

# 階層制約付き凸最適化問題に関する反復アルゴリズムと ネットワーク帯域幅割り当てへの応用

飯塚 秀明\*

## 1 はじめに

### 1.1 ネットワーク帯域幅割り当て問題

ネットワーク帯域幅をネットワーク送信者に効率的, かつ, 公平に割り当てる仕組みを実現することは, 通信ネットワークの設計・制御・管理・運用における中核となる課題の一つである [10, 11, 12]. ネットワーク送信者への資源割り当てに関する効率性や公平性がある指標関数 (「満足度関数」と呼ぶ) で表わされるとき, これを実行可能な物理制約である「リンク容量制約」の下で最大化することがネットワーク帯域幅割り当ての目的となる. 送信者  $s$  の満足度関数  $u_s$  は送信者  $s$  に割り当てられた送信レート  $x_s$  の連続微分可能な凹関数として表現される. 本論文では, 以下のように定義される proportional fair 関数 [10, 11, 12] と呼ばれる満足度関数を扱う: 任意の  $s \in \mathcal{S} := \{1, 2, \dots, S\}$  に対して,

$$u_s(x) := \log x \quad (x \in \mathbb{R}_+ \setminus \{0\}).$$

ただし,  $\mathcal{S} := \{1, 2, \dots, S\}$  は全送信者の集合を表す. ネットワーク全体の満足度関数は, 以下で定義される:

$$U(\mathbf{x}) := \sum_{s \in \mathcal{S}} u_s(x_s) \quad (\mathbf{x} := (x_s)_{s \in \mathcal{S}} \in \mathbb{R}_+^S \setminus \{0\}). \quad (1)$$

各リンク  $l \in \mathcal{L} := \{1, 2, \dots, L\}$  に対するリンク容量制約とは, リンク  $l$  を共有する全送信者の送信レートの和がリンク  $l$  の容量  $c_l$  以下になる制約のことである. すなわち, リンク容量制約集合  $C$  は以下の集合で表現される:

$$C := \mathbb{R}_+^S \cap \bigcap_{l \in \mathcal{L}} C_l \neq \emptyset, \quad (2)$$
$$C_l := \left\{ \mathbf{x} := (x_s)_{s \in \mathcal{S}} \in \mathbb{R}_+^S : \sum_{s \in \mathcal{S}} x_s I_{s,l} \leq c_l \right\} \quad (l \in \mathcal{L}).$$

\*九州工業大学ネットワークデザイン研究センター 〒 100-0011 東京都千代田区内幸町 2-2-3 日比谷国際ビル 1F 107 区, E-mail: iiduka@ndrc.kyutech.ac.jp

ただし,  $I_{s,l}$  ( $s \in \mathcal{S}, l \in \mathcal{L}$ ) は, 送信者  $s$  がリンク  $l$  を使用しているとき,  $I_{s,l} = 1$ , それ以外のときは,  $I_{s,l} = 0$  をとる関数とする. 式 (1), (2) と帯域幅割り当ての目的により, ネットワーク帯域幅割り当て問題 [10, 11, 12] を以下のように定式化することができる:

**問題 1.1.**  $C$  上で  $U$  を最大化せよ.

本論文では, リンク容量制約だけでなく, 送信者の送信レートに関する要求にも配慮したネットワーク帯域幅割り当て問題 [6, 2, 7, 5] を考察する. 送信者  $s$  は, ある  $r_s$  ( $> 0$ ) 以上の送信レートでデータを送りたいといった要求をもつものとする. このとき, 望ましい送信レートに関する制約集合  $D_s$  ( $s \in \mathcal{S}$ ) は以下のように表すことができる:

$$D_s := \{(x_s)_{s \in \mathcal{S}} \in \mathbb{R}^{\mathcal{S}} : x_s \geq r_s\} \quad (s \in \mathcal{S}).$$

よって,  $C \cap \bigcap_{s \in \mathcal{S}} D_s$  上での  $U$  の最大点を見つけることが最も望ましい. しかしながら, リンク容量よりも大きいレートを要求する送信者  $s_0$  がネットワーク上に存在した場合,  $C \cap D_{s_0}$  が空になるので, 一般には,

$$C \cap \bigcap_{s \in \mathcal{S}} D_s = \emptyset$$

となる. 実行不可能な状況を回避するために, ネットワーク帯域幅割り当て問題の制約集合を再定義する必要がある. 集合  $C$  はネットワーク帯域幅割り当て問題の絶対集合なので,  $C \cap \bigcap_{s \in \mathcal{S}} D_s$  の代わりに,

