2006, s. 564–594; Runka 2011, s. 194–200].

W przypadku ograniczeń kwadratowych (nierówności) i liniowych (równania) układ prymalno-dualny na podstawie (12) przyjmie następującą postać:

$$\left[\begin{array}{cccccccccccc} \frac{1}{2}x^T Q_1 x + c_1^T x & & & & & & & & & & & \\ & \vdots & & & & & & & & & & \\ & & \frac{1}{2}x^T Q_K x + c_K^T x & & & & & & & & & \\ & & & A_{\langle E \rangle} x & & & & & & & & \\ & & & -x & & & & +v & & & & \\ & & & x & & & & & & +w & & \\ [Q_1 x + c_1 \dots Q_K x + c_K] \pi & -A_{\langle E \rangle}^T \rho & & & & & & & -s_{\langle l \rangle} & +s_{\langle u \rangle} & & \\ & & & & Y \Pi e & & & & & & & \\ & & & & & V S_{\langle l \rangle} e & & & & & & \\ & & & & & & W S_{\langle u \rangle} e & & & & & \end{array} \right] =$$

$$= \left[\begin{array}{c} b_1 - d_1 \\ \vdots \\ b_K - d_K \\ b_{\langle E \rangle} \\ -l \\ u \\ -Q_0 x + c_0 \\ \mu e \\ \mu e \\ \mu e \end{array} \right], \quad (19)$$

w którym (w porównaniu z układem (17)):

$$J_{\langle g \rangle} = \begin{bmatrix} Q_1 x^{(k)} + c_1 \\ \vdots \\ Q_K x^{(k)} + c_K \end{bmatrix}, \quad J_{\langle h \rangle} = A_{\langle E \rangle}, \quad \nabla_{xx}^2 \tilde{L} = Q_0, \quad \nabla_x \tilde{L} = Q_0 x^{(k)} + c_0.$$

Otrzymujemy znacznie prostszy układ do rozwiązywania w kolejnych iteracjach.

W przypadku problemu optymalizacji kwadratowej odpowiednik układu (19) jest przedstawiony w pracy Runki [2006, s. 227].

Przykład

W poniższym przykładzie przedstawiamy translację zapisu matematycznego zadania optymalizacji nieliniowej na zapis w języku modelowania, opracowanego przez SAS Institute Inc., który umożliwia również formułowanie modeli, co wymaga czytania danych z plików zapisanych w formacie wymaganym przez procedury oprogramowania SAS. Tutaj operujemy tylko zadaniami. W celu pokazania postaci zadania z ograniczeniami kwadratowymi i liniowymi, które otrzymujemy poprzez aproksymacje funkcji nieliniowych wyjściowego zadania funkcjami kwadratowymi, wyznacza się punkt startowy za pomocą procedury *model* (*proc model*), której głównym zastosowaniem jest szacowanie modeli ekonometrycznych, ale można również rozwiązywać układy równań nieliniowych. Następnie wykorzystano procedurę IML (*proc IML*) dla wyznaczenia gradientów i hesjanów funkcji nieliniowych, co pozwoliło sformułować zadanie problemu QCQO.

Rozważmy następujące zadanie optymalizacji nieliniowej

$$\min f(x) =$$

$$= 1,5x_1^2 + 0,5x_2^4 + 0,25e^{3x_3} - 6x_1x_2 - 8x_1x_3 - 4x_2x_3 - 58x_1 - 92x_2 - 75x_3$$

przy ograniczeniach:

