

無制約最適化問題に対する降下方向を生成する 拡張三項共役勾配法の大域的収束性

東京理科大学・数理情報科学科 矢部 博 (Hiroshi Yabe)
Department of Mathematical Information Science,
Tokyo University of Science

横浜国立大学・経営システム科学科 成島 康史 (Yasushi Narushima)
Department of Management System Science,
Yokohama National University

Mehiddin Al-Baali
Department of Mathematics and Statistics,
Sultan Qaboos University

1 はじめに

本論文では、以下の無制約最適化問題:

$$\text{minimize } f(x)$$

を考える。ただし、 $f: \mathbf{R}^n \rightarrow \mathbf{R}$ は連続微分可能とし、その勾配ベクトルを $g \equiv \nabla f$ で表わす。通常、無制約最適化問題に対する数値解法として反復法が広く使われており、その反復式は

$$x_{k+1} = x_k + \alpha_k d_k \quad (1.1)$$

によって与えられる。ここで、 x_k は k 回目の近似解、 $\alpha_k > 0$ はステップ幅、 $d_k \in \mathbf{R}^n$ は探索方向である。反復法には探索方向 d_k の選び方によって多くの種類があり、よく知られた方法として、最急降下法、Newton 法、準 Newton 法、非線形共役勾配法などがある。中でも、非線形共役勾配法 (以下 CG 法と呼ぶ) は行列を保存する必要がなく、大規模問題に適しているため、近年注目を集めている。CG 法の探索方向は探索方向は

$$d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{if } k \geq 1, \end{cases} \quad (1.2)$$

によって定義される。ただし、 $g_k \equiv g(x_k)$ であり、 β_k は CG 法を特徴づけるパラメータである。CG 法はパラメータ β_k の選択法によって数値的な効率が大きく異なるため様々な β_k の選択法が提案されており、Hestenes-Stiefel(HS), Fletcher-Reeves(FR), Polak-Ribiere(PR), Dai-Yuan(DY), Conjugate Descent(CD), Liu-Storey(LS) などの方法がよく知られている:

$$\begin{aligned} \beta_k^{HS} &= \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, & \beta_k^{FR} &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, & \beta_k^{PR} &= \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \\ \beta_k^{DY} &= \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, & \beta_k^{CD} &= \frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}}, & \beta_k^{LS} &= \frac{g_k^T y_{k-1}}{-g_{k-1}^T d_{k-1}}. \end{aligned} \quad (1.3)$$

ここで, $y_{k-1} = g_k - g_{k-1}$ とし, $\|\cdot\|$ を ℓ_2 ノルムとする. 他にも多くのパラメータの選択法が提案されており, それらを用いた CG 法のアルゴリズムの大域的な収束性の議論も盛んに行われている. 詳しくは [7, 8]などを参照されたい.

通常, CG 法のアルゴリズムでは α_k を決定する際に目的関数値が下がる条件を課すのが一般的である. そのため, 探索方向における方向微係数が負 ($g_k^T d_k < 0$) であることが望ましい. これを降下条件と呼んでいる. さらに, 降下条件よりも強い条件として, ある正定数 $\bar{c}$ が存在してすべての k に対して

$$g_k^T d_k \leq -\bar{c} \|g_k\|^2$$

を満たす場合, 十分な降下条件を満たすという. 代表的な上記の β_k のうち, HS 法, PR 法, LS 法は他の三つよりも有効な方法であるといわれているが, 必ずしも降下条件を満たすとは限らない. 一方, FR 法, DY 法, CD 法は直線探索において, ある種の条件を課すことで降下方向を生成することが知られている.

近年, 探索方向 (1.2) を修正することで, 直線探索に依存せずに降下方向を生成する CG 法が盛んに研究されている. Zhang, Zhou and Li はサイジング FR 法 [12], 3 項 PR 法 [13], 3 項 HS 法 [14] を提案しており, Cheng [1] はサイジング PR 法を提案している. さらに, Narushima, Yabe and Ford [9] はこれら 4 つを含むような 3 項 CG 法の族 (以下 3TCG 法と呼ぶ) を提案している:

$$d_k = \begin{cases} -g_k & k = 0, \\ -g_k + \beta_k (g_k^T p_k)^\dagger \{(g_k^T p_k) d_{k-1} - (g_k^T d_{k-1}) p_k\} & k \geq 1. \end{cases} \quad (1.4)$$

ここで, $p_k \in \mathbf{R}^n$ は任意のパラメータベクトルであり, $\dagger$ は

$$a^\dagger = \begin{cases} \frac{1}{a} & \text{if } a \neq 0, \\ 0 & \text{if } a = 0. \end{cases}$$

で定義される一般化逆数である. ここで, 探索方向 (1.4) の左側から g_k^T をかけると $g_k^T d_k = -\|g_k\|^2$ が得られる. これは, (1.4) が $\bar{c} = 1$ で十分な降下条件を満たしていることを意味する. 3TCG 法では, 方向微係数における $\|g_k\|^2$ の係数は -1 に固定されているが, この部分のコントロールが可能な方法を考えることで, より効果的な数値解法の構築が期待できる. したがって, 今回, 我々は 3TCG 法を拡張し, $\|g_k\|^2$ の係数のコントロールが可能な 3 項共役勾配法を提案する.

