

特別研究報告書

逐次二次制約二次計画法とその応用

指導教官

福島 雅夫 教授
山下 信雄 助手

京都大学工学部情報学科
数理工学コース
平成 11 年入学
平成 15 年卒業

榎崎 崇

平成 15 年 1 月 31 日提出

摘要

今日、社会における様々なシステムが多様化・複雑化するに従って、最適化問題も多様で大規模なものとなってきている。そのため、最適化問題を解くアルゴリズムには、適用範囲の広さと計算の速さが要求される。そのような要求を満たすアルゴリズムとして、逐次二次計画法（SQP法）がよく知られている。SQP法は、一般の非線形最適化問題に適用され、多くの場合解の近傍で超一次収束するが、ときにマラトス効果と呼ばれる現象が生じて収束速度が非常に遅くなってしまうことがある。これは、SQP法が部分問題の制約条件に主問題の一次近似を用いるために、引き起こされるものであることが知られている。この点に注目して、近年、制約条件の二次近似を取り入れることでマラトス効果を回避する逐次二次制約二次計画法（SQCQP法）と呼ばれる方法が提案されている。SQCQP法は、適用できる問題が微分可能な凸計画問題に限られるが、大域的収束性と局所的二次収束性が理論的に保証されている。しかし、実際の最適化問題に対してSQCQP法を適用したという報告はまだ行われていない。本報告書の目的はSQCQP法の実装を行い、通信工学の分野に現れる現実的な問題に対して適用することである。今日主として衛星通信において、マルチキャリア・マルチプルアクセスと呼ばれるシステムが用いられている。マルチキャリア・マルチプルアクセスでは、各ユーザの送信信号が相互に干渉しあうため、各ユーザの送信出力を適切に設定する必要がある。これまでこの問題に対して特化されたアルゴリズムは提案されているが、必ずしも効率的というわけではない。本報告書では、この最適化問題に対してSQCQP法を適用し、その有用性をしめす。

目次

1	序論	1
2	逐次二次制約二次計画法	1
2.1	SQCQP 法のアルゴリズム	1
2.1.1	問題	2
2.1.2	部分問題	2
2.1.3	反復点の更新	3
2.1.4	アルゴリズムの記述	4
2.2	理論的収束性	5
3	通信工学における最適化問題	5
3.1	準備	5
3.2	レート最適化問題の定式化	6
3.3	従来のアルゴリズム	7
4	数値実験	8
4.1	定数パラメータの設定	8
4.2	実験方法	9
4.3	実験結果	9
4.4	考察	10
5	結論	11
A	部分問題 (2.3) の二次錐計画問題への変換	13
A.1	二次錐計画問題 (SOCP) とは	13
A.2	QCQP の SOCP への変形	14
B	実験結果のグラフによる表示	15
B.1	反復点の更新の様子	15

1 序論

今日、社会の各分野におけるシステムは多様化・複雑化の一途をたどっている。そのなかで、実際に生じる最適化問題も多様で大規模なものとなってきている。そのため、最適化問題を解くアルゴリズムには、適用範囲の広さと計算の速さが要求される。そのような要求を満たすアルゴリズムとして、逐次二次計画法（SQP法）がよく知られている。SQP法は、一般の非線形最適化問題に適用され、大域的収束性を有するとともに、多くの場合解の近傍で超一次収束するが、ときにマラトス効果と呼ばれる現象が生じて収束速度が非常に遅くなってしまうことがある。これは、SQP法が部分問題の制約条件に主問題の一次近似を用いるために、引き起こされるものであることが知られている。マラトス効果を回避するために、SQP法を修正する方法がいくつか提案されているが、それらの方法ではアルゴリズムは複雑なものとなる。一方、部分問題に制約条件の二次近似を取り入れることでマラトス効果を回避できる可能性が示唆されており [4]、この点に注目して、近年、制約条件の二次近似を取り入れることでマラトス効果を回避する逐次二次制約二次計画法（SQCQP法）と呼ばれる方法が提案されている [2]。SQCQP法は、適用できる問題が微分可能な凸計画問題に限られるが、大域的収束性と局所的二次収束性が理論的に保証されており、マラトス効果の回避とともに、収束の速さも期待できる。しかし、実際の最適化問題に対して SQCQP法を適用したという報告はまだ行われていない。本報告書の目的は SQCQP法の実装を行い、通信工学の分野に現れる現実的な問題に対して適用することである。今日主として衛星通信において、マルチキャリア・マルチプルアクセスと呼ばれるシステムが用いられている。マルチキャリア・マルチプルアクセスでは、各ユーザの送信信号が相互に干渉しあうため、各ユーザの送信出力を適切に設定する必要がある。これまでこの問題に対して特化されたアルゴリズムは提案されているが、必ずしも効率的というわけではない。本報告書では、この最適化問題に対して SQCQP法を適用し、その有用性をしめす。

本報告書では、まず第2節で SQCQP法のアルゴリズムとその収束性について説明する。次に、第3節で、通信工学において実際に生じる最適化問題を紹介し、その定式化について説明する。第4節では、第3節で定式化した問題に対して SQCQP法を適用した数値実験の結果を報告し、それに対する考察を与える。最後に、第5節で結論を述べる。

2 逐次二次制約二次計画法

この節では、初めに逐次二次制約二次計画法（SQCQP法）のアルゴリズムを説明し、次に、その理論的収束性について述べる。

2.1 SQCQP法のアルゴリズム

SQCQP法は文献 [2] において、微分可能な凸計画問題に対して提案されているが、目的関数が非凸関数である場合に対しても適用可能である。しかし、ここでは簡単のために、凸計画問題に限定して議論を進める。

2.1.1 問題

本報告書ではつぎの最適化問題を取り扱う。

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) \leq 0, \quad i = 1, \dots, m \end{aligned} \tag{2.1}$$

ここで, $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$, $c_i: \mathfrak{R}^n \rightarrow \mathfrak{R}, i = 1, \dots, m$ は二回連続的微分可能な凸関数とする. 一般に凸計画問題の制約条件には不等式制約のほかに線形の等式制約も含まれるが, ここでは簡単のために, 制約条件は不等式のものに限るとする. 以下の議論は線形の等式制約が存在する場合にも拡張できる. 問題 (2.1) に対して次の仮定を設ける.

仮定 1

1. 問題 (2.1) は空でない最適解の集合を持つ.
2. スレイター条件を満たす $\bar{x} \in \mathfrak{R}^n$, すなわち, $c_i(\bar{x}) < 0, i = 1, \dots, m$ を満たす \bar{x} が存在する.

問題 (2.1) に対して, l_1 型の厳密なペナルティ関数 $F_r: \mathfrak{R}^n \rightarrow \mathfrak{R}$ を次のように定義する.

$$F_r(x) := f(x) + r \sum_{i=1}^m [c_i(x)]_+ \tag{2.2}$$

ここで, $r > 0$ はペナルティパラメータであり, $[\cdot]_+$ は $\max\{0, \cdot\}$ を意味する. 仮定 1 の下では, 十分大きい $r > 0$ に対して F_r の制約なし最小解と問題 (2.1) の最適解が一致することが分かっている [1, §5.5]. また, スレイター条件を満たす点, すなわちスレイター点が存在するとき, そのような点を有限回の手続きで計算するアルゴリズムが文献 [3] で提案されている.