$$0,25x_1^4 + 0,5x_2^2 + 0,125x_3^2 \leq 408,$$

$$0,125e^{2x_1} + 0,5x_2^4 + 2x_3^2 \leq 15\,100,$$

$$2x_1 + 4x_2 + 5x_3 \leq 96,$$

$$6x_1 + 5x_3 \geq 56,$$

$$3,5 \leq x_1 \leq 20, \quad 4,5 \leq x_2 \leq 16, \quad 2 \leq x_3 \leq 18.$$

W notacji języka modelowania stosowanego przez SAS *proc optmodel* [SAS/OR 2011, s. 23–166], zapis tego zadania jest następujący:

```
proc optmodel;
number l{1..3} = [3.5 4.5 2.0];
number u{1..3} = [20 16 18.0];
var x{j in 1..3} >= l[j] <= u[j];
minimize fobj= 1.5*x[1]^2 + 0.5*x[2]^4 + 0.25 *
exp(3*x[3]) - 6 * x[1] * x[2] - 8*x[1] * x[3] - 4 * x[2]
* x[3] - 58*x[1] - 92*x[2] - 75*x[3];
con c1: 0.25 * x[1]^4 + 0.5 * x[2]^2 + 0.125*x[3]^2 <= 408;
con c2: 0.125 * exp(2*x[1]) + 0.5 * x[2]** 4 + 2 * x[3]**2
<= 15100;
con c3: 2*x[1] + 4*x[2] + 5*x[3] <= 96;
con c4: 6.0 * x[1] + 5 * x[3] >= 56;
... następne polecenia ...
```

Wyznaczono rozwiązania dla przykładowych układów tzw. ograniczeń aktywnych, wykorzystując procedurę SAS *proc model* [SAS/ETS 2011, s. 1079 i n.], **rozwiązując odpowiadające im układy równań nieliniowych.**

Plik zawierający tzw. punkt startowy, wykorzystywany przez *proc model* jest następujący:

```
data G1_2_3_4_QCQO_IN;
input x1 x2 x3;
datalines;
3.61 4.51 2.4
;
run;
```

W poniższym kodzie *proc model* przez SATISFY określa się ograniczenia, które mają być spełnione z równością.

```
proc model data = G1_2_3_4_QCQO_IN;
eq.g1 = 0.25 * x1**4 + 0.5 * x2**2 + 0.125*x3**2 - 408;
eq.g2 = 0.125 * exp(2*x1) + 0.5 * x2** 4 + 2 * x3**2 -
15100;
eq.g3 = 2*x1 + 4*x2 + 5*x3 - 96;
eq.g4 = 6.0 * x1 + 5 * x3 - 56;
eq.g5 = x2 - 4.5;
solve x1 x2 x3 SATISFY=(g1 g2 g3) / solveprint out=G1_2_3_
QCQO_SOLUT;
solve x1 x2 x3 SATISFY=(g2 g3 g4) / solveprint out=G2_3_4_
QCQO_SOLUT;
solve x1 x2 x3 SATISFY=(g1 g2 g4) / solveprint out=G1_2_4_
QCQO_SOLUT;
solve x1 x2 x3 SATISFY=(g3 g4 g5) / solveprint out=G3_4_5_
QCQO_SOLUT;
```

```
solve x1 x2 x3 SATISFY=(g2 g3 g5) / solveprint out=G2_3_5_
QCQO_SOLUT;
solve x1 x2 x3 SATISFY=(g2 g4 g5) / solveprint out=G2_4_5_
QCQO_SOLUT;
run;
```

Wybrano punkt, w którym **drugie i trzecie ograniczenia** są spełnione z równością oraz ograniczenie dolne jest postaci $x_2 = 4,5$. Fragment raportu z *proc model* jest następujący:

Procedura MODEL

Simultaneous Simulation

Obserwacja1Iteracje7CC0.000000eq.g2-0.000000

Solution Values		
x1	x2	x3
5.83215	4.50000	13.26714
Residual Values		
x1	x2	x3
-2.22215	0.01000	-10.86714

Procedura MODEL

Simultaneous Simulation

Opcje zbioru
DATA=G1_2_3_4_QCQO_IN
OUT= G2_3_5_QCQO_SOLUT

Podsumowanie rozwiązania	
Variables Solved	3
Implicit Equations	3
Solution Method	NEWTON
CONVERGE=	1E-8
Maximum CC	7.28E-12
Maksymalna liczba iteracji	7
Total Iterations	7
Average Iterations	7

Przetworzone obserwacje	
Read	1
Solved	1

Variables Solved For	x1 x2 x3
Equations Solved	g2 g3 g5

Wyznaczone rozwiązanie: $x_1 = 5,83215$, $x_2 = 4,5$, $x_3 = 13,26714$. Ten punkt wybrano na zasadzie „zdrowego rozsądku”, który może być względnie blisko punktu optymalnego.