平均 2 乗ノルムの意味で  $D_s$  に最も近い要素から成る  $C$  の部分集合

をネットワーク帯域幅割り当て問題の制約集合として扱うことは妥当であろうと考えられる. この集合は,  $\mathbf{x} \in \mathbb{R}^{\mathcal{S}}$  から  $D_s$  への距離の平均 2 乗の値  $\Phi(\mathbf{x})$ , すなわち,  $\sum_{s \in \mathcal{S}} w_s = 1$  を満たす  $w_s \in (0, 1)$  ( $s \in \mathcal{S}$ ) に対して,

$$\Phi(\mathbf{x}) := \frac{1}{2} \sum_{s \in \mathcal{S}} w_s d(\mathbf{x}, D_s)^2 = \frac{1}{2} \sum_{s \in \mathcal{S}} w_s \left( \min_{\mathbf{y} \in D_s} \|\mathbf{x} - \mathbf{y}\| \right)^2 \quad (\mathbf{x} \in \mathbb{R}^{\mathcal{S}})$$

を用いて, 以下のように定義することができる:

$$C_\Phi := \left\{ \mathbf{x}^* \in C : \Phi(\mathbf{x}^*) = \min_{\mathbf{y} \in C} \Phi(\mathbf{y}) \right\} \neq \emptyset.$$

$C_\Phi$  は  $C \cap \bigcap_{s \in \mathcal{S}} D_s = \emptyset$  でさえも, 定義することができ,  $C$  のコンパクト性と  $\Phi$  の連続性により,  $C_\Phi \neq \emptyset$  が常に成立する. さらに,  $C \cap \bigcap_{s \in \mathcal{S}} D_s \neq \emptyset$  のとき,  $C_\Phi = C \cap \bigcap_{s \in \mathcal{S}} D_s$  が成り立つことから,  $C_\Phi$  は  $C \cap \bigcap_{s \in \mathcal{S}} D_s$  の一般化にもなっていることがわかる. この  $C_\Phi$  は, 一般化凸実行可能集合 [1, 13] と呼ばれており, 信号処理 [1], 電力割り当て [3], 最適制御 [9] にも応用されている集合である.

以上の議論により, 本論文では, 以下のネットワーク帯域幅割り当て問題 [6, 2, 7, 5] を考察する:

**問題 1.2** (ネットワーク帯域幅割り当て問題).

$C_\Phi$  上で  $U$  を最大化せよ.

## 1.2 既存の帯域幅割り当てアルゴリズムとそれらの問題点

文献 [6, 2] では, 問題 1.2 の導入と問題 1.2 を解くためのアルゴリズムを与えている<sup>1</sup>. [6, 2] で提案されたアルゴリズムは, 常に絶対集合  $C$  に収束する保証はなく, アルゴリズムが  $C$  に収束するようあるパラメータを設定する必要がある. このようなパラメータ設定は, 実装の観点から望ましいとは言えず, パラメータ設定の必要のない帯域幅割り当てアルゴリズムを開発することが望ましい.

パラメータ設定の必要性が生じた原因は, 問題 1.2 を 2 重階層制約付き凸最適化問題 ( $C_\Phi$  上で  $u$  を最大化する問題) として扱ったためである (詳細については, [6, Section II], [2, Subsection 1.2] を参照せよ). 本論文では, 問題 1.2 を 3 重階層制約付き凸最適化問題 ( $C$  上での  $\Phi$  の最小点の集合上で  $u$  を最大化する問題) として定式化し, その問題を解くためのアルゴリズムとその収束解析 [4, 5] を与える. 本提案アルゴリズムは, パラメータ設定の必要はなく, 常に絶対集合  $C$  に収束することが保証される.