2.1.2 部分問題

各反復において点 $x \in \mathfrak{R}^n$ が与えられたとする. SQCQP 法の部分問題を, 次の問題 (2.3) として定式化する. すなわち, SQCQP 法は各反復において, 部分問題 (2.3) を d について最小化し, 探索方向を決定する.

$$\begin{aligned} & \text{minimize} && g(x)^T d + \frac{1}{2} d^T B d \\ & \text{subject to} && c_i(x) + g_i(x)^T d + \frac{\alpha_i}{2} d^T G_i(x) d \leq 0, \quad i = 1, \dots, m \end{aligned} \tag{2.3}$$

ここで, $\alpha_i \in [0, 1], g(x) := \nabla f(x), g_i(x) := \nabla c_i(x), G_i(x) := \nabla^2 c_i(x) \in \mathcal{S}^n, i = 1, \dots, m$ とし, $B \in \mathcal{S}^n$ は正定値とする. ただし, \mathcal{S}^n は $n \times n$ 対称行列全体の集合を表す. 部分問題 (2.3) は凸二次制約二次計画問題であるから, 等価な二次錐計画問題に変換できⁱ, 内点法を用いて効率的に解くことができる.

問題 (2.3) に含まれるパラメータ $(\alpha_i)_{i=1}^m$ は部分問題 (2.3) の実行可能性が保証されるよう, 次

ⁱ詳しくは, 付録 A を参照.

の規則 (2.4) によって定められる [2, Lemma2.1],

$$\begin{aligned} s_1 \leq 2s_1 &\implies \alpha_i = 0, \quad i = 1, \dots, m \\ s_3 \leq 2s_1 \leq s_2 &\implies \alpha_i = \begin{cases} 1, & i \in \mathcal{J} \\ 0, & i \notin \mathcal{J} \end{cases} \\ s_3 > 2s_1 &\implies \alpha_i = 1, \quad i = 1, \dots, m \end{aligned} \quad (2.4)$$

ここで, \mathcal{J} および s_1, s_2, s_3 は次式によって定義される.

$$\mathcal{J} := \{i \mid \theta c_i(\bar{x}) \leq c_i(x)\}, \quad (2.5)$$

$$s_1 := \max_{i: c_i(x) > 0} \frac{c_i(x)}{c_i(x) - \vartheta c_i(\bar{x})}, \quad (2.6)$$

$$s_2 := \min \left\{ \min_{i \in \mathcal{J}} \frac{c_i(x) - \vartheta c_i(\bar{x})}{\kappa_i}, 1 \right\}, \quad (2.7)$$

$$s_3 := \min \left\{ s_2, \min_{i \notin \mathcal{J}} \frac{-2(\vartheta - \theta)c_i(\bar{x})}{\kappa_i} \right\} \quad (2.8)$$

ただし, \bar{x} は仮定 1 で与えられたスレイター点, $\theta \in [0, 1)$ と $\vartheta \in (\theta, 1)$ は任意に固定されたパラメータ, $\kappa_i := (\bar{x} - x)^T G_i(x)(\bar{x} - x)$, $i = 1, \dots, m$, とする.

このとき, 部分問題 (2.3) はスレイター条件を満足する実行可能解を持ち [2], さらにカルーシュ・キューン・タッカー条件 (KKT 条件) を満たすラグランジュ乗数ベクトル $v = (v_1, \dots, v_m)^T$ が存在して,

$$\begin{aligned} Bd + g(x) + \sum_{i=1}^m v_i(\alpha_i G_i(x)d + g_i(x)) &= 0, \\ c_i(x) + g_i(x)^T d + \frac{\alpha_i}{2} d^T G_i(x)d &\leq 0, \quad v_i \geq 0, \\ v_i \left(c_i(x) + g_i(x)^T d + \frac{\alpha_i}{2} d^T G_i(x)d \right) &= 0, \quad i = 1, \dots, m \end{aligned} \quad (2.9)$$

が成り立つことが知られている [8, Theorem 28.2]. この (d, v) を部分問題 (2.3) の KKT 点と呼ぶ.

2.1.3 反復点の更新

d を部分問題 (2.3) の最適解とすると, 次の反復 x^{new} はステップ幅 β を用いて

$$x^{new} := x + \beta d$$

と与えられる. アルゴリズムの大域的収束性を保証するために, $\beta > 0$ は次式を満たすように選ばれる.

$$F_r(x + \beta d) - F_r(x) \leq \sigma \beta (\bar{F}_r(x, d, \alpha) - F_r(x)) \quad (2.10)$$

ここで, $\sigma \in (0, 1)$ は適当な定数, $\alpha = (\alpha_i)_{i=1}^m$ であり, $\bar{F}_r(x, d, \alpha)$ は $G(x) := \nabla^2 f(x)$ をもちいて次式であたえられる.

$$\bar{F}_r(x, d, \alpha) := f(x) + g(x)^T d + \frac{1}{2} d^T G(x)d + r \sum_{i=1}^m \left[c_i(x) + g_i(x)^T d + \frac{\alpha_i}{2} d^T G_i(x)d \right]_+$$

d は部分問題 (2.3) の実行可能解であるから, $c_i(x) + g_i(x)^T d + \frac{1}{2} \alpha_i d^T G_i(x) d \leq 0$ となるので,

$$\bar{F}_r(x, d, \alpha) - F_r(x) = g(x)^T d + \frac{1}{2} d^T G(x) d - r \sum_{i=1}^m [c_i(x)]_+ \quad (2.11)$$

が成り立つ.

ここで, 任意の $x \in \mathfrak{R}^n$ と $\alpha = (\alpha_i)_{i=1}^m$ に対して, 部分問題 (2.3) の KKT 点 (d, v) が存在すると仮定する. さらに,

$$2B - G(x) + \sum_{i=1}^m \alpha_i v_i G_i(x) \succeq \mu I \quad (2.12)$$

を満たすような定数 $\mu > 0$ が存在して, $r \geq \max_i v_i$ が成り立つと仮定すると, 式 (2.11) の右辺を

$$g(x)^T d + \frac{1}{2} d^T G(x) d - r \sum_{i=1}^m [c_i(x)]_+ \leq -\frac{1}{2} \mu \|d\|^2 \quad (2.13)$$

と評価でき, 次式のように, d がペナルティ関数の減少方向となることがいえる [2, Lemma3.1].

$$\bar{F}_r(x, d, \alpha) - F_r(x) \leq -\frac{1}{2} \mu \|d\|^2 \quad (2.14)$$

さらに, $d \neq 0$ であれば, 十分小さいすべての $\beta > 0$ に対して, 式 (2.10) が成り立つ [2, Lemma3.2].

