Früherkennung von Redundanzen bei quadratischen Optimierungsproblemen

P. Recht

DoMuS-Kurzpräsentation, 08.November 2004

Was erwartet Sie?

1. Motivation

2. Eliminationskriterien

3. Beispiele

4. Schlussbemerkungen

1. Motivation

Betrachte quadratische Optimierungsproblem **QP**:

$$\min_{x \in S} f(x) = \frac{1}{2} \cdot x^t C x + d^t x$$

$$\text{mit} \quad S = \left\{ x \mid a_i^t x \leq b_i, i = 1, 2, ..., m \right\}$$

C symmetrisch, positiv-definit $(n \times n)$ Matrix und $d, a_i \in \mathbf{R^n}$.
$x_0 := -C^{-1}d$.
O.B.d.A. $f = \frac{1}{2} \cdot (x - x_0)^t C (x - x_0)$. Annahme: $x_0 \notin S \neq \emptyset$.

Warum ?

- Sub-Probleme des Typs **QP** tauchen auf im Zusammenhang mit

  - der Lösung von **NLP**-Probleme

$$\min_{x \in M} g(x) \text{ mit } M = \{x \mid h_i(x) \leq 0, i = 1, 2, ..., m\}\,.$$

  $(g, h_i \in C^2)$ mittels **SQP**-Verfahren.

- 90er Jahre: **IP**-Methoden werden State-of-the-art um **QP** zu lösen
  In jedem Iterationsschritt: Lösung eines "grossen" linearen Systems. Dimension wesentlich bestimmt durch n und $\mathbf{m}$.

$\boxed{\textbf{Simple Tatsache: } x^* \text{ bestimmt durch } n_1 \leq n \text{ Restriktionen.}}$

$K \approx \binom{\mathbf{m}}{n_1}$ "$\Rightarrow$" ??? $\Rightarrow$ Verkleinerung von $\mathbf{m}$

"**Früh**erkennung" und Elimination von in der Optimallösung x^* "überflüssigen" Restriktionen ($a_i^t x^* < b_i$)

wesentlich:

- "einfache" Kalkulationen
- unabhängig vom Lösungsverfahren für **QP**

- Lösungsverfahren nicht "stören"

Annahme: Verwendter QP-solver generiert eine (fallende) Folge von oberen Schranken für den Optimalwert $f(x^*)$.

2. Eliminationskriterien

Berechnung struktureller Grössen **QP**:

$$\boxed{\begin{array}{c} C^{-1} \\ r_i := a_i^t x_0 - b_i, \qquad i = 1, 2, ..., m \\ q_{ij} := a_i^t C^{-1} a_j, \qquad i, j = 1, 2, ..., m \\ J := \{i \mid r_i < 0\} \end{array}}$$

Bemerkung: $(\tilde{x}_0, \tilde{C}) \Rightarrow \tilde{f}, (\tilde{r}_i)_{i=1...m}, (\tilde{q}_{ij})_{i=1...m;j=1....n}, \tilde{J}$

Nicht-linearer Trennungssatz liefert vollständige Charakterisierung "überflüssiger" Restriktionen:

Proposition 1 *Für QP sind folgende beiden Aussagen äquivalent:*

i. Die $i-te$ Nebenbedingung ist nicht-aktiv in x^*.

ii. Es gibt ein Tripel $\tilde{x}_o \in \mathbf{R}^n, \tilde{C} \in \mathbf{R}^{n,n}$, positiv-definit und $\tilde{\alpha} > 0$, welches den beiden folgenden Bedingungen genügt

a.) $\tilde{f}(x^*) \le \tilde{\alpha}$

b.) $i \in \tilde{J}$ and $\tilde{r}_i^2 > 2 \cdot \tilde{\alpha} \cdot \tilde{q}_{ii}$.

Leider: Unbefriedigend für algorithmische Anwendung

Daher:

(x_0, C) bleibt (zunächst) unverändert,

$\alpha \mathrel{\widehat{=}}$ beliebige obere Schranke an $f(x^*)$, $\quad f(x^*) \le \alpha$.

Eliminations-Kriterium 1

Korollar 1 *Sofern* $i \in J$ *die Bedingung* $\beta_i := r_i^2 \cdot (2q_{ii})^{-1} < \alpha$ *erfüllt, ist diese Nebenbedingung nicht-aktiv in* x^*.