## 2 3 重階層制約付き凸最適化問題に関するアルゴリズム

### 2.1 3 重階層制約付き凸最適化問題

$H$  を内積  $\langle \cdot, \cdot \rangle$ , ノルム  $\|\cdot\|$  をもつヒルベルト空間とする. この節では, 次の最適化問題を考察する:

**問題 2.1** (3 重階層制約付き凸最適化問題).

- (I)  $T: H \rightarrow H$  は  $\text{Fix}(T) := \{x \in H: T(x) = x\} \neq \emptyset$  を満たす非拡大写像<sup>2</sup>,
- (II)  $f_1: H \rightarrow H$  は連続フレッシュェ微分可能な凸関数とし,  $\nabla f_1: H \rightarrow H$  は  $L_1$ -リプシッツ連続<sup>3</sup>,

$$\begin{aligned} & \text{VI}(\text{Fix}(T), \nabla f_1) \\ & := \{x^* \in \text{Fix}(T): \langle x - x^*, \nabla f_1(x^*) \rangle \geq 0 \ (x \in \text{Fix}(T))\} \neq \emptyset. \end{aligned}$$

- (III)  $f_2: H \rightarrow H$  は連続フレッシュェ微分可能な凸関数とし,  $\nabla f_2: H \rightarrow H$  は  $\alpha$ -強単調<sup>4</sup>,  $L_2$ -リプシッツ連続

のとき, 以下の  $x^*$  を見つけよ<sup>5</sup>.

$$\begin{aligned} \{x^*\} &= \text{VI}(\text{VI}(\text{Fix}(T), \nabla f_1), \nabla f_2) \\ &:= \{x^* \in \text{VI}(\text{Fix}(T), \nabla f_1): \langle x - x^*, \nabla f_2(x^*) \rangle \geq 0 \ (x \in \text{VI}(\text{Fix}(T), \nabla f_1))\}. \end{aligned}$$

<sup>1</sup>強凸目的関数に関する最小化問題を解くためのアルゴリズムを提案している文献 [13, 8] とは対照的に, [2] は, 狭義凸目的関数に関する最小化問題を解くためのアルゴリズムを考案している.

<sup>2</sup> $T: H \rightarrow H$  が非拡大であるとは,  $\|T(x) - T(y)\| \leq \|x - y\|$  ( $x, y \in H$ ) が成り立つときをいう. 非拡大写像  $T$  の不動点集合  $\text{Fix}(T)$  は閉凸性を満たす.

<sup>3</sup> $A: H \rightarrow H$  がリプシッツ連続であるとは, ある  $L > 0$  が存在して,  $\|A(x) - A(y)\| \leq L\|x - y\|$  ( $x, y \in H$ ) が成り立つときをいう. 本論文では, このような  $A$  を  $L$ -リプシッツ連続作用素と呼ぶ.

<sup>4</sup> $A: H \rightarrow H$  が強単調であるとは, ある  $\alpha > 0$  が存在して,  $\langle x - y, A(x) - A(y) \rangle \geq \alpha\|x - y\|^2$  ( $x, y \in H$ ) が成り立つときをいう. 本論文では, このような  $A$  を  $\alpha$ -強単調作用素と呼ぶ.

<sup>5</sup> $\text{VI}(\text{Fix}(T), \nabla f_1)$  の閉凸性と  $\nabla f_2$  の強単調及びリプシッツ連続性により,  $\text{VI}(\text{VI}(\text{Fix}(T), \nabla f_1), \nabla f_2)$  の要素は一意に存在する [4, p.585].

ここで, 問題 1.2 が問題 2.1 の一例となることを示そう. 写像  $T: \mathbb{R}^S \rightarrow \mathbb{R}^S$  を

$$T := P_{\mathbb{R}_+^S} P_{C_1} \cdots P_{C_L}.$$

と定義する<sup>6</sup>. このとき,  $P_{\mathbb{R}_+^S}, P_{C_l} (l \in \mathcal{L})$  は非拡大なので,  $T$  は非拡大性を満たし,

$$\emptyset \neq \text{Fix}(T) = \mathbb{R}_+^S \cap \bigcap_{l \in \mathcal{L}} C_l =: C$$

が成り立つ. さらに,  $f_1 := \Phi$  は, 連続微分可能な凸関数であり,

$$\nabla f_1 = I - \sum_{s \in S} w_s P_{D_s}$$

は 2-リプシッツ連続となり,

$$\text{VI}(\text{Fix}(T), \nabla f_1) = \left\{ \mathbf{x}^* \in C: \Phi(\mathbf{x}^*) = \min_{\mathbf{y} \in C} \Phi(\mathbf{y}) \right\} = C_\Phi \neq \emptyset$$