Parametry dla zadania optymalizacji kwadratowej z ograniczeniami kwadratowymi wyliczono za pomocą procedury SAS *proc iml* [SAS/IML 2009], następująco:

```

title 'IML - g2 g3 g5';
proc IML;
use work.G2_3_5_QCQO_SOLUT;
read point 1 var {x1 x2 x3} into x;
constr_arr = (0.25 * x[1]**4 + 0.5 * x[2]**2 +
0.125*x[3]**2 - 408) // (0.125 * exp(2*x[1]) + 0.5 *
x[2]** 4 + 2 * x[3]**2 - 15100) //
(2*x[1] + 4*x[2] + 5*x[3] - 96) // (6.0 * x[1] + 5 *
x[3] - 56) // (x[2] - 4.5);
print , constr_arr;
grad_f = (3 * x[1] - 6*x[2] - 8* x[3] - 58) // (2 *
x[2]**3 - 6*x[1] - 4*x[3] - 92) // (0.75 * exp(3*x[3]) -
8*x[1] - 4*x[2] - 75);
* g3, g4, g5 ;
JC = (0.5*exp(2*x[1]) || (2*x[2]**3) || (4*x[3])) // ((-
2) || (-4) || (-5)) // ((0) || (-1) || (0));
grad_g1 = (1*x[1]**3) // (1*x[2]) // (0.5*x[3]) ;
grad_g2 = (0.5*exp(2*x[1]) ) // (2*x[2]**3) // (4*x[3]);
grad_g3 = (-2) // (-4) // (-5);
grad_g4 = (6) // (0) // (5);
grad_g5 = (0) // (-1) // (0);
Lmult = -inv(JC`) * grad_f;
print , "g2 & g3 & g5";
print , x;
print , lmult;
print , "Trans of Jacobian Matrix";
print , (JC`);
print , grad_g1;
print , grad_g2;
print , grad_g3;
print , grad_g4;
print , grad_g5;
print , grad_f;
print , x;
fobj= 1.5*x[1]**2 + 0.5*x[2]**4 + 0.25 * exp(3*x[3]) - 6 *
x[1] * x[2] - 8* x[1] * x[3] - 4 * x[2] * x[3] - 58*x[1] -
92*x[2] - 75*x[3];

```

```

print fobj;
Mult_nams={ "Lmult2" "Lmult3" "Lmult5" };
create G2_3_5_QCQO_Lmult from Lmult [colname = Mult_nams];
append from Lmult;
close G2_3_5_QCQO_Lmult;
close work.G2_3_5_QCQO_SOLUT;
Hessian_f= ((3) || (-6) || (-8)) // ((-6) || 6*x[2]**2 ||
(-4)) // ((-8) || (-4) || 2.25*exp(3*x[3])) );
Hessian_g1 = (3*x[1] || (0) || (0)) // ((0) || (1) || (0))
// ((0) || (0) || (0.5));
Hessian_g2 = (1.0*exp(2*x[1]) || (0) || (0)) // ((0) || 6*
x[2]**2 || (0)) // ((0) || (0) || (4));
print ,Hessian_f;
print ,Hessian_g1;
print ,Hessian_g2;
call eigen(Lambda, P, Hessian_f);
print ,Lambda;
print, P;
quit;

```

Za pomocą tej procedury wyznaczono gradienty i hesjany dla funkcji nieliniowych oraz sprawdzono dopuszczalność rozwiązania (contr_arr), poniżej:

contr_arr
-86.63507
-7.28E-12
0
45.328604
0

Ograniczenia: drugie, trzecie i piąte są spełnione „prawie” z równością, a pierwsze i czwarte ze ścisłą nierównością. Gradienty są następujące (oznaczone grad_g1 – grad_g5):

grad_g1	grad_g2	grad_g3	grad_g4	grad_g5
198.3747	58171.739	-2	6	0
4.5	182.25	-4	0	-1
6.6335698	53.068558	-5	5	0

Gradient funkcji celu jest

grad_f
-173.6407
2.1885354
1.4474E17

Hesjany funkcji nieliniowych:

Hessian_f			Hessian_g1			Hessian_g2		
3	-6	-8	17.496453	0	0	116343.48	0	0
-6	121.5	-4	0	1	0	0	121.5	0
-8	-44.3423E17		0	0	0.5	0	0	4