Eliminations-Kriterium 2

Proposition 2 *Für $i \in 1, 2, ..., m$ sei*

$$S_{i,\alpha} := \{x | a_i^t x = b_i\} \cap \{x | f(x) \leq \alpha\}.$$

*Für $j \in 1, 2, ..., m$, sei $\tau_{j,i}^*(\alpha)$ der Optimal*wert *des Problems*

$$\min_{x \in S_{i,\alpha}} a_j^t x.$$

i. $\tau_{j,i}^*(\alpha) = a_j^t x_o - q_{ii}^{-1} \cdot \left[r_i \cdot q_{ij} + (2 \cdot \alpha \cdot q_{ii} - r_i^2)^{\frac{1}{2}} \cdot (q_{ii} \cdot q_{jj} - q_{ij}^2)^{\frac{1}{2}} \right].$

ii. Falls $\tau_{j,i}^*(\alpha) - b_j > 0$, dann ist die $i-te$ Nebenbedingung nicht-aktiv in x^*.

$\Rightarrow \alpha_{i,j}^* = (2 \cdot q_{ii})^{-1} \cdot [r_i^2 + (r_j \cdot q_{ii}^{\frac{1}{2}} - r_i \cdot q_{ij})^2 \cdot (q_{ii} \cdot q_{jj} - q_{ij}^2)^{-1}]$
Falls $\alpha < \alpha_{i,j}^*$, dann ist die $i-te$ Nebenbedingung nicht-aktiv in x^*.

Algorithmisches Schema

QP-solver $QPSOLV_{(f,S)}$ sei gegeben (**POZESSOR 1)**. Übergibt in Schritt k den Wert α_k an **PROZESSOR 2**.

PROZESSOR 2
Schritt 0:

Berechne $q_{ij}, r_i, J, \alpha^*_{i,j}, \alpha^{**}_{i,j}$ (wenn möglich)	setze $I_1 := \{1, 2, .., m\}, I_2 := I_3 := \emptyset$
Für $i \in I_1$, berechne $\alpha^*_i := \max_j \alpha^*_{i,j}$	
Für $i \in J$, berechne $\beta^*_i := \frac{1}{2} q_{ii}^{-1} r_i^2$	
Falls $\alpha^*_{ij} = \alpha^{**}_{i,j} = b_j$	setze $I_1 := I_1 \setminus \{i\}, J := J \setminus \{i\}$.
	falls $q_{ij} < 0$, setze $I_3 := I_3 \cup \{j\}$

Schritt k:

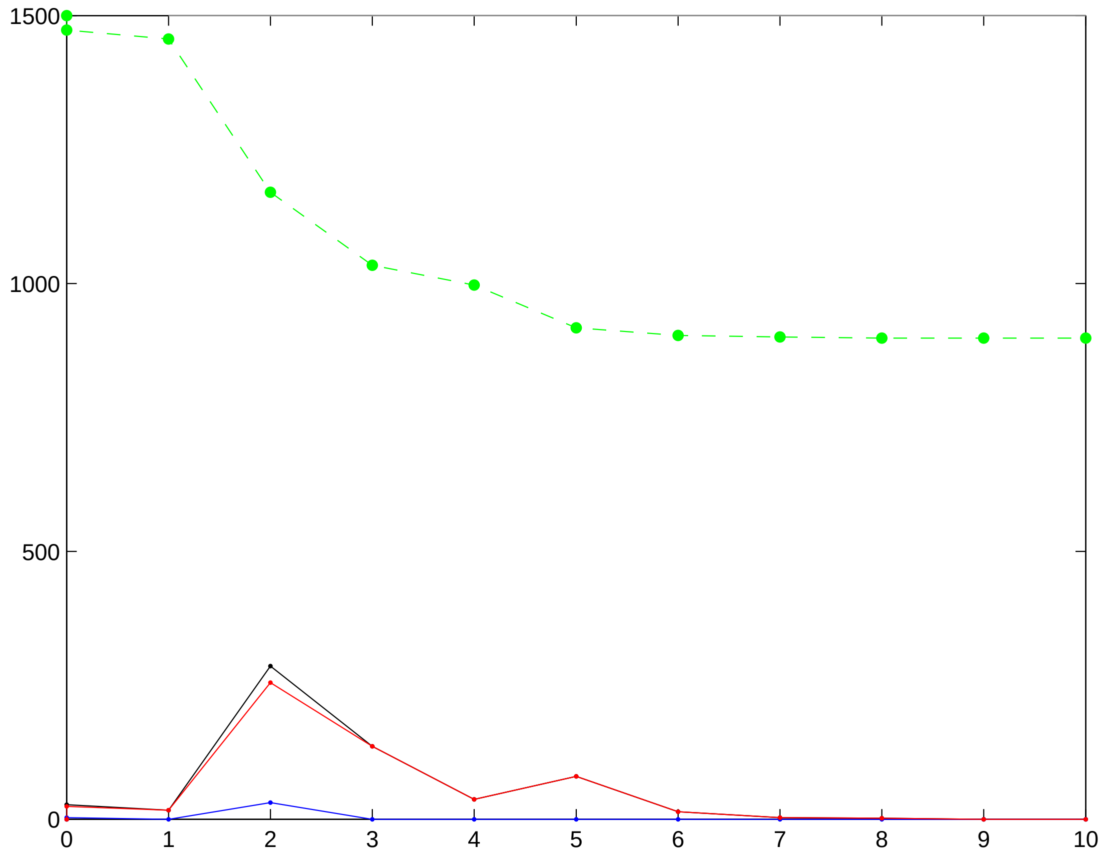
Für $i \in J$:	
falls $\alpha_k < \beta^*_i$,	setze $I_1 := I_1 \setminus \{i\}, J := J \setminus \{i\}$,
Für $i, j \in I_1$:	
Falls $\alpha_k < \alpha^*_i$,	setze $I_1 := I_1 \setminus \{i\}, J := J \setminus \{i\}$.
Falls $\alpha_k < \alpha^{**}_{i,j}$	setze $I_2 := I_2 \cup \{(i, j)\}$.
Setze $S := \{x \mid a_i^t x \leq b_i, i \in I_1\}$	Übergebe S, I_2, I_3 an **PROZESSOR 1**