2 提案法のアルゴリズム

本節では, 3TCG 法 (1.4) を拡張して, 以下の 3 項共役勾配法 (以下, G3TCG 法と呼ぶ) を提案する:

$$d_k = \begin{cases} -g_k & \text{if } k = 0 \text{ or } |g_k^T p_k| \leq \theta \|g_k\| \|p_k\|, \\ -g_k + \beta_k d_{k-1} + \eta_k p_k & \text{otherwise.} \end{cases} \quad (2.1)$$

ここで, $\theta \in (0, 1)$, β_k をパラメータ, $p_k \in \mathbf{R}^n$ を非ゼロベクトルとし, さらに

$$\eta_k = -\frac{(\gamma_k - 1)\|g_k\|^2 + \beta_k g_k^T d_{k-1}}{g_k^T p_k}$$

とする. ただし, γ_k は $\bar{\gamma}_1 \leq \gamma_k \leq \bar{\gamma}_2$ ($0 < \bar{\gamma}_1 \leq 1 \leq \bar{\gamma}_2$) を満たすパラメータとする. G3TCG 法の探索方向 (2.1) は, $\gamma_k = 1$ となると 3TCG 法の探索方向 (1.4) に帰着することを注意しておく. ここで, (2.1) より

$$g_k^T d_k = -\gamma_k \|g_k\|^2 \quad (2.2)$$

を得ることができる. したがって, G3TCG 法はパラメータ γ_k を調節することで方向微係数 (2.2) をコントロールすることが可能である. 一方, (2.1) の 2 式目の場合 (つまり, $d_k = -g_k$ ではない場合), 探索方向は

$$d_k = \left(I - \frac{p_k g_k^T}{g_k^T p_k} \right) (-g_k + \beta_k d_{k-1}) + \frac{p_k g_k^T}{g_k^T p_k} (-\gamma_k g_k)$$

と表すことができる. ここで, $I - p_k g_k^T / g_k^T p_k$ は $\text{Span}\{p_k\}$ に沿って $\text{Span}\{g_k\}$ の直交補空間へ射影する行列であり, $p_k g_k^T / g_k^T p_k$ は $\text{Span}\{g_k\}$ の直交補空間に沿って $\text{Span}\{p_k\}$ へ射影する行列である. したがって, 探索方向 (2.1) のイメージは図 1 のようになる.

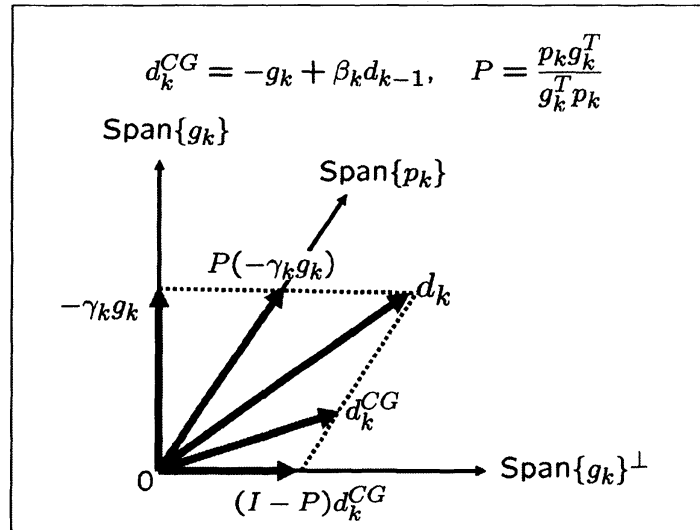


図 1: 探索方向 (2.1) のイメージ図

ここで, G3TCG 法のアルゴリズムを以下のように与える.

アルゴリズム G3TCG

Step 0 初期点 x_0 とパラメータ $\theta \in (0, 1)$, $0 < \delta < \sigma_1 < 1$, $\sigma_2 > 0$ を与える. $k := 0$ とする.

Step 1 停止判定条件を満たすならば, アルゴリズムを停止して x_k を解とする.

Step 2 探索方向 d_k を (2.1) によって計算する.

Step 3 直線探索により一般化強 Wolfe 条件:

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq -\delta \alpha_k g_k^T d_k, \\ -\sigma_2 g_k^T d_k &\geq g(x_k + \alpha_k d_k)^T d_k \geq \sigma_1 g_k^T d_k \end{aligned}$$

を満たすステップ幅 $\alpha_k > 0$ を計算する.

Step 4 点 x_{k+1} を (1.1) により更新する.

Step 5 $k := k + 1$ として Step 1 へ戻る.

3 大域的収束性

本節では、前節で提案したアルゴリズム G3TCG の大域的収束性について議論する. そのために、目的関数 f に対して以下を仮定する.

仮定 1 初期点における準位集合 $\mathcal{L} = \{x | f(x) \leq f(x_0)\}$ は有界であり、その近傍 $\mathcal{N}$ において f は連続微分可能で、かつ、 g はリプシッツ連続である.

ここで、アルゴリズム G3TCG の大域的収束性を議論するために、パラメータ β_k に関する以下の性質を考える.

Property A アルゴリズム G3TCG を考える. すべての k に対して、 $\|g_k\| \geq \varepsilon$ となる正定数 ε が存在すると仮定する. このとき、定数 $b > 1$ と $\xi > 0$ が存在して、すべての k に対して $|\beta_k| \leq b$ と

$$\|s_{k-1}\| \leq \xi \implies |\beta_k| \leq \frac{1}{4\mu^4 b}$$

が成り立つとき、アルゴリズム G3TCG は Property A を持つという. ただし、 $s_{k-1} = x_k - x_{k-1}$, $\mu = (1 + \bar{\gamma}_2)/\theta$ とする.

この Property A は通常の CG 法における Property (*) に対応している. Property (*) の詳細は、例えばサーベイ論文 [7] などを参照されたい.

アルゴリズム G3TCG の大域的な収束性を議論する際、Property A に加えて、パラメータ β_k に対して下記の条件が要求される:

$$\beta_k \geq \frac{-1}{\|d_{k-1}\| \min\{\bar{\nu}, \|g_{k-1}\|\}} \equiv \nu_k. \quad (3.1)$$

ただし、 $\bar{\nu}$ は正の定数とする. したがって、以降では (3.1) を満たすような β_k を考えることとする. なお、一般的には、この条件は成り立たないが、任意の β_k に対して、

$$\beta_k^+ = \max\{\zeta_k, \beta_k\} \quad (\text{ただし、}\zeta_k \in [\nu_k, 0]) \quad (3.2)$$

と補正を行うことで (3.1) が満たされる. また、元の β_k を用いたアルゴリズム G3TCG が Property A を持つならば、補正した β_k^+ を用いたものも Property A を持つことを注意しておく. ここで、アルゴリズム G3TCG の大域的収束性を証明するのに必要な二つの補題を与える.