Na podstawie powyższych danych rozwiązujemy zadanie z ograniczeniami kwadratowymi, stosując procedurę *optmodel* (*proc optmodel*).

```

title 'QCQO Problem';
proc optmodel;
number l{1..3} = [3.5 4.5 2.0];
number u{1..3} = [20 16 18.0];
var x{j in 1..3} >= l[j] <= u[j];
minimize fobj= 3*x[1]^2 + 121.5*x[2]^2 + 4.3423e1 *x[3]^2
- 12 * x[1] * x[2] - 16 * x[1] * x[3] - 8 * x[2] * x[3] -
173.6407*x[1] + 2.1885354*x[2] + 1.4474e17*x[3];
con c1: 17.496453* x[1]^2 + 1.0 * x[2]^2 + 0.5*x[3]^2 <=
408;
con c2: 116.343 *x[1]^2 + 121.5 * x[2]^2 + 4 * x[3]^2 <=
15100;
con c3: 2*x[1] + 4*x[2] + 5*x[3] <= 96;
con c4: 6.0 * x[1] + 5 * x[3] >= 56;
/* Starting point x1 = 7, x2 = 5, x3 = 7 */
x[1] = 6.7; x[2] = 4; x[3] = 1.5;
SOLVE obj fobj with ipnlp / maxiter=2000;
print x 18.4;
print c1.dual 18.4 c2.dual 18.4 c3.dual 18.4 c4.dual 18.4;
print x.dual 18.4;
print fobj 18.4;
quit;

```

Otrzymano rozwiązanie (fragment raportu z *proc optmodel*):

QCQO Problem

The OPTMODEL Procedure

Podsumowanie rozwiązania	
Solver	IPNLP/IPQN
Objective Function	fobj
Solution Status	Optimal
Objective Value	8.2055437E17
Iterations	138
Infeasibility	8.5345601E-9
Optimality Error	9.0532093E-7

[1]	x
1	4.6090
2	4.5000
3	5.6692

c1.DUAL	c2.DUAL	c3.DUAL	c4.DUAL
-1124336466717570	-11409.9256	-0.003830222808885	300692

[1]	x.DUAL
1	3197266.3754
2	10119028212925762
3	53801166.6453

fobj
820554367232925184

Otrzymano wartości zmiennych: $x_1 = 4,6090$, $x_2 = 4,5$, $x_3 = 5,6692$. Należy zauważyć, że określając punkt startowy w powyższym kodzie, przyjęto $x_3 = 1,5 < 2$, które jest dolnym ograniczeniem dla tej zmiennej. Stosując algorytm punktów wewnętrznych, można startować od dowolnego niedopuszczalnego rozwiązania. Nie jest jednak zalecane, ażeby początkowe wartości zmiennych „zbyt daleko” odbiegały od wartości dopuszczalnych, bo może to „istotnie” wydłużyć proces obliczeniowy. Tutaj rozwiązując zadanie z ograniczeniami kwadratowymi, zastosowano w *proc model* algorytm punktów wewnętrznych optymalizacji nieliniowej.

W przedstawionym przykładzie zaprezentowano wykonanie tylko początkowych kroków w celu pokazania przejścia od zadania problemu NLO do zadania problemu QCQO, wykorzystując procedury oprogramowania SAS Institute Inc.

Podstawowym utrudnieniem w stosowaniu aproksymacji kwadratowych dla funkcji nieliniowych jest wyznaczanie hesjanów i gradientów. Można je wyznaczać numerycznie [Nocedal i Wright 2006, s. 193–217] lub analitycznie, co wymaga napisania odpowiedniego oprogramowania, w którym wyznaczy się formuły analityczne do wyliczenia wartości elementów wektora gradientu i macierzy hesjanu.

Zamiast rozwiązywania zadań optymalizacji nieliniowej z ograniczeniami kwadratowymi i liniowymi można dokonać jego transformacji do zadania problemu SOC (*second-order cone*). W języku angielskim często używa się SOCP problem (*second-order cone program*). Tutaj będziemy stosować SOCO (*second-order cone optimization*) problem.