2.1.4 アルゴリズムの記述

SQCQP 法のアルゴリズムは, つぎのように記述される.

step0: 初期点 $x^1 \in \mathfrak{R}^n$, ペナルティパラメータの初期値 $r^0 \in (0, +\infty)$, 定数 $\gamma \in (0, 1)$, $\theta \in [0, 1)$, $\vartheta \in (\theta, 1)$, $\sigma \in (0, 1)$ および, スレイター条件を満たす \bar{x} を選ぶ. $k := 1$ とおく.

step1: 正定値対称行列 B^k を選ぶ. (2.4) に従って $\alpha^k = (\alpha_i^k)_{i=1}^m$ を決定する. $(x, B, \alpha) = (x^k, B^k, \alpha^k)$ に対する部分問題 (2.3) を解き, KKT 点 $(d^k, v^k) \in \mathfrak{R}^n \times \mathfrak{R}^m$ を見つける. もし, $d^k = 0$ であれば終了ⁱⁱ. そうでなければ step2 へ.

step2: ペナルティパラメータを

$$r^k := \begin{cases} r^{k-1} & r^{k-1} \geq \max_i v_i^k + \delta \text{ の時} \\ \max_i v_i^k + 2\delta & \text{ そうでない時} \end{cases} \quad (2.15)$$

として更新し, 次式を満たす最大の β^k を $\{1, \gamma, \gamma^2, \gamma^3 \dots\}$ から選ぶ.

$$F_{r^k}(x^k + \beta d^k) - F_{r^k}(x^k) \leq \sigma \beta (\bar{F}_{r^k}(x^k, d^k, \alpha^k) - F_{r^k}(x^k))$$

そして, $x^{k+1} := x^k + \beta^k d^k$, $k := k + 1$ と更新し, step1 へ.

ⁱⁱこの時, (x^k, v^k) は問題 (2.1) の KKT 点となっており, 確かに x^k は問題 (2.1) の最適解である.

2.2 理論的収束性

まず, SQCQP 法の大域的収束性について述べる. 大域的収束性については, 次の定理 1 が知られている [2, Theorem3.1].

定理 1 SQCQP 法により生成された点列 $\{x^k\}, \{B^k\}, \{\alpha^k\}$ に対して, 点列 $\{x^k\}$ は有界であり, さらに, すべての k について次式が成り立つような $\mu > 0$ と $\mu_1 \geq \mu_2 > 0$ が存在すると仮定する.

$$2B^k - G(x^k) + \sum_{i=1}^m \alpha_i^k v_i^k G_i(x^k) \succeq \mu I \quad \text{かつ} \quad \mu_1 I \succeq B^k \succeq \mu_2 I$$

そのとき, SQCQP 法は問題 (2.1) の最適解において停止するか, もしくは, $\{x^k\}$ の任意の集積点は問題 (2.1) の最適解となる.

次に, SQCQP 法の局所的な収束性について述べる. 局所的収束性について, 次の仮定 2 のもとで定理 2 が成り立つことが知られている [2, Theorem4.1].

仮定 2

1. 問題 (2.1) は, ある定数 $\mu^* > 0$ に対して,

$$H(x^*, v^*) \succeq \mu^* I, \quad \forall v^* \in V^*$$

をみたす最適解 x^* をもつ. ただし, $H(x, v) := G(x) + \sum_{i=1}^m v_i G_i(x), V^* := \{v^* \in \mathbb{R}^m \mid (x^*, v^*)$ は問題 (2.1) の KKT 点である.

2. G および G_1, \dots, G_m は x^* の近傍においてリプシッツ連続である.

定理 2 SQCQP 法において初期点 x^1 を最適解 x^* の十分近くに選び, かつ, すべての k について $B^k := G(x^k)$ とすれば, 生成される点列 $\{x^k\}$ は x^* に Q -2 次収束する. さらに, $\text{dist}(v^k, V^*)$ は 0 に R -2 次収束する.ⁱⁱⁱ

3 通信工学における最適化問題

本節では, 通信工学における現実的な問題の一つである, 符号間干渉の生じるチャンネル上でのマルチキャリア・マルチプルアクセスにおけるレート最適化問題について説明し, 凸最適化問題としての定式化を行う.

3.1 準備

本節で用いられるいくつかの用語について説明する.

- レートと

レートとは, 信号の送信側から受信側にどれだけの情報が正確に伝送されたかを意味するものであり, レートと和は, システム内ユーザ全体でのレートの和を意味する.

ⁱⁱⁱ $\text{dist}(v, V) = \min\{\|v - \bar{v}\| \mid \bar{v} \in V\}$

- マルチプルアクセス
マルチプルアクセスとは複数のユーザが無線通信路を同時に共有して通信を行うことをいう。このため、限られた周波数資源を有効に活用する必要がある。
- マルチキャリア
正確には、マルチキャリア変調方式をという。マルチキャリア変調方式とは、情報を複数のサブキャリアと呼ばれる低速なキャリアに分けて伝送する方式である。サブキャリアの数を N とすれば、送信側においてデジタル情報列を N 個の並列符号に変換し、つぎに、その符号を N 個の異なる周波数のサブキャリアの和に変換してマルチキャリア変調信号を生成する。その後、ガードインターバルを挿入して信号を送信する。受信側では、逆の操作によって復号を行う。
- マルチキャリア・マルチプルアクセス
マルチプルアクセスにおいて、マルチキャリア変調方式を採用したもので、複数のユーザが複数のキャリアを共有して通信を行う方式である。

マルチキャリア・マルチプルアクセスでは、複数のユーザが同時に複数のキャリアを共有するため、個々の送信符号間で干渉が生じ、情報がすべて正確に伝送されるわけではない。仮に信号の出力に制約がないとすれば、ユーザの信号の出力を上げることで、伝送される情報量を増加させることができる。しかし、現実には各ユーザの出力には物理的な制限があり、あるユーザの出力を上げたとき同じキャリアを使用する他のユーザのレートは下がってしまうため、単純に信号の出力を上げることで問題は解決しない。したがって、それらの制約のもとでシステム全体のレートとを最大化するような出力を求めることは重要な課題である。

3.2 レートと最適化問題の定式化

マルチキャリア・マルチプルアクセスにおける達成可能なレートとをの最大値は、送・受信信号間における相互情報量で表され、出力に関する制約と併せて次のように定式化されることが知られている [7].

$$\begin{aligned}
 & \text{maximize} && \sum_{c=1}^C \log \left(1 + \rho_c \sum_{u=1}^U |H_u(\omega_c)|^2 p_{u,c} \right) \\
 & \text{subject to} && \sum_{c=1}^C p_{u,c} \leq P_u \quad u = 1, \dots, U
 \end{aligned} \tag{3.1}$$

ここで、添え字 u, c はそれぞれユーザ番号とキャリア番号を表し U, C はユーザ数とキャリア数を表す。また、 $H_u(\omega_c)$ は u 番目のユーザの c 番目のキャリアにおける、周波数 $\omega_c = 2\pi c/C$ の伝達関数であり、 ρ_c は c 番目のキャリアにおける相関係数を表している。ところで、問題 (3.1) の目的関数は凹関数であり、 $H_u(\omega_c)$ と ρ_c はそれぞれ、添え字に依存する定数であることに注意すると、式

(3.1) は次のような凸計画問題として定式化できる.

$$\begin{aligned} & \text{minimize} && -\sum_{c=1}^C \log \left(1 + \sum_{u=1}^U \frac{\alpha_{u,c} p_{u,c}}{n_c} \right) \\ & \text{subject to} && \sum_{c=1}^C p_{u,c} \leq P_u \quad u = 1, \dots, U \end{aligned} \quad (3.2)$$

ただし, $n_c = 1/\rho_c$ は c 番目のキャリアにおけるノイズパワーをあらわし, $\alpha_{u,c} = |H_u(\omega_c)|^2$ は u 番目のユーザの c 番目のキャリアにおける減衰率を表している.