補題 1 仮定 1 が成立しているとし, $\{x_k\}$ をアルゴリズム G3TCG によって生成される無限点列とする. アルゴリズムは Property A を持ち, さらに, すべての k に対し, (3.1) を満たしているとする. さらに, 正の定数 ε が存在し, すべての k に対し, $\varepsilon \leq \|g_k\|$ を満たしていると仮定する. このとき, 探索方向 d_k は以下の関係を満たす:

$$\sum_{k=1}^{\infty} \|u_k - u_{k-1}\|^2 < \infty,$$

ただし, $u_k = d_k / \|d_k\|$ とする.

ここで, 自然数の集合を $\mathbf{N}$ とし, 正の定数 λ と自然数 Δ に対して, 以下の添え字集合を定義する:

$$\mathcal{K}_{k,\Delta}^\lambda := \{i \in \mathbf{N} \mid k \leq i \leq k + \Delta - 1, \|s_{i-1}\| > \lambda\}.$$

さらに, $|\mathcal{K}_{k,\Delta}^\lambda|$ は集合 $\mathcal{K}_{k,\Delta}^\lambda$ の要素数を表すものとする.

補題 2 補題 1 のすべての仮定が成り立っているものとする. このとき, 正定数 λ が存在して, すべての $\Delta \in \mathbf{N}$ と k_0 に対して,

$$|\mathcal{K}_{k,\Delta}^\lambda| > \frac{\Delta}{2}$$

が成立するような k ($k \geq k_0$) が存在する.

補題 1, 2 を用いることで, 以下の大域的収束性が得られる.

定理 1 仮定 1 が成立しているとし, $\{x_k\}$ をアルゴリズム G3TCG によって生成される無限点列とする. このとき, アルゴリズムが Property A を持ち, さらに, すべての k に対し, (3.1) を満たすならば, 点列 $\{x_k\}$ は

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

の意味で大域的に収束する.

次に, β_k を具体的に与えた場合の G3TCG 法の大域的収束性について議論する. 1 節で述べたとおり, 様々な β_k の選択法が提案されているが, 今回, 我々は 1 節で紹介した選択法 ((1.3) を参照) のうち, HS, PR, LS, さらに, 近年の研究で提案されている 4 つの選択法: Dai-Liao (DL) [2], Hager-Zhang (HZ) [5], Yu-Guan-Li (DPR) [10], および Zhang (DLS) [11] を考える. DL, HZ, DPR, DLS の選択法はそれぞれ以下のようにになっている:

$$\beta_k^{DL} = \frac{g_k^T(y_{k-1} - ts_{k-1})}{d_{k-1}^T y_{k-1}}, \quad (3.3)$$

$$\beta_k^{HZ} = \beta_k^{HS} - \frac{\phi \|y_k\|^2}{(d_{k-1}^T y_{k-1})^2} g_k^T d_{k-1}, \quad (3.4)$$

$$\beta_k^{DPR} = \beta_k^{PR} - \frac{\phi \|y_k\|^2}{\|g_{k-1}\|^4} g_k^T d_{k-1}, \quad (3.5)$$

$$\beta_k^{DLS} = \beta_k^{LS} - \frac{\phi \|y_k\|^2}{(-g_{k-1}^T d_{k-1})^2} g_k^T d_{k-1}. \quad (3.6)$$

ただし, $t \geq 0$, $\phi > 1/4$ とする. 定理 1 を用いることで, これら 7 つの具体的な方法に対する大域的収束性が得られる.

定理 2 $\{x_k\}$ を (3.2) の補正を用いたアルゴリズム G3TCG によって生成される無限点列とする. このとき, (3.2) の β_k として β_k^{HS} , β_k^{PR} , β_k^{LS} , β_k^{DL} , β_k^{HZ} , β_k^{DPR} , または β_k^{DLS} を選択したならば, 点列 $\{x_k\}$ は

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0$$

の意味で大域的に収束する.

4 数値実験

本節では (3.2) の補正を用いたアルゴリズム G3TCG の数値実験結果を報告する. テスト問題として, CUTEr 問題集 [4] から 167 問を選んで実験を行った. 表 1 ではテスト問題の名前とその次元を挙げている.