2. Optymalizacja na stożku

Problem SOCO (second-order cone optimization) jest zdefiniowany następująco (por. [Lobo i in. 1998, s. 194]):

$$\min f^T x$$

przy ograniczeniach

$$\begin{aligned} \|A_i x + b_i\| &\leq c_i^T x + d_i, \quad i = 1, \dots, K, \\ Fx &= g, \\ l &\leq x \leq u, \end{aligned} \tag{20}$$

gdzie:

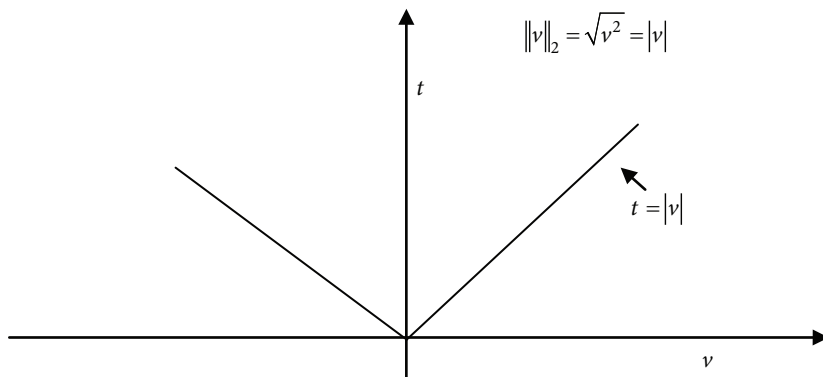
$$\begin{aligned} f &\in \mathbb{R}^n, \quad A_i \in \mathbb{R}^{(n_i-1) \times n}, \quad b_i \in \mathbb{R}^{n_i-1}, \quad c_i \in \mathbb{R}^n, \quad d_i \in \mathbb{R}, \quad F \in \mathbb{R}^{n_{(E)} \times n}, \quad g \in \mathbb{R}^{n_{(E)}}, \\ l, u &\in \mathbb{R}^n, \quad u > l, \quad x \in \mathbb{R}^n, \quad \|\cdot\| \equiv \|\cdot\|_2, \end{aligned}$$

przy czym $\|A_i x + b_i\| \leq c_i^T x + d_i$ jest ograniczeniem na stożku drugiego rzędu wymiaru n_i .

Jednostkowy stożek drugiego rzędu o wymiarze k zdefiniowany jest jako

$$\left\{ \begin{bmatrix} v \\ t \end{bmatrix} : v \in \mathbb{R}^{k-1}, t \in \mathbb{R}, \|v\| \leq t \right\}$$

i nazywany jest [Lobo i in. 1998, s. 104] stożkiem: kwadratowym, *ice-cream* lub Lorentza. Jego prezentację w przypadku $k = 2$ zawiera poniższy rysunek.



Stożek Lorentza dla $k = 2$. Wyznaczamy $t \geq \|v\|_2$

Bazaraa, Sherali i Shetty [2006, s. 62 i n.] przedstawili inne typy stożków.

W przypadku problemu optymalizacji kwadratowej (QO) [Lobo i in. 1998, s. 197] postaci

$$\min f^T Qx + 2c^T x + d$$

przy ograniczeniach

$$\alpha_i^T x \leq b_i, \quad i = 1, \dots, p, \quad (21)$$

$$l_j \leq x_j \leq u_j, \quad j = 1, \dots, n,$$

gdzie $Q \succ 0$, $Q^T = Q$,

chcąc przekształcić go w problem SOCO, wprowadza się podstawienie

$$v = Q^{\frac{1}{2}} x + Q^{-\frac{1}{2}} c. \quad (22)$$

w którym:

- $Q^{\frac{1}{2}} = V \times \text{diag} \left(\lambda_1^{\frac{1}{2}}, \dots, \lambda_n^{\frac{1}{2}} \right) \times V^T$,

gdzie $V, \lambda : Q = V \times \text{diag}(\lambda_1, \dots, \lambda_n) \times V^T$, zatem V jest macierzą wektorów własnych, a λ_i ($i = 1, \dots, n$) są wartościami własnymi macierzy Q ,

- $Q^{-\frac{1}{2}} = V \times \text{diag} \left(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_n^{-\frac{1}{2}} \right) \times V^T$,

gdzie $V, \lambda : Q^{-1} = V \times \text{diag}(\lambda_1^{-1}, \dots, \lambda_n^{-1}) \times V^T$ i, podobnie jak wyżej, V jest macierzą wektorów własnych, a λ_i ($i = 1, \dots, n$) są wartościami własnymi macierzy Q^{-1} .