ここで式(3.2)を現実にもっとした形に一般化することを考える. 一般化に際して, 次のことを考慮する.

1. 式(3.2)の目的関数は, 全てのキャリアについて平等にレートと和をとっているが, 現実にはキャリアの間に優先順位が存在すると考える方が自然である. そこで, レートに重みを付けて和をとるようにする. これを, 重み付レートと和と呼ぶことにする.
2. ユーザ側からみて, 各キャリアへの出力として望ましい値があると考えられる. そこで, 目的関数に重み付レートと和だけでなく, ユーザの要望が考慮されるような項を付け加える.
3. 優先順位をつけた場合で, ある程度のレートを保証する必要があるユーザも存在すると考えられる. そこで, 特定のユーザのレートを保証するような制約条件を追加する.
4. 各キャリアに対するユーザの出力に対して, ある一定の制限を設ける.

1.,2.に留意して目的関数を修正し, 3.,4.のそれぞれを第2, 第3の制約条件として追加することにより, 次の問題(3.3)を得る.

$$\begin{aligned} & \text{minimize} && -\sum_{c=1}^C w_c \log \left(1 + \sum_{u=1}^U \frac{\alpha_{u,c} p_{u,c}}{n_c} \right) + \epsilon \sum_{c=1}^C \sum_{u=1}^U (p_{u,c} - q_{u,c})^2 \\ & \text{subject to} && \sum_{c=1}^C p_{u,c} \leq P_u \quad (\text{すべての } u \text{ に対して}) \\ & && \sum_{c=1}^C \log \left(1 + \frac{\alpha_{u,c} p_{u,c}}{n_c} \right) \geq I_u \quad (\text{いくつかの } u \text{ に対して}) \\ & && 0 \leq p_{u,c} \leq \bar{P} \quad (\text{すべての } u \text{ に対して}) \end{aligned} \quad (3.3)$$

ここで, $\rho > 0$ は重み付レートとユーザの希望する出力との比重をあらわす定数であり, $q_{u,c}, u = 1, \dots, U, c = 1, \dots, C$ はユーザ u が望むキャリア c への出力をあらわす. また, $\sum_{c=1}^C \sum_{u=1}^U (p_{u,c} - q_{u,c})^2$ という項は, 2. を反映するためのものであり, $p_{u,c} = q_{u,c}$ のときに最小となり, ユーザの出力 $p_{u,c}$ が望ましい値 $q_{u,c}$ から離れるほど大きい値をとるようになっている. 問題(3.3)は微分可能な凸計画問題であるから, SQCQP法によって解を求めることができる.

3.3 従来のアルゴリズム

文献[7]で問題(3.1)に対して特化されたアルゴリズムが提案されている. ここでは, そのアルゴリズムを問題(3.3)にも適用できるように修正したものを示す.

まず、アルゴリズム中で用いる部分問題を問題 (3.4) として定式化する。

$$\begin{aligned}
& \text{minimize} && - \sum_{c=1}^C w_c \log \left(1 + \sum_{v \neq u} \frac{\alpha_{v,c} p_{v,c}}{n_c} + \frac{\alpha_{u,c} p_{u,c}}{n_c} \right) \\
& && + \epsilon \left\{ \sum_{c=1}^C \sum_{v \neq u} (p_{v,c} - q_{v,c})^2 + \sum_{c=1}^C (p_{u,c} - q_{u,c})^2 \right\} \\
& \text{subject to} && \sum_{c=1}^C p_{u,c} \leq P_u \\
& && \sum_{c=1}^C \log \left(1 + \frac{\alpha_{u,c} p_{u,c}}{n_c} \right) \geq I_u \\
& && 0 \leq p_{u,c} \leq \bar{P}
\end{aligned} \tag{3.4}$$

この部分問題 (3.4) の変数はユーザ u の出力 $p_{u,c}$ のみであり、 u 以外のユーザの出力、すなわち、 $p_{v,c} (v \neq u)$ はすべて定数とみなしている。この部分問題 (3.4) は特定のユーザ u に関する最適化問題と考えることができる。修正したアルゴリズムを次にしめす。

step0: $p_{u,c}^1 = 0, u \in [1, U], c \in [1, C], k := 1$ と初期化し、定数 $\epsilon > 0$ を選ぶ。step1 へ。

step1: $\bar{p}_{u,c} := p_{u,c}^k, u = 1, \dots, U, c = 1, \dots, C$ および、 $u := 1$ とする。step2 へ。

step2: ユーザ u に関する部分問題 (3.4) を解き、最適解 $p_{u,c}^*, c = 1, \dots, C$ を求め、 $p_{u,c}^k := p_{u,c}^*, c = 1, \dots, C$ と更新する。 $u = U$ であれば step3 へ。 そうでなければ $u := u + 1$ として step2 へ。

step3: $\delta^k := \sum_u \sum_c |\bar{p}_{u,c} - p_{u,c}| < \epsilon$ であれば終了。さもなければ、 $k := k + 1$ として step1 へ。

このアルゴリズムを、アルゴリズム 1 と呼ぶことにする。

4 数値実験

本節では、最適化問題 (3.3) に対して、SQCQP 法およびアルゴリズム 1 を適用し、その結果を報告する。

4.1 定数パラメータの設定

最適化問題 (3.3) に含まれる定数パラメータは、現実の問題に対して妥当なものでなければならぬ。そこで、各定数パラメータの設定方針を以下に示す。

ρ 目的関数の評価基準を左右する値である。log 関数に比べ 2 次関数は値の変化量が非常に大きいので、 $10^{-6} \leq \rho \leq 10^{-4}$ 程度とする。

w_c 重み係数は 1 から 10 までの整数値で、無作為に設定する。

$\alpha_{u,c}$ 減衰率を表すので、 $\alpha_{u,c} \in (1, 0)$ となるように無作為に設定する。

- n_c 一般に $10 \leq n_c \leq 20$ であるので、この範囲の整数値で無作為に設定する。
- P_u 一般に、ユーザの総出力はサブキャリア数の 500 倍程度となっている。本実験では、サブキャリア数の 480 倍から 520 倍の範囲に収まるように無作為に設定する。
- I_u ある特定のキャリアに対する出力を最大にしたときのレートは 5 ないし 20 程度となるので、この範囲でランダムに設定する。
- \bar{P} 一般に、出力の最大値 \bar{P} は 1000 程度とされている。本実験では、 $\bar{P} := 1000$ と固定する。
- $q_{u,c}$ $p_{u,c}$ の範囲は 0 以上 1000 以下であるから、その範囲内に収まるように無作為に設定する
- また、ユーザ数 U は $5 \leq U \leq 10$ 、サブキャリア数 C は $8 \leq C \leq 16$ 程度で問題を設定する。
- 最後に、問題 (3.3) の第 2 の制約条件を適用するユーザの特定方法を述べる。ユーザに関する定数パラメータは無作為に選ぶので、ユーザ番号を指定するのではなく、制約条件が適用されるユーザの人数 s を決定して、ユーザ $1, \dots, s$ に対して第 2 の制約条件を適用すればよい。したがって、 s は $0 \leq s \leq U$ の範囲で無作為に設定する。