表 1: テスト問題

Code	n	Code	n	Code	n	Code	n
3PK	30	DENSCHNA	2	GROWTHLS	3	PALMER5C	6
AIRCRFTB	8	DENSCHNB	2	GULF	3	PALMER6C	8
AKIVA	2	DENSCHNC	2	HAIRY	2	PALMER7C	8
ALLINIT	4	DENSCHND	3	HATFLDD	3	PALMER8C	8
ALLINITU	4	DENSCHNE	3	HATFLDE	3	PENALTY1	1000
ARGLINA	200	DENSCHNF	2	HATFLDFL	3	PENALTY1	10000
ARGLINB	200	DIXMAANA	9000	HEART6LS	6	PENALTY2	200
ARGLINC	200	DIXMAANB	9000	HEART8LS	8	PENALTY3	100
ARWHEAD	5000	DIXMAANC	9000	HELIX	3	PFIT1LS	3
BARD	3	DIXMAAND	9000	HIELOW	3	PFIT2LS	3
BDEXP	5000	DIXMAANE	9000	HILBERTA	100	PFIT3LS	3
BDQRTIC	5000	DIXMAANF	9000	HILBERTB	100	PFIT4LS	3
BEALE	2	DIXMAANG	9000	HIMMELBB	2	POWELLSG	20000
BIGGS3	6	DIXMAANH	9000	HIMMELBF	4	POWER	20000
BIGGS5	6	DIXMAANI	9000	HIMMELBG	2	QUARTC	10000
BIGGS6	6	DIXMAANJ	9000	HIMMELBH	2	ROSENBR	2
BIGGSB1	5000	DIXMAANK	3000	HUMPS	2	S308	2
BOX	7500	DIXMAANL	9000	JENSMP	2	SBRYBND	5000
BOX2	3	DIXON3DQ	10000	KOWOSB	4	SCHMVETT	5000
BOX3	3	DJTL	2	LIARWHD	10000	SENSORS	100
BQPGABIM	50	DQDRTIC	5000	LOGHAIRY	2	SINEVAL	2
BQPGASIM	50	DQRTIC	5000	MANCINO	100	SINQUAD	10000
BRKMCC	2	EDENSCH	10000	MARATOSB	2	SISSER	2
BROWNAL	500	EG2	1000	MEXHAT	2	SPARSINE	5000
BROWNBS	2	EIGENALS	930	MODBEALE	10000	SPARSQUR	10000
BROWNDEN	4	EIGENBLS	930	MOREBV	5000	SROSENBR	10000
BROYDN7D	5000	EIGENCLS	462	MOREBV	10000	STRATEC	10
BROYDN7D	10000	ENGVAL1	10000	MSQRTALS	1024	TESTQUAD	5000
BRYBND	10000	ENGVAL2	3	MSQRTBLS	1024	TOINTGSS	10000
CAMEL6	2	ERRINROS	50	NONCVXU2	5000	TOINTPSP	50
CHAINWOO	4000	EXPFIT	2	NONDIA	10000	TOINTQOR	50
CHAINWOO	10000	EXTROSNB	1000	NONDQUAR	5000	TQUARTIC	10000
CHEBYQAD	100	EXTROSNB	10000	NONDQUAR	10000	TRIDIA	10000
CHNROSNB	50	FLETCHCR	1000	NONSCOMP	5000	VARDIM	200
CLIFF	2	FLETCHCR	10000	OSBORNEA	5	VAREIGVL	5000
COSINE	10000	FMINSRF2	5625	OSBORNEB	11	VIBRBEAM	8
CRAGGLVY	5000	FMINSURF	5625	OSCPATH	10000	WATSON	31
CUBE	2	FREUROTH	5000	PALMER1C	8	WOODS	4000
CURLY10	10000	GENHUMPS	5000	PALMER1D	7	WOODS	10000
CURLY20	10000	GENROSE	500	PALMER2C	8	YFITU	3
CURLY30	5000	GENROSE	5000	PALMER3C	8	ZANGWIL2	2
DECONVU	61	GENROSE	10000	PALMER4C	8		

また, 表 2 では実験を行った方法を紹介している.

表 2: 実験を行った方法

CG-DESCENT	Hager and Zhang [5, 6] による CG 法のソフトウェア
GHS	$\beta_k = \beta_k^{HS}$ としたアルゴリズム G3TCG
GPR	$\beta_k = \beta_k^{PR}$ としたアルゴリズム G3TCG
GLS	$\beta_k = \beta_k^{LS}$ としたアルゴリズム G3TCG
GDL	$\beta_k = \beta_k^{DL}$ としたアルゴリズム G3TCG
GHZ	$\beta_k = \beta_k^{HZ}$ としたアルゴリズム G3TCG
GDPR	$\beta_k = \beta_k^{DPR}$ としたアルゴリズム G3TCG

アルゴリズム G3TCG において, 探索方向 (2.1) には任意のベクトル p_k やパラメータ γ_k が含まれている. 今回, $p_k = g_k$ または $p_k = y_{k-1}$ を選択した. 各方法において, もし $p_k = g_k$ とした場合には名称の後ろに “1” を, また, $p_k = y_{k-1}$ とした場合には名称の後ろに “2” を付すこととする. 例えば, $\beta_k = \beta_k^{HS}$, $p_k = g_k$ を用いた G3TCG 法は GHS1 法と呼ぶ. また, γ_k は, 条件 $\bar{\gamma}_1 \leq \gamma_k \leq \bar{\gamma}_2$ を満たすように,

$$\gamma_k = \max \{ \bar{\gamma}_1, \min \{ \bar{\gamma}_2, \hat{\gamma}_k \} \}$$

とした. ただし, $\hat{\gamma}_k$ はパラメータとする. 通常, CG 法では正確な直線探索 (つまり, $g_k^T d_{k-1} = 0$) が行われた場合には, $g_k^T d_k = -\|g_k\|^2$ が成立する. よって, $g_k^T d_{k-1} \rightarrow 0$ の場合には $\gamma_k \rightarrow 1$ となるような $\hat{\gamma}_k$ を選択することが自然である. このことから, 今回, 以下の $\hat{\gamma}_k$ について実験を行った:

$$\begin{aligned} \hat{\gamma}_k^{(1)} &= 1 - \bar{\gamma} \frac{|\beta_k g_k^T d_{k-1}|}{\|g_k\| \|d_{k-1}\|}, & \hat{\gamma}_k^{(2)} &= 1 + \bar{\gamma} \frac{|\beta_k g_k^T d_{k-1}|}{\|g_k\| \|d_{k-1}\|}, \\ \hat{\gamma}_k^{(3)} &= 1 - \bar{\gamma} \frac{\beta_k g_k^T d_{k-1}}{\|g_k\| \|d_{k-1}\|}, & \hat{\gamma}_k^{(4)} &= 1 + \bar{\gamma} \frac{\beta_k g_k^T d_{k-1}}{\|g_k\| \|d_{k-1}\|}, \\ \hat{\gamma}_k^{(5)} &= 1 - \bar{\gamma} |\beta_k g_k^T d_{k-1}|, & \hat{\gamma}_k^{(6)} &= 1 + \bar{\gamma} |\beta_k g_k^T d_{k-1}|, \\ \hat{\gamma}_k^{(7)} &= 1 - \bar{\gamma} \beta_k g_k^T d_{k-1}, & \hat{\gamma}_k^{(8)} &= 1 + \bar{\gamma} \beta_k g_k^T d_{k-1}, \\ \hat{\gamma}_k^{(9)} &= 1 - \bar{\gamma} \frac{|g_k^T d_{k-1}|}{\|g_k\| \|d_{k-1}\|}, & \hat{\gamma}_k^{(10)} &= 1 + \bar{\gamma} \frac{|g_k^T d_{k-1}|}{\|g_k\| \|d_{k-1}\|}, \\ \hat{\gamma}_k^{(11)} &= 1 - \bar{\gamma} \frac{g_k^T d_{k-1}}{\|g_k\| \|d_{k-1}\|}, & \hat{\gamma}_k^{(12)} &= 1 + \bar{\gamma} \frac{g_k^T d_{k-1}}{\|g_k\| \|d_{k-1}\|}, \\ \hat{\gamma}_k^{(13)} &= 1 - \bar{\gamma} |g_k^T d_{k-1}|, & \hat{\gamma}_k^{(14)} &= 1 + \bar{\gamma} |g_k^T d_{k-1}|, \\ \hat{\gamma}_k^{(15)} &= 1 - \bar{\gamma} g_k^T d_{k-1}, & \hat{\gamma}_k^{(16)} &= 1 + \bar{\gamma} g_k^T d_{k-1}. \end{aligned}$$

ただし, $\bar{\gamma}$ は非負の定数とする. また, 今回提案したアルゴリズムの比較対象として Hager-Zhang による CG 法のソフトウェアである CG-DESCENT [5, 6] の数値実験も行った. なお我々の方法のコードは CG-DESCENT を修正し作成しており, 実験したすべての方法に

において、直線探索等の設定は CG-DESCENT のデフォルトの設定を用いている。その他のパラメータは $\theta = 10^{-12}$, $\zeta_k = \nu_k$, $\bar{\nu} = 0.01$, $\bar{\gamma}_1 = 0.1$, $\bar{\gamma}_2 = 100$, $\bar{\gamma} = 0.8$, $t = 1$, $\phi = 2$ とした。収束判定条件は

$$\|g_k\|_{\infty} \leq 10^{-6}$$

を使用している。また、実行時間が 500(秒) を超えた場合もアルゴリズムを停止している。

今回、我々は、各方法の CPU 時間を比較するために、Dolan and Moré [3] の提案したパフォーマンスプロファイルを用いた。各方法のパフォーマンスプロファイル $P(\tau)$ の $\tau = \bar{\tau}$ のときの値は、その解法がすべての問題の中で、最も早く解くことができた方法の求解時間の $\bar{\tau}$ 倍以内に解くことのできた問題の割合を表している。 $\tau = 1$ のときの値は、その方法がすべて方法の中で、最も早く解くことができた問題の割合を表しており、一方、 τ が十分大きい時は、解くことのできた問題の割合を表すこととなる。どの τ においても、1 に近いほうが好ましく、複数の数値解法を比較する場合、パフォーマンスプロファイルが上に位置するほど効率が良いと考えることができる。

まず最初に、 γ_k の選択によって計算効率がどう変化するかを調べるために、上で提案した 16 種類の γ_k に対する比較を行った。実験を行った結果、 $\hat{\gamma}_k$ の 2 項目が $\|g_k\| \|d_{k-1}\|$ で割られていないもの (つまり 5-8, 13-16 番目の $\hat{\gamma}_k$) を用いた場合には効率が良くないことが観測できた。よって、それ以外の結果を紹介することとする。図 2-5 では、それぞれ GHS1, GHS2, GPR1, GPR2 において $\hat{\gamma}_k$ を変化させて実験を行ったときのパフォーマンスプロファイルである。図 2 から、GHS1 に対しては、 $\gamma_k = 1$ の場合と比較すると、 $\hat{\gamma}_k^{(1)}$, $\hat{\gamma}_k^{(10)}$, $\hat{\gamma}_k^{(11)}$ が効果的であることが見て取れる。同様に図 3-5 から、GHS2 に対しては $\hat{\gamma}_k^{(1)}$ と $\hat{\gamma}_k^{(11)}$, GPR1 に対しては $\hat{\gamma}_k^{(1)}$, $\hat{\gamma}_k^{(3)}$, $\hat{\gamma}_k^{(4)}$, $\hat{\gamma}_k^{(10)}$, GHS2 に対しては $\hat{\gamma}_k^{(10)}$ が効果的であることが分かる。

次に、表 2 の方法間の比較を行った結果を報告する。この比較においては、各方法で $\hat{\gamma}_k^{(11)}$ を使用し実験を行った。図 6 では $p_k = g_k$ とした場合の各方法の比較を行い、図 7 では $p_k = y_{k-1}$ とした場合の各方法の比較を行っている。図 6 と 7 から GHS1, GHZ1, GDL1, GHS2, GPR2, GDPR2 が効率的であることが見て取れる。さらに、図 8 では、図 6 と 7 で効率的だった GHS1, GHZ1, GDL1, GHS2, GPR2, GDPR2 と CG-DESCENT を比較している。この図から、GDPR2 が最も効果的であることが分かる。