Mamy wówczas:

$$v^T v = x^T Q^{\frac{1}{2}} Q^{\frac{1}{2}} x + 2x^T Q^{\frac{1}{2}} Q^{-\frac{1}{2}} c + c^T Q^{-\frac{1}{2}} Q^{\frac{1}{2}} x = x^T Q x + 2x^T c + c^T Q^{-1} c. \quad (23)$$

Funkcja $f(x) = x^T Q x + 2c^T x + d$ w (21) ma rozwiązanie optymalne bez ograniczeń $\bar{x}_{\text{opt}} = -Q^{-1}c$, wyznaczone z warunków optymalności pierwszego rzędu:

$$\nabla_x f(x) = 2Qx + 2c = 0 \Leftrightarrow Qx + c = 0,$$

a jej wartość optymalna

$$\begin{aligned} f(\bar{x}_{\text{opt}}) &= (-1)c^T Q^{-1} Q (-1) Q^{-1} c + 2(-1)c^T Q^{-1} c + d = \\ &= c^T Q^{-1} c - 2c^T Q^{-1} c + d = -c^T Q^{-1} c + d. \end{aligned}$$

Zatem, pomijając powyżej stałą d , zachodzi: $v^T v = x^T Q x + 2x^T c - f(\bar{x}_{\text{opt}})$. Ponieważ $f(\bar{x}_{\text{opt}}) \leq f(x)$, $x : Ax \leq b$, $l \leq x \leq u$ w (21), zatem $v^T v \geq 0$.

Przyjmując w (20) $c_i^T = 0, i = 1, \dots, K$, otrzymujemy problem SOCO postaci:

$$\begin{aligned} & \min t \\ & \text{przy ograniczeniach} \\ & \left\| Q^2 x + Q^{-2} c \right\|_2 \leq t, \\ & Ax \leq b, \\ & l \leq x \leq u, \end{aligned}$$

lub w ogólniejszym zapisie, korzystając z (21) i (22):

$$\begin{aligned} & \min t \\ & \text{przy ograniczeniach} \\ & v = Q^2 x + Q^{-2} c, \\ & Ax \leq b, \\ & l \leq x \leq u, \\ & (t; v) \succeq_{\mathbb{C}} 0, \end{aligned}$$

gdzie $(t; v) \succeq_{\mathbb{C}} 0 \Leftrightarrow \|v\|_2 \leq t$ i t wyraża minimalną odległość, wyrażoną za pomocą normy, pomiędzy rozwiązaniem względem zmiennych x a rozwiązaniem optymalnym bez ograniczeń $\bar{x}_{\text{opt}} = -Q^{-1}c$.

Korzystając z przekształcenia problemu QO w problem SOCO, ogólniejszy problem QCQO postaci

$$\begin{aligned} & \min x^T Q_0 x + 2x^T c_0 + d_0 \\ & \text{przy ograniczeniach:} \\ & x^T Q_i x + x^T c_i + d_i \leq b_i, i \in \langle I \rangle \text{ lub } x^T Q_i x + x^T c_i + d_i - b_i \leq 0, i \in \langle I \rangle, \\ & \alpha_i^T x = b_i, i \in \langle E \rangle, \\ & l_j \leq x_j \leq u_j, j = 1, \dots, n, \end{aligned} \quad (24)$$

można (po podstawieniu $\tilde{d}_i = d_i - b_i$) przetransformować w SOCO:

$$\begin{aligned} & \min t \\ & \text{przy ograniczeniach} \\ & \left\| Q_0^2 x + Q_0^{-2} c_0 \right\|_2 \leq t, \\ & \left\| Q_i^2 x + Q_i^{-2} c_i \right\|_2 \leq \left(c_i^T Q_i^{-1} c_i - \tilde{d}_i \right)^{\frac{1}{2}}, i \in \langle I \rangle, \\ & \alpha_i^T x = b_i, i \in \langle E \rangle, \end{aligned} \quad (25)$$