4.2 実験方法

実験の具体的な方法は以下のとおりである。まず最適化問題 (3.3) のユーザ数、サブキャリア数、および定数パラメータの値を妥当な範囲内で決定する。そして、SQCQP 法およびアルゴリズム 1 を適用して最適解の近似解を求める。ただし、SQCQP 法は $\|d^k\| < 10^{-6}$ となれば終了するものとし、アルゴリズム 1 は $\varepsilon := 10^{-6}$ として、終了を判定する。また、いずれのアルゴリズムにおいても 1 回の反復での目的関数の減少が 10^{-10} 以下となった場合は、最適解に十分近づいたとみなし、アルゴリズムを終了する。なお、反復は最大で 100 回までとする。ただし、アルゴリズム 1 での 1 回の反復とは、step1 から step3 までの処理をさす。

また、実験で用いる数値例は、4.1 節で述べた方法で決定する。したがって、恣意的に決定する値は、ユーザ数 U 、サブキャリア数 C 、および、 ϵ となる。

4.3 実験結果

本実験は DELL 社の DIMENSION8200 上で、turbolinux 8 の環境のもとで行った。また、SQCQP 法の部分問題の SOCP を解くにあたり、Jos F. Sturm 氏が提供する MATLAB のソルバ SeDuMi1.05 を使い、アルゴリズム 1 の部分問題を解くにあたっては、MATLAB の Optimization Toolbox が提供する fmincon 関数を用いた。

実験に用いる数値例は、4.1 節で述べた方法で決定したものを 4 種類用意した。それらを数値例 1、数値例 2、数値例 3、数値例 4 とする。それぞれの数値例に対して SQCQP 法とアルゴリズム 1 および SQP 法を適用した。実験で使用した数値例のパラメータとその結果のグラフの対応を表 1 に、各数値例におけるそれぞれのアルゴリズムの近似最適解および実行時間（単位は秒）を表 2 に示す。また、得られた結果のうち SQCQP 法および SQP 法の $\|d^k\|$ とアルゴリズム 1 の δ^k に

関する結果を B.1 節にグラフとして示した. それぞれのグラフでは, 横軸に反復回数 k をとり, 縦軸には $\log_{10} \|d^k\|$ または $\log_{10}(\delta^k)$ をとった.

表 1 : 実験に用いた数値例

数値例	パラメータ			グラフ
	U	C	S	
1	5	8	2	図 1
2	5	16	2	図 2
3	10	8	5	図 3
4	10	16	2	図 4

表 2 : 数値例と各アルゴリズムの結果

数値例	検証結果					
	SQCQP 法		アルゴリズム 1		SQP 法	
	近似最適解	実行時間	近似最適解	実行時間	近似最適解	実行時間
1	-303.916579	14.1818	-302.934261	8.3152	-303.916579	12.0804
2	-469.028033	46.6824	-468.428752	13.8602	-469.028033	44.0745
3	-328.256035	45.5442	-327.108074	15.6066	-328.256035	49.3501
4	-624.208358	499.5916	-622.889778	24.4265	-624.208358	495.6086

4.4 考察

まず, 図 1~4 をみると, SQCQP 法の局所的超一次収束性が確認できる. また, レート和最適化問題に対して, 従来のアルゴリズムでは最適解近傍での収束速度が非常に遅くなっており, 反復回数が上限の 100 回に達しても収束していない. これは, 表 2 の近似最適値の項目をみると, アルゴリズム 1 の近似最適値が SQCQP のそれよりも大きい, すなわち, 最適性が劣っていることから推測できる. さらに, SQCQP 法はいずれの数値例に対しても 11~13 回で収束しているので, 収束するまでの反復回数は問題の規模にはほとんど依存しないと考えてよい.

次に, 実行時間に注目してみる. 図 1~4 をみると, アルゴリズム 1 は反復回数が上限の 100 回に達しているにも関わらず, 終了していない. しかし, 表 1 に注目すると, SQCQP 法に比べて実行時間が短いことがわかる. SQCQP 法は, いずれの数値例の場合においても反復回数は 13 回を越えていないので, 1 回の反復に要する時間はアルゴリズム 1 の方が圧倒的に少ないということがわかる. ただし, 図 1~図 4 をみると, 途中から δ^k の値がほとんど変化していないことが分かる. これでは, 近似解が真の最適解の近傍に近づいたかどうかを判定することは困難である. 一方, SQCQP 法の実行時間であるが, 表 1, 2 より, 問題の規模が大きくなるに従い, かなり大きくなっていることが分かる. SQCQP 法を実装するのであれば, この点に留意する必要があるといえる.

5 結論

本報告書では、微分可能な凸最適化問題に対して提案されている SQCQP 法が、局所的超一次収束を示すことを確認した。そして、通信工学におけるある種の凸最適化問題に対して、従来より提案されているアルゴリズムと比較して、よりよい解が得られることを確認した。

その一方、大規模問題を扱う上では、実行時間の面で難点があると考えられ、今後、実装の際の課題となると思われる。本報告書では、反復回数は問題の規模にあまり依存しないことが分かったので、部分問題の QCQP を解く効率的な手法が提案されれば、SQCQP 法の実行時間は短縮され、大規模問題に対する適用も容易になると考えられる。

謝辞

日頃から御教授いただき、本研究に対しても貴重な御指摘と御指導を賜りました福島雅夫教授、本報告書で扱った問題に対して貴重な御指導をいただきました McMaster 大学の Z.-Q. Luo 教授、ならびに山下信雄助手に深く感謝の意を表します。また、大変お世話になった滝根哲哉助教授をはじめとする福島研究室の皆様には厚く御礼申し上げます。

参考文献

- [1] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, NY, 1982.
- [2] M. Fukushima, Z.-Q. Luo and P. Tseng, *A sequential quadratically constrained quadratic programming method for differentiable convex minimization*, SIAM J. Optimization, 12 (2002), pp. 436–460.
- [3] M. Fukushima, *A finitely convergent algorithm for convex inequalities*, IEEE Trans. Autom. Contr., AC-27 (1982), pp. 1126–1127.
- [4] M. Fukushima, *A successive quadratic programming algorithm with global and superlinear convergence properties*, Math. Programming, 35 (1986), pp. 253–264.
- [5] Z.-Q. Luo and S. Zhang, *On extensions of Frank-Wolfe theorems*, Comput. Optim. Appl., 13 (1999), pp. 87–110.
- [6] M.S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, *Applications of second-order cone programming*, Linear Algebra Appl., 284 (1998), pp. 193–228.
- [7] S. Ohno, P. Anghel, G. B. Giannakis, and Z.-Q. Luo, *Multi-carrier multiple access is sum-rate optimal for block transmissions over circulant ISI channels*, Proc. of Intl. Conf. on Communications, vol. 3., pp. 1656-1660, New York City, N.Y., April 28-May 2, 2002.
- [8] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

付録

A 部分問題 (2.3) の二次錐計画問題への変換

部分問題 (2.3) は二次制約二次計画問題 (QCQP) と呼ばれる問題のクラスに属する. この節では, 一般的に, 次の QCQP を二次錐計画問題 (SOCP) に変換する方法を紹介する.

$$\begin{aligned} & \text{minimize} && x^T P_0 x + 2q_0^T x + r_0 \\ & \text{subject to} && x^T P_i x + 2q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \end{aligned} \tag{A.1}$$

ただし, P_0, P_i は半正定値対称であるとする. 特に, $P_0 := \frac{1}{2}B, q_0 := \frac{1}{2}g(x), r_0 := 0$, および, $P_i := \frac{\alpha_i}{2}G_i(x), q_i := \frac{1}{2}g_i(x), r_i := c_i(x), i = 1, \dots, m$, とすれば (A.1) は問題 (2.3) に帰着する.