を満たす.  $f_2(\mathbf{x}) := -\mathcal{U}(\mathbf{x}) = -\sum_{s \in S} \log x_s$  ( $\mathbf{x} := (x_s)_{s \in S} \in \mathbb{R}_+^S \setminus \{0\}$ ) は, ある有界閉凸集合上で強単調かつリプシッツ連続となる<sup>7</sup>. さらに,

$$\text{VI}(\text{VI}(\text{Fix}(T), \nabla f_1), \nabla f_2) = \text{VI}(C_\Phi, -\nabla \mathcal{U}) = \underset{\mathbf{x} \in C_\Phi}{\text{Argmax}} \mathcal{U}(\mathbf{x})$$

が成り立つ. 以上のことから, 問題 1.2 が問題 2.1 の一例となることが示された.

## 2.2 問題 2.1 を解くためのアルゴリズムとその収束解析

問題 2.1 を解くための以下のアルゴリズムを提案する [4, 5]:

### アルゴリズム 2.1.

*Step 0.*  $(\alpha_n)_{n \in \mathbb{N}}, (\lambda_n)_{n \in \mathbb{N}} \subset (0, \infty)$ ,  $\mu > 0$  をとり, 初期点  $x_0 \in H$  を任意に選ぶ.  $n := 0$  とする.

*Step 1.*  $x_{n+1} \in H$  を以下で計算する:

$$\begin{cases} y_n := T(x_n - \lambda_n \nabla f_1(x_n)), \\ x_{n+1} := y_n - \mu \alpha_n \nabla f_2(y_n). \end{cases}$$

$n := n + 1$  とおき, *Step 1* に進む.

アルゴリズム 2.1 の収束解析は以下である [4, Theorem 4.1]:

**定理 2.1.**  $(y_n)_{n \in \mathbb{N}}$  は有界であるとし,  $\mu \in (0, 2\alpha/L_2)$  とする.  $(\alpha_n)_{n \in \mathbb{N}} \subset (0, 1]$ ,  $(\lambda_n)_{n \in \mathbb{N}} \subset (0, 2/L_1]$  が  $\lim_{n \rightarrow \infty} \alpha_n = 0$ ,  $\sum_{n=0}^{\infty} \alpha_n = \infty$ ,  $\sum_{n=0}^{\infty} |\alpha_{n+1} - \alpha_n| < \infty$ ,  $\sum_{n=0}^{\infty} |\lambda_{n+1} - \lambda_n| < \infty$ ,  $\lambda_n \leq \alpha_n (n \in \mathbb{N})$  を満たすとする. このとき, アルゴリズム 2.1 は, 以下の性質を満たす:

- (a)  $(x_n)_{n \in \mathbb{N}}$  は有界.
- (b)  $\lim_{n \rightarrow \infty} \|x_n - y_n\| = 0, \lim_{n \rightarrow \infty} \|x_n - T(x_n)\| = 0$ .
- (c)  $\|x_n - y_n\| = o(\lambda_n)$  ならば,  $(x_n)_{n \in \mathbb{N}}$  は問題 2.1 の一意解に強収束する.

<sup>6</sup>閉凸集合  $D \subset H$  への距離射影を  $P_D$  で表す.  $P_D$  は非拡大性を満たす.

<sup>7</sup>例えば, [2, Remark 3.1 (a)] を参照せよ.

### 3 アルゴリズム 2.1 のネットワーク帯域幅割り当て問題への応用

2リンク, 3送信者からなる単純なネットワーク (図 1, [12, Fig.2.1]) に関する帯域幅割り当て問題を考察する. 図 1 での  $C_l$  及び  $D_s$  は以下のように定義される:  $C_l := \{(x_1, x_2, x_3)^T \in \mathbb{R}^3 : x_l + x_3 \leq c_l\}$  ( $l = 1, 2$ ),  $D_s := \{(x_1, x_2, x_3)^T \in \mathbb{R}^3 : x_s \geq r_s\}$  ( $s = 1, 