Beispiele

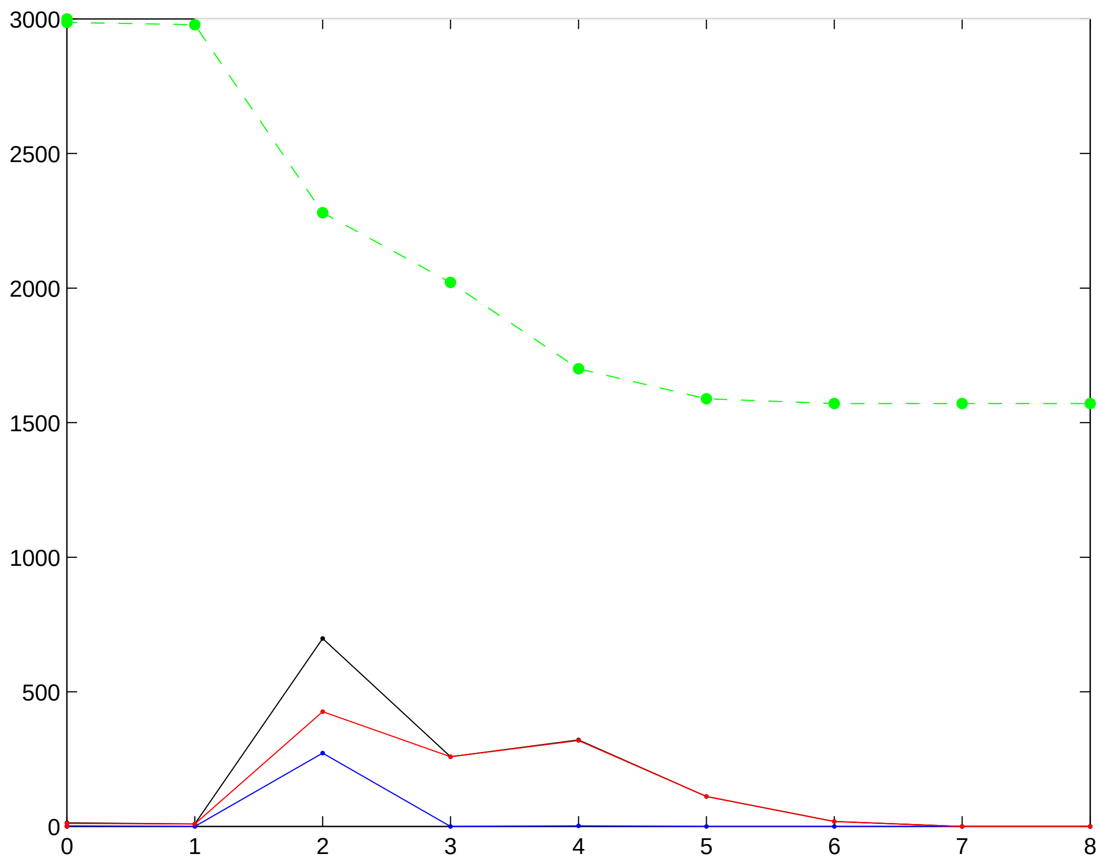
n	m	k	r_s	$r_s^{(1)}$	$r_s^{(2)}$	r_a	α_0	α_{k^*}
2	5	4	3	1	2	2	3.99e+03	1.01e+03
2	12	6	11	3	8	1	2.37e+03	1.46e+02
2	20	7	18	2	16	2	3.59e+02	1.51e+02
2	50	6	49	31	18	1	4.87e+02	1.274e+02
2	100	17	98	57	41	2	4.80e+02	1.340e+02
2	120	6	116	40	76	2	1.18e+03	5.23e+02
2	200	9	196	89	107	2	7.23e+02	1.87e+02
2	600	8	597	333	294	2	1.39e+02	7.18e+01
2	1500	6	1499	1058	441	1	1.26e+03	2.25e+02
2	3000	11	2998	518	2480	2	8.39e+02	6.16e+02