A.1 二次錐計画問題 (SOCP) とは

一般に, SOCP は次のように表される.

$$\begin{aligned} & \text{minimize} && f^T x \\ & \text{subject to} && \|A_i x + b_i\| \leq c_i^T x + d_i, \quad i = 1, \dots, N \end{aligned} \tag{A.2}$$

ここで, $x \in \mathfrak{R}^n$ は変数であり, $f \in \mathfrak{R}^n, A_i \in \mathfrak{R}^{(n_i-1) \times n}, b_i \in \mathfrak{R}^{n_i-1}, c_i \in \mathfrak{R}^n, d_i \in \mathfrak{R}$ はいずれも与えられた定数パラメータである.

二次錐とは

$$\mathcal{K}^j = \left\{ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| x_1 \in \mathfrak{R}, x_2 \in \mathfrak{R}^{j-1}, \|x_2\| \leq x_1 \right\}$$

と定義される集合であり, これを次元 j の二次錐と呼ぶ. 特に, $j = 1$ であれば

$$\mathcal{K}^1 = \{x_1 | x_1 \in \mathfrak{R}, 0 \leq x_1\}$$

となる. また, 定義より

$$\|A_i x + b_i\| \leq c_i^T x + d_i \iff \begin{bmatrix} c_i^T \\ A_i \end{bmatrix} x + \begin{bmatrix} d_i \\ b_i \end{bmatrix} \in \mathcal{K}^{n_i}$$

であることが分かる. このことを利用すると, SOCP を次のように書くこともできる.

$$\begin{aligned} & \text{minimize} && f^T x \\ & \text{subject to} && \bar{A}_i x + \bar{b}_i \in \mathcal{K}^{n_i}, \quad i = 1, \dots, N \end{aligned} \tag{A.3}$$

ただし, $\bar{A}_i = (c_i, A_i^T)^T, \bar{b}_i = (d_i, b_i^T)^T$ である.

A.2 QCQP の SOCP への変形

QCQP(A.1) は, 新たに変数 t を導入することによって, 次のように書き換えられる.

$$\begin{aligned} & \text{minimize} && t \\ & \text{subject to} && x^T P_0 x + 2q_0^T x + r_0 \leq t \\ & && x^T P_i x + 2q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \end{aligned} \quad (\text{A.4})$$

ここで, 次の方法を適用することにより, (A.4) を (A.2) の形に変換することができる. なお, $y = (t, x^T)^T$ とする.

1. $P_0 \succ O$ または $P_i \succ O$ の場合は, それらに対応する制約条件は $\|P_0^{1/2}x + P_0^{-1/2}q_0\| \leq t$ または $\|P_i^{1/2}x + P_i^{-1/2}q_i\| \leq (q_i P_i^{-1} q_i - r_i)^{1/2}$ と SOCP の制約条件に変換される [6].
2. $P_0 = O$ または $P_i = O$ の場合は, それらに対応する制約条件は $2q_0^T x + r_0 \leq t, 2q_i^T x + r_i \leq 0$ と線形になるので, それぞれ $\bar{q}_0^T = (-1, 2q_0^T), \bar{q}_i^T = (0, 2q_i^T)$ とおくことで, $0 \leq -\bar{q}_0^T y - r_0, 0 \leq -\bar{q}_i^T y - r_i$ と 1 次元の SOCP の制約条件に変換できる.
3. 上記のいずれにも当てはまらない場合, すなわち, $P_0 \neq O$ かつ $P_0 \not\succeq O$ または $P_i \neq O$ かつ $P_i \not\succeq O$ の場合には,

$$\bar{P}_0 = \begin{pmatrix} 0 & 0 \\ 0 & P_0 \end{pmatrix}, \quad \bar{P}_i = \begin{pmatrix} 0 & 0 \\ 0 & P_i \end{pmatrix}, \quad \bar{q}_0^T = (-1, 2q_0^T), \quad \bar{q}_i^T = (0, 2q_i^T)$$

とおく. すると, 制約条件は

$$y^T \bar{P}_0 y + \bar{q}_0^T y + r_0 \leq 0, \quad y^T \bar{P}_i y + \bar{q}_i^T y + r_i \leq 0$$

と変換できる. $\bar{P}_0 \succeq O, \bar{P}_i \succeq O$ であるから, $\bar{P}_0 = C_0^T C_0, \bar{P}_i = C_i^T C_i$ と分解できる. 従って, 制約条件は

$$\begin{pmatrix} 1 - r_0 \\ 1 + r_0 \\ 0 \end{pmatrix} - \begin{pmatrix} q_0^T \\ -q_0^T \\ -2C_0 \end{pmatrix} y \in \mathcal{K}^{\gamma_0+2}, \quad \begin{pmatrix} 1 - r_i \\ 1 + r_i \\ 0 \end{pmatrix} - \begin{pmatrix} q_i^T \\ -q_i^T \\ -2C_i \end{pmatrix} y \in \mathcal{K}^{\gamma_i+2}$$

となる. ただし $\gamma_i = \text{rank} C_i, i = 0, 1, \dots, m$ である.