Zgodnie z (23), przekształcając w (25)

$$\left\| Q_i^{\frac{1}{2}}x + Q_i^{-\frac{1}{2}}c_i \right\|_2 \leq \left(c_i^T Q_i^{-1} c_i - \tilde{d}_i \right)^{\frac{1}{2}}$$

w

$$\left\| Q_i^{\frac{1}{2}}x + Q_i^{-\frac{1}{2}}c_i \right\|_2^2 \leq c_i^T Q_i^{-1} c_i - \tilde{d}_i,$$

wrażenie $c^T Q^{-1} c$ występuje po lewej i prawej stronie (co oznacza, że może być wyeliminowane), więc otrzymamy ograniczenia zgodne z (2) postaci $x^T Q_i x + 2x^T c_i + \tilde{d}_i \leq 0$ ($x^T Q_i x + 2x^T c_i + d_i \leq b_i$).

W przypadku funkcji kwadratowej

$$\frac{1}{2}x^T Qx + x^T c + d,$$

stosujemy podstawienie

$$v = \sqrt{\frac{1}{2}}Q^{\frac{1}{2}}x + \sqrt{\frac{1}{2}}Q^{-\frac{1}{2}}c.$$

Wówczas

$$\begin{aligned} v^T v &= \sqrt{\frac{1}{2}}x^T Q^{\frac{1}{2}}Q^{\frac{1}{2}}x \sqrt{\frac{1}{2}} + 2\sqrt{\frac{1}{2}}x^T Q^{\frac{1}{2}}Q^{-\frac{1}{2}}c \sqrt{\frac{1}{2}} + \sqrt{\frac{1}{2}}c^T Q^{-\frac{1}{2}}Q^{-\frac{1}{2}}c \sqrt{\frac{1}{2}} = \\ &= \frac{1}{2}x^T Qx + x^T c + \frac{1}{2}c^T Q^{-1}c. \end{aligned}$$

Teraz (pominąwszy stałą d) zachodzi

$$\begin{aligned} f(\bar{x}_{\text{opt}}) &= (-1)\frac{1}{2}c^T Q^{-1}Q(-1)Q^{-1}c + (-1)c^T Q^{-1}c = \\ &= \frac{1}{2}c^T Q^{-1}c - c^T Q^{-1}c = -\frac{1}{2}c^T Q^{-1}c, \end{aligned}$$

skąd

$$v^T v = \frac{1}{2}x^T Qx + x^T c - f(\bar{x}_{\text{opt}}) \geq 0.$$

Podobnie jak przy rozwiązywaniu instancji problemu QCQO, również rozwiązując instancje SOCO oprogramowanie nie jest powszechnie dostępne. Głównymi producentami tego typu oprogramowania są przykładowo: IBM CPLEX, MatLab, Mosek, OpenOpt.

Innym problemem wywodzącym się z ekonometrii jest problem estymacji ℓ_1 parametrów nieliniowych modeli. Stosując aproksymacje kwadratowe nieliniowych funkcji [Runka 2011, s. 148 i n.]:

$$\min e^T r^+ + e^T r^-$$

przy ograniczeniach:

$$\begin{aligned} \nabla_{\beta} \Phi(x_i^T; \beta_k)^T \Delta\beta + \Delta\beta^T \nabla_{\beta\beta}^2 \Phi(x_i^T; \beta_k) \Delta\beta + r_i^+ - r_i^- &= \tilde{y}_i, \\ r_i^+, r_i^- &\geq 0, \quad i = 1, \dots, n, \end{aligned} \quad (26)$$

gdzie:

$$\begin{aligned} r^+, r^- &\in \mathbb{R}_+^n, \quad \Delta\beta \in \mathbb{R}^{m+1}, \quad \nabla_{\beta} \Phi(x_i^T; \beta_k) \in \mathbb{R}^{m+1}, \\ \nabla_{\beta\beta}^2 \Phi(x_i^T; \beta_k) &\in \mathbb{R}^{(m+1) \times (m+1)}, \end{aligned}$$