n	m	k	r_s	$r_s^{(1)}$	$r_s^{(2)}$	r_a	α_0	α_{k^*}
20	200	7	142	55	87	10	2.03e+04	2.36e+03
20	400	9	324	65	259	5	1.66e+04	3.62e+03
20	600	10	97	1	96	17	1.49e+04	6.45e+03
20	1000	11	89	0	98	16	2.31e+04	1.13e+04
20	1500	11	602	34	568	14	1.29e+04	4.75e+03
20	3000	11	348	1	347	15	2.34e+04	1.30e+04
20	3200	11	2599	527	2072	9	2.17e+04	5.64e+03
40	200	6	183	125	58	4	4.41e+04	1.52e+03
40	400	8	212	42	170	9	3.27e+04	5.73e+03
40	600	10	166	32	134	16	2.88e+04	5.78e+03
40	1000	9	33	1	32	26	3.99e+04	1.44e+04
40	1500	10	72	0	72	31	3.55e+04	1.19e+04
40	3000	8	1429	276	1153	18	3.80e+04	9.13e+03
40	3200	11	891	24	867	25	3.93e+04	1.24e+04
40	3300	10	441	3	438	29	3.81e+04	1.44e+04

n	m	k	r_s	$r_s^{(1)}$	$r_s^{(2)}$	r_a	α_0	α_{k^*}
100	200	8	199	119	80	1	8.19e+04	2.13e+03
100	390	9	257	105	152	16	7.71e+04	3.11e+03
100	400	12	0	0	0	80	9.95e+04	6.85e+04
100	500	8	150	72	78	20	7.75e+04	5.47e+03
100	580	9	480	340	140	9	7.20e+04	2.38e+03
100	590	9	523	412	111	12	9.21e+04	1.56e+03
100	600	11	0	0	0	92	8.45e+04	6.36e+04
100	690	9	419	279	140	16	8.82e+04	4.11e+03
100	992	8	660	341	319	19	8.76e+04	6.27e+03
100	1000	11	0	0	0	50	8.99e+04	2.05e+04
100	1500	10	36	2	34	35	8.89e+04	1.91e+04
100	2000	12	0	0	0	55	8.33e+04	2.98e+04
100	2003	11	760	151	609	31	1.09e+05	1.31e+04
100	2500	12	0	0	0	82	1.03e+05	6.29e+04
100	2513	11	868	290	578	38	9.04e+04	1.07e+04
100	3000	11	463	98	365	51	8.13e+04	1.19e+04
100	3200	10	1649	743	906	30	9.41e+04	8.19e+03



Eliminationsverlauf 1: $n = 20$, $m = 1500$



Elimination sverlauf 2: $n = 40$, $m = 3000$

Schlussbemerkungen

- "einfache" Bedingungen für einen "Nicht-Aktivitäts-Check "
- unterscheidet sich von klassischer "Active-set Strategie"
- kann "vor" oder "während" der Optimierungsprozedur durchgeführt werden.
- behindert die eigentliche Optimierungsroutine nicht
- ideal für den Einbau in ein "(Multi)- Agenten-System"

Offene Fragen und Problemstellungen:

- weitere Eliminationskriterien
- Welcher Löser ist der best-geeignete (wie schnell fallen die Schranken α)?
- Trade-off zwischen der Berechnungszeit für die q_{ij} und der Beschleunigungszeit des Optimierungsverfahrens
- primal-duale Verfahren zur Früherkennung *aktiver* Restriktionen

- binäre Quadratische Optimierungsprobleme

- Einbau in branch and cut Verfahren für ganzzahlige quadratische Optimierungsprobleme