B 実験結果のグラフによる表示

B.1 反復点の更新の様子

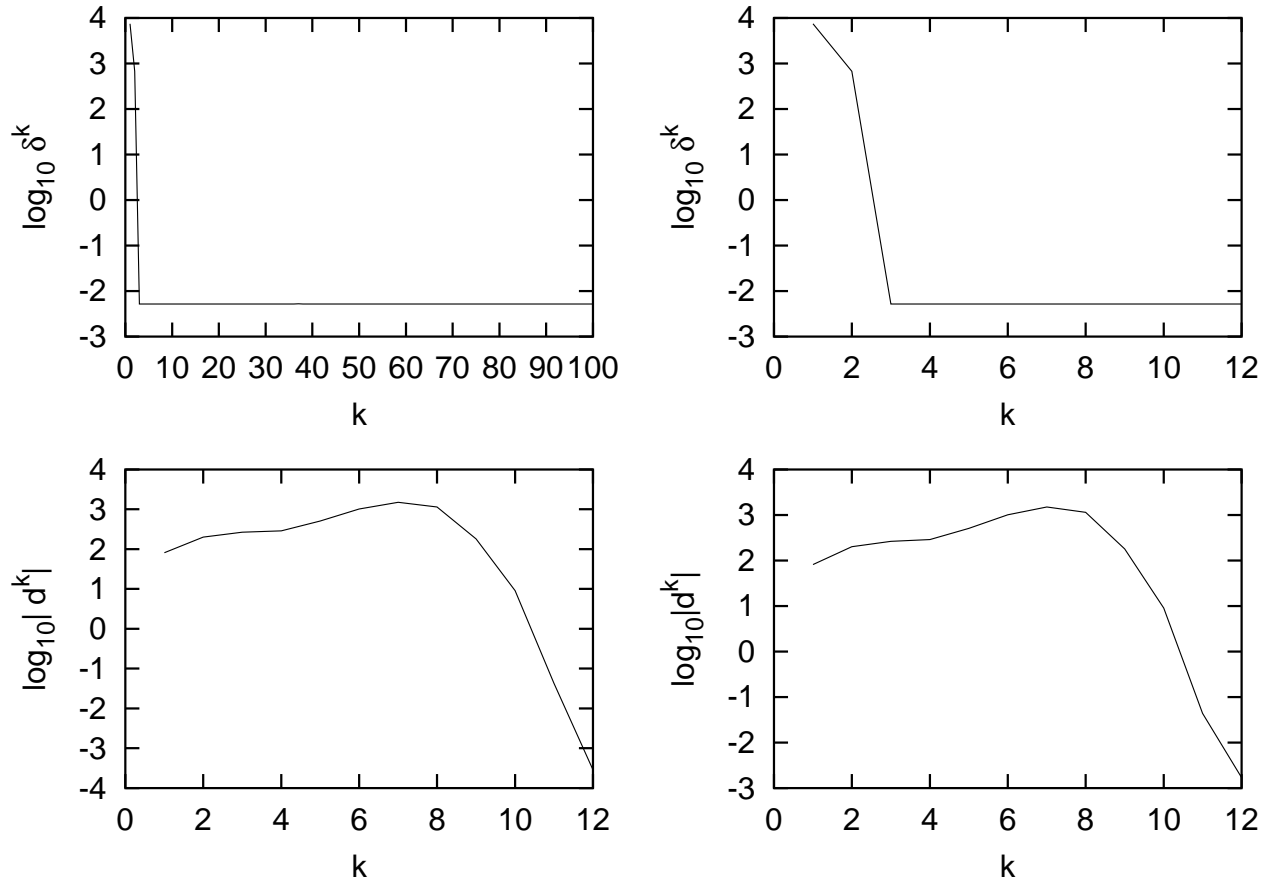


図 1: 数値例 1, アルゴリズム 1 (左上: $k \leq 100$, 右上: $k \leq 12$), SQCQP 法 (左下), SQP 法 (右下)

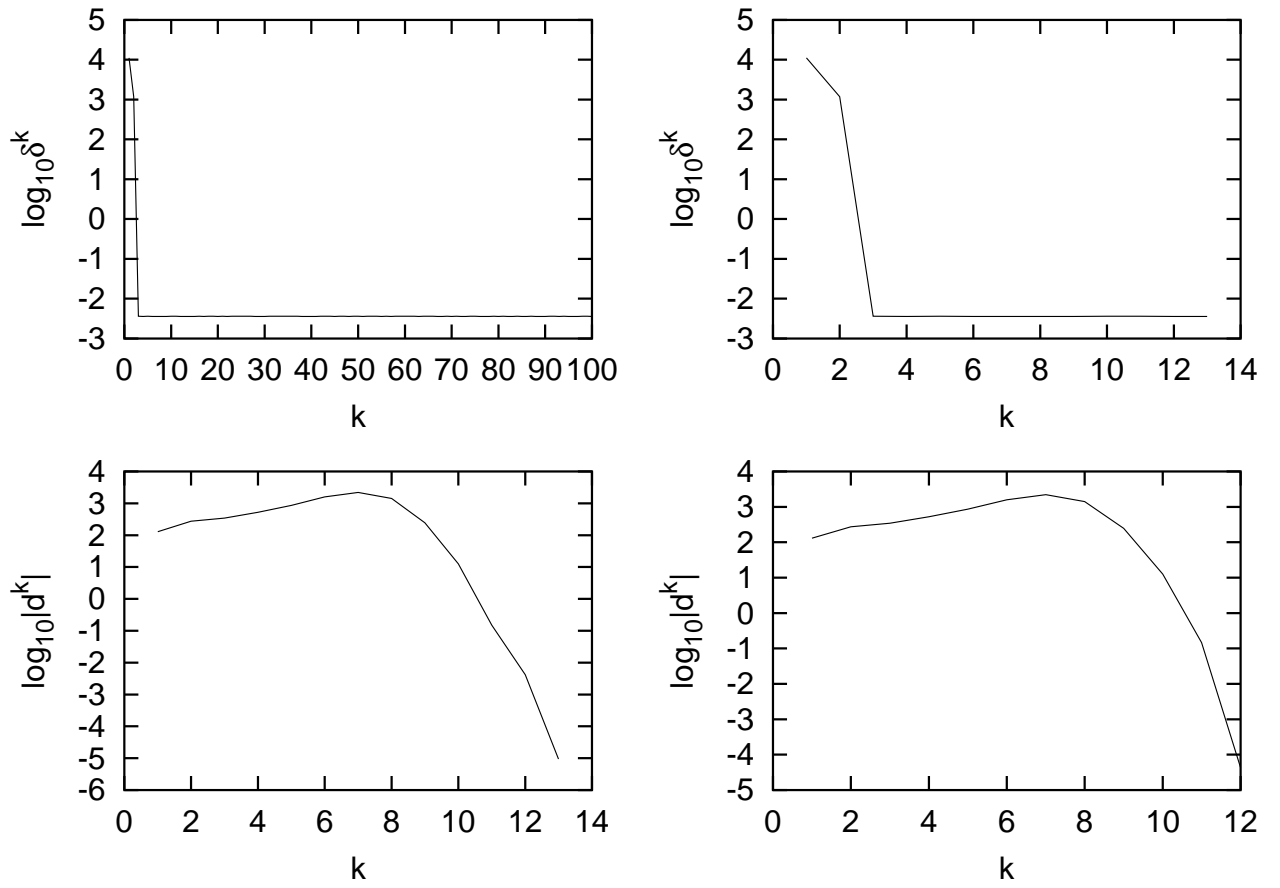


図 2: 数値例 2, アルゴリズム 1 (左上: $k \leq 100$, 右上: $k \leq 13$), SQCQP 法 (左下), SQP 法 (右下)