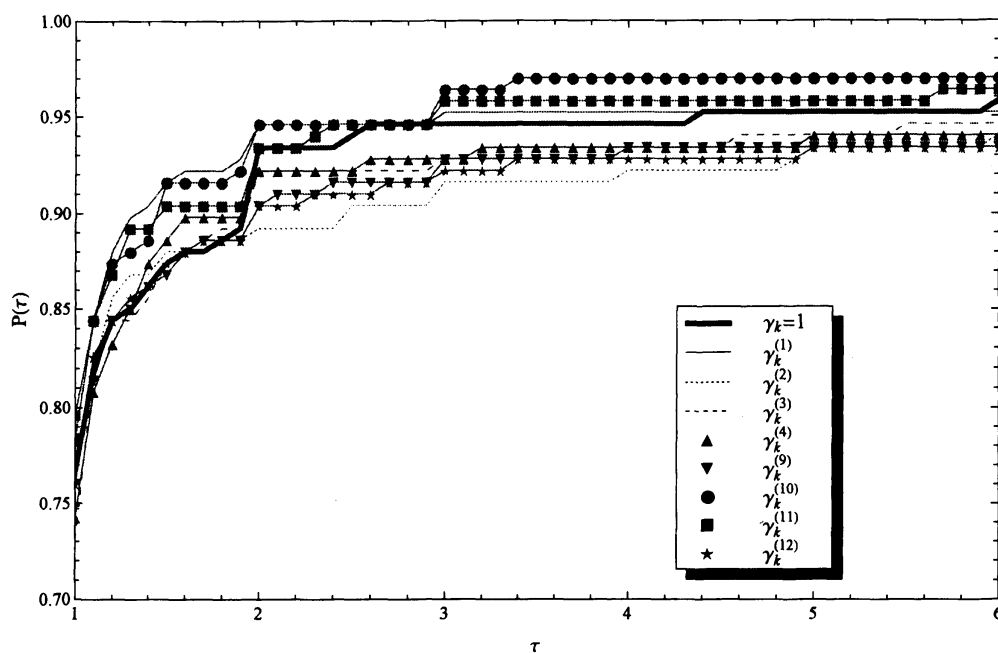


図 2: GHS1 において γ_k を変化させた場合の比較

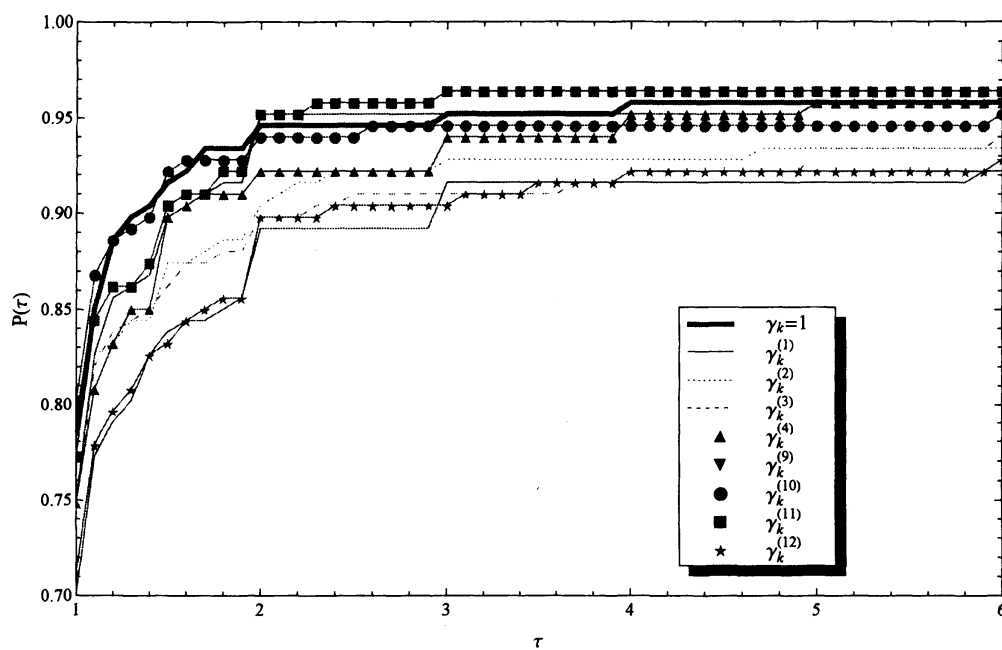


図 3: GHS2 において γ_k を変化させた場合の比較

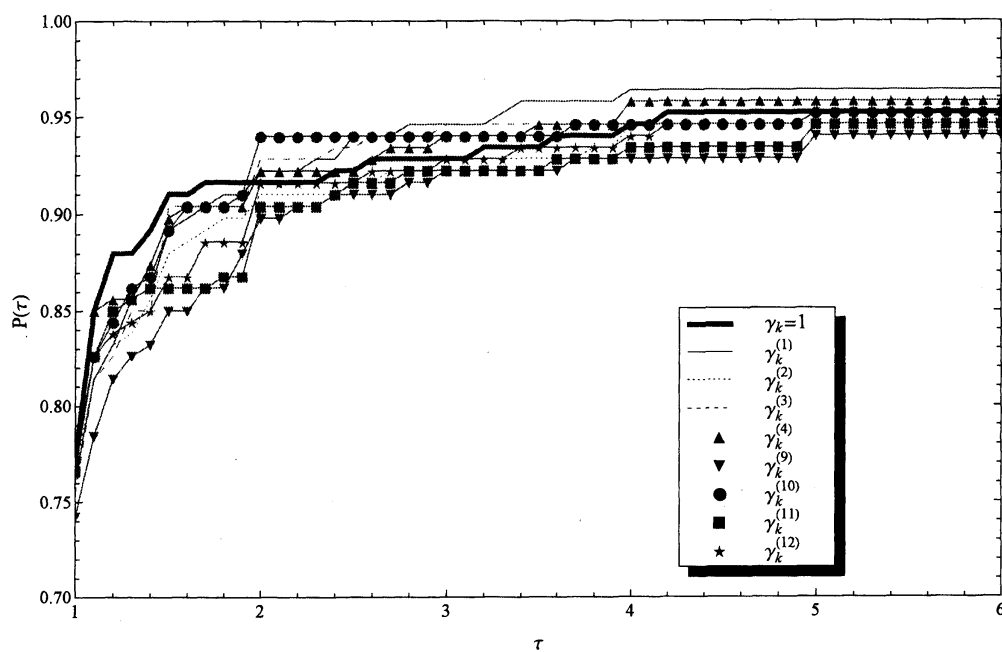


図 4: GPR1 において γ_k を変化させた場合の比較

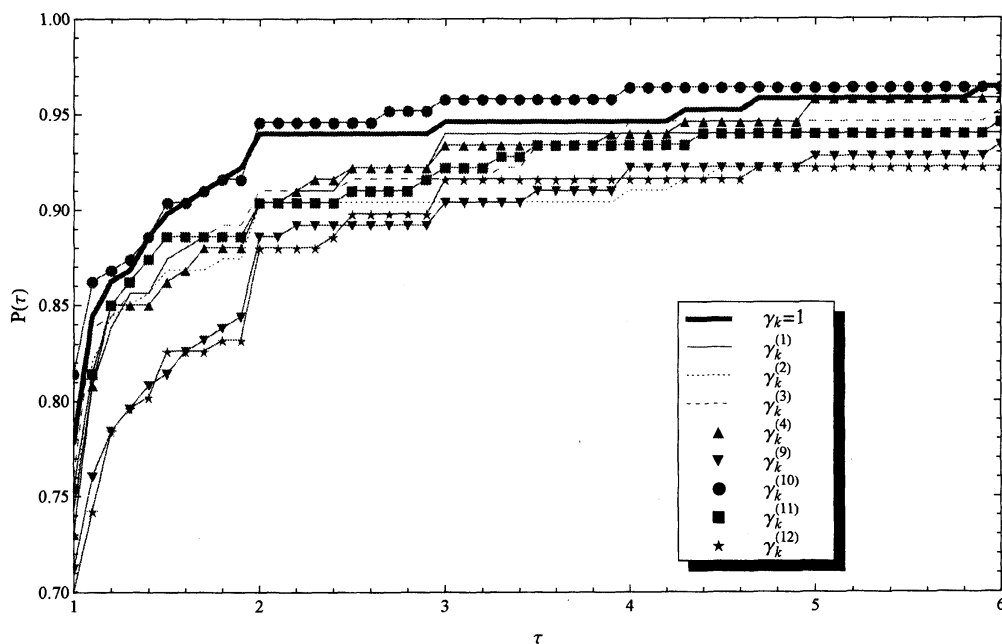


図 5: GPR2 において γ_k を変化させた場合の比較

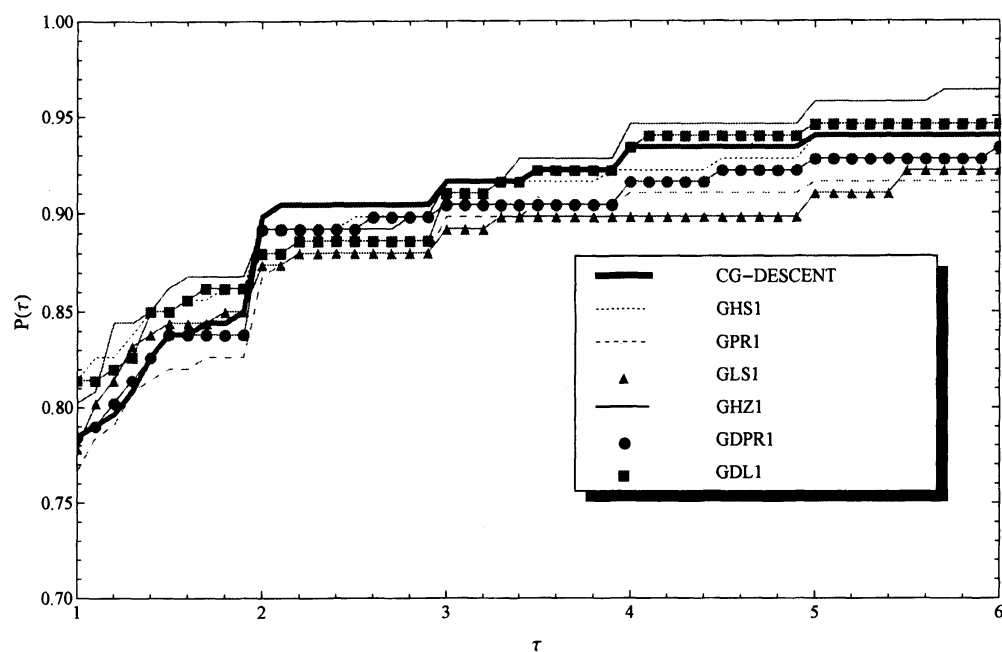


図 6: $p_k = g_k$ とした場合の各方法の比較

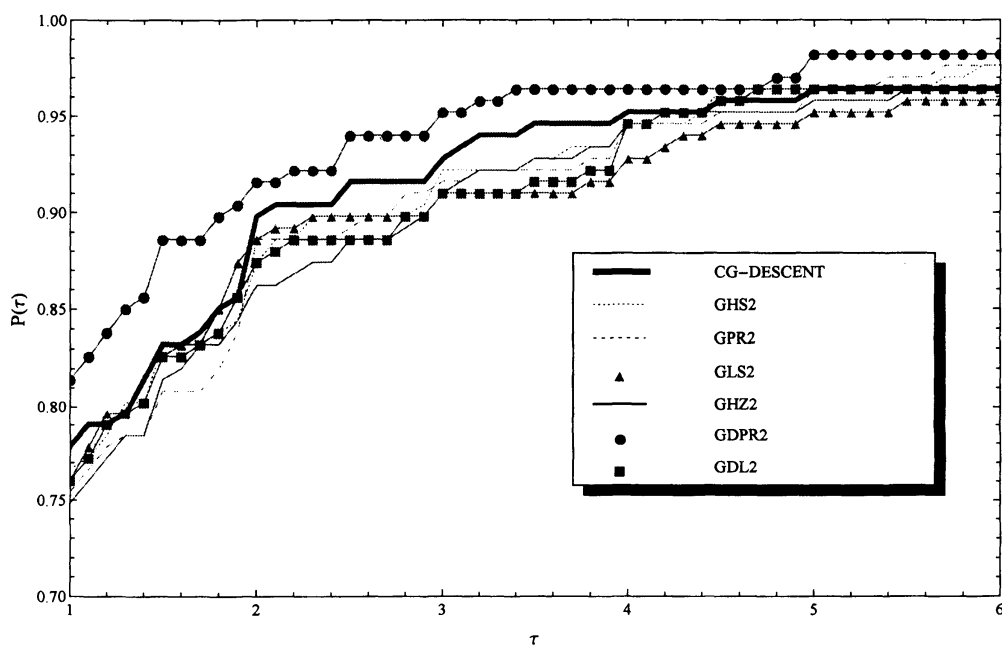


図 7: $p_k = y_{k-1}$ とした場合の各方法の比較

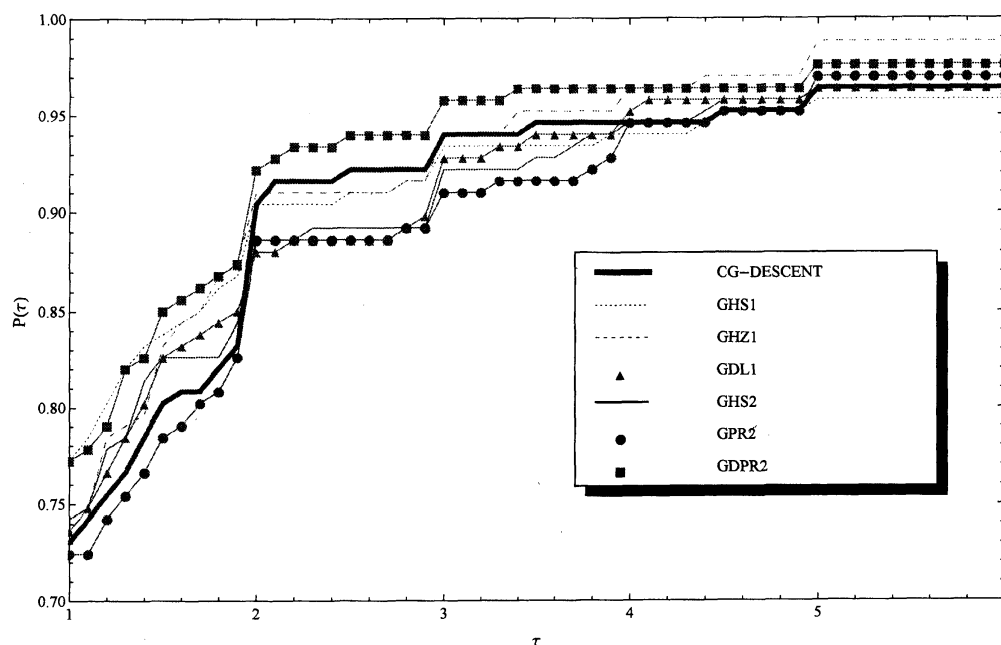


図 8: 図 6 と 7 で効果的であった方法の比較

5 終わりに

本論文では、Narushima らの非線形 3 項共役勾配法を拡張し、方向微係数の大きさを調節できる非線形 3 項共役勾配法を提案し、その大域的収束性を示した。また、数値実験を行って提案法の有効性を検証した。数値実験から、提案法はパラメータの選択によっては既存の手法よりも優れていることを確認した。今後の課題としては、パラメータ γ_k に対する、より有効な選択法の構築があげられる。

謝辞

本研究において、第一および第二著者は日本学術振興会科学研究費補助金基盤研究 (C) (25330030) の支援を受けている。

参考文献

- [1] W. Cheng, A two-term PRP-based descent method, *Numerical Functional Analysis and Optimization*, **28** (2007), 1217–1230.