możemy przekształcić go w problem SOCO

$$\min \sum_{i=1}^n t_i$$

przy ograniczeniach:

$$\sqrt{\left(\nabla_{\beta} \Phi(x_i^T; \beta_k) \Delta\beta + \frac{1}{2} \Delta\beta^T \nabla_{\beta\beta}^2 \Phi(x_i^T; \beta_k) \Delta\beta - \tilde{y}_i \right)^2} \leq t_i, \quad i = 1, \dots, n, \quad (27)$$

$$t_i \geq 0, \quad i = 1, \dots, n,$$

gdzie:

$$\tilde{y}_i = y_i - \Phi(x_i^T; \beta_k), \quad \tilde{y} = y - \Phi(X; \beta_k), \quad \tilde{y} \in \mathbb{R}^n.$$

W (27) ograniczenia są stożkami jednostkowymi, jak na rys. 1, zgodnie z którymi dla skalnych wartości $v_i = \nabla_{\beta} \Phi(x_i^T; \beta_k) \Delta\beta + \frac{1}{2} \Delta\beta^T \nabla_{\beta\beta}^2 \Phi(x_i^T; \beta_k) \Delta\beta - \tilde{y}_i$, zachodzi $\|v_i\|_2 = |v_i|$.

Zatem stosując estymację ℓ_1 parametrów nieliniowych modeli, możemy rozwiązywać instancje problemu (27), zamiast instancji problemu (26).

Stosując transformacje do problemu SOCO, napotykamy takie same trudnienia jak w problemie QCQO, a mianowicie – jak wyznaczać gradient i hesjan.

Podsumowanie

Dla rozwiązywania instancji (zadań) problemu optymalizacji nieliniowej, wykorzystuje się zarówno solwery zawierające implementacje wielu algorytmów stosowanych od dziesiątek lat, jak i nowsze, jak na przykład algorytmu punktów wewnętrznych rozwiązywania zadań optymalizacji kwadratowej i nieliniowej. Poszukiwanie alternatywnych koncepcji algorytmów oraz specyfika zagadnień praktycznych spowodowały pojawienie się problemów QCQO i SOCO. Innym podejściem jest transformacja problemu QCQO w problem SDP (Semidefinite Programming) [Anstreicher 2004], do którego można również transformować problem QCQO.

Bibliografia

- Anstreicher, K.M., 2004, *SDP versus RLT for Nonconvex QCQP*, Workshop on Integer Programming and Continuous Optimization, Chemnitz.
- Bazaraa, M.S., Sherali, H.D., Shetty, C.M., 2006, *Nonlinear Programming. Theory and Algorithms*, J. Wiley & Sons, 3rd ed.
- Boyd, S., Vandenberghe, L., 2009, *Convex Optimization*, Cambridge University Press.
- Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H., 1998, *Applications of Second-Order Cone Programming*, Linear Algebra and its Applications.
- Nocedal, J., Wright, S.J., 2006, *Numerical Optimization*, Springer (Series in Operations Research).
- Runka, H.J., 2006, *Ograniczenia na zmienne w algorytmach punktów wewnętrznych*, w: Panek, E. (red.), *Matematyka w ekonomii*, Wydawnictwo Akademii Ekonomicznej w Poznaniu, Poznań.
- Runka, H.J., 2009, *Interior Points in Nonlinear Optimization with Bounds on Variables*, w: Panek, E. (red.), *Mathematics in Economics*, red. nauk. E. Panek, Wydawnictwo Uniwersytetu Ekonomicznego w Poznaniu, Poznań.
- Runka, H.J., 2012, *Metody punktów wewnętrznych w estymacji parametrów modeli ekonometrycznych*, w: Panek, E. (red.), *Matematyka i informatyka na usługach ekonomii. Modelowanie zjawisk gospodarczych. Elementy teorii*, Wydawnictwo Uniwersytetu Ekonomicznego w Poznaniu, Poznań.
- SAS/ETS * 9.3, User's Guide, 2011, SAS Publishing.
- SAS/IML * 9.3, User's Guide, 2011, SAS Publishing.
- SAS/OR * 9.3, User's Guide: Mathematical Programming, 2011, SAS Publishing.