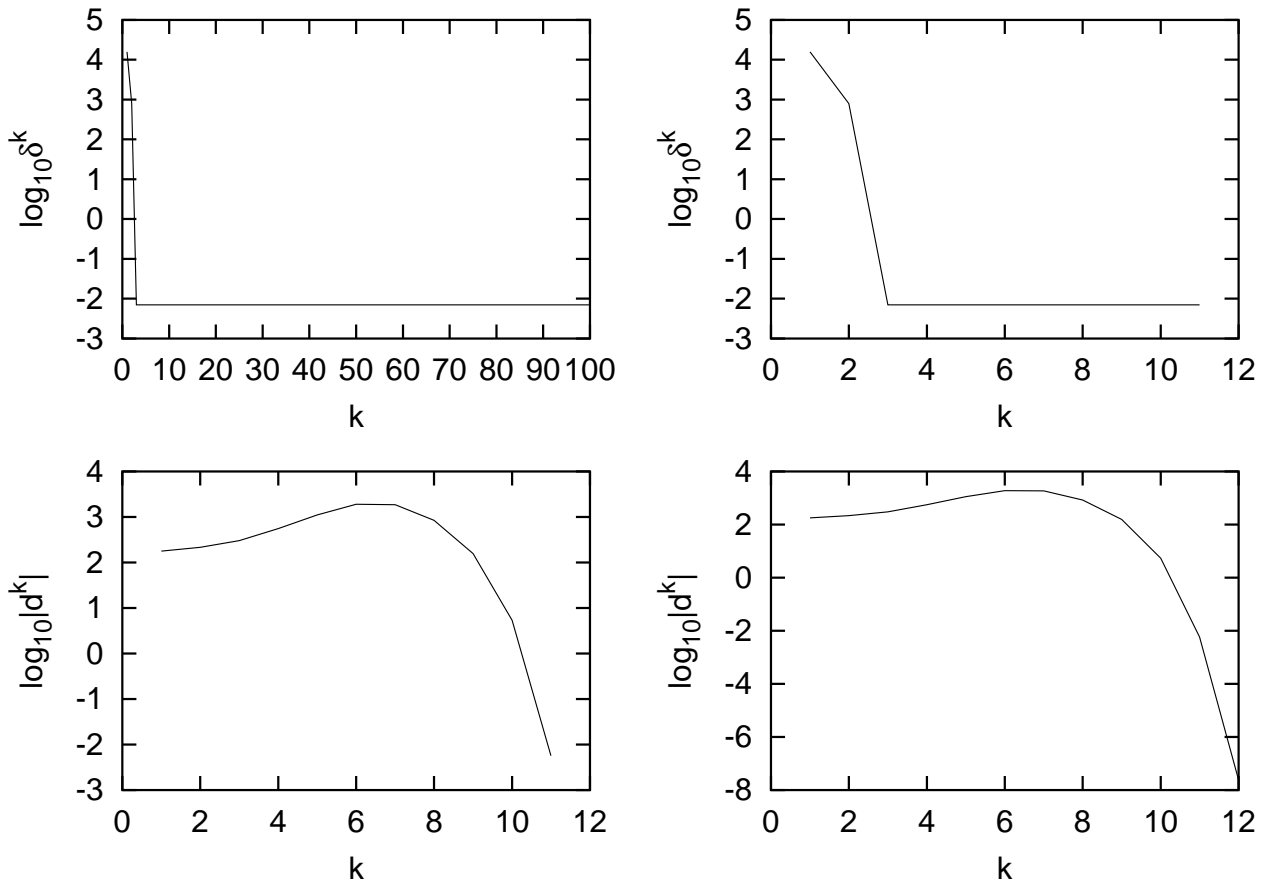


図 3: 数値例 3, アルゴリズム 1 (左上: $k \leq 100$, 右上: $k \leq 11$), SQCQP 法 (左下), SQP 法 (右下)

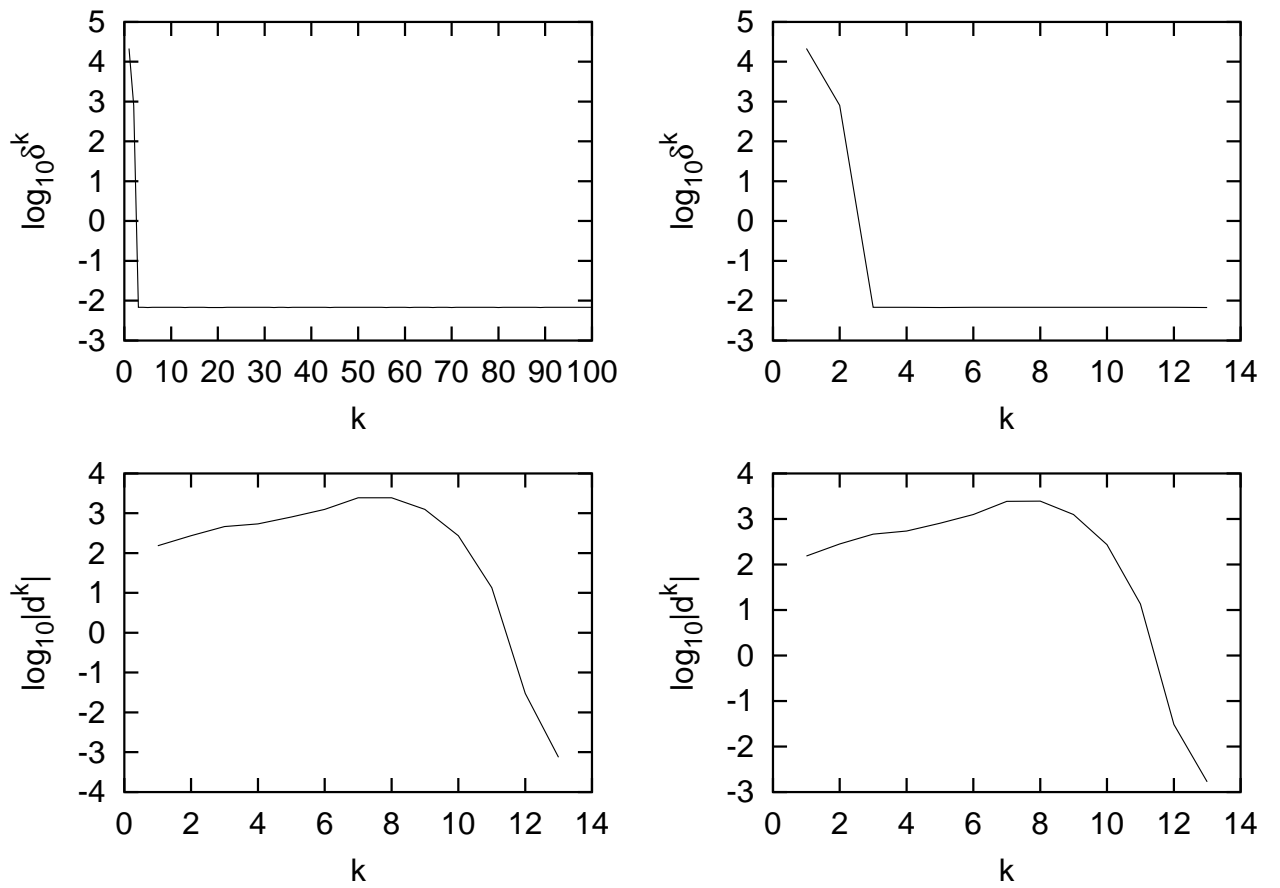


図 4: 数値例 4, アルゴリズム 1 (左上: $k \leq 100$, 右上: $k \leq 13$), SQCQP 法 (左下), SQP 法 (右下)