- [2] Y.H. Dai and L.Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Applied Mathematics and Optimization*, **43** (2001), 87–101.
- [3] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, **91** (2002), 201–213.

- [4] N.I.M. Gould, D. Orban and P.L. Toint, CUTeR and SifDec: A constrained and unconstrained testing environment, revisited, *ACM Transactions on Mathematical Software*, **29** (2003), 373–394.
- [5] W.W. Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM Journal on Optimization*, **16** (2005), 170–192.
- [6] W.W. Hager and H. Zhang, Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent, *ACM Transactions on Mathematical Software*, **32** (2006), 113–137.
- [7] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific Journal of Optimization*, **2** (2006), 35–58.
- [8] 成島康史, 大規模無制約最適化問題に対する最近の研究動向, 応用数理, **22** (2012), 27–39.
- [9] Y. Narushima, H. Yabe and J.A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, *SIAM Journal on Optimization*, **21** (2011), 212–230.
- [10] G. Yu, L. Guan, and G. Li, Global convergence of modified Polak-Ribière-Polyak conjugate gradient methods with sufficient descent property, *Journal of Industrial and Management Optimization*, **4** (2008), 565–579.
- [11] L. Zhang, A new Liu-Storey type nonlinear conjugate gradient method for unconstrained optimization problems, *Journal of Computational and Applied Mathematics*, **225** (2009), 146–157.
- [12] L. Zhang, W. Zhou and D.H. Li, Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, *Numerische Mathematik*, **104** (2006), 561–572.
- [13] L. Zhang, W. Zhou and D.H. Li, A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence, *IMA Journal of Numerical Analysis*, **26** (2006), 629–640.
- [14] L. Zhang, W. Zhou and D.H. Li, Some descent three-term conjugate gradient methods and their global convergence, *Optimization Methods and Software*, **22** (2007), 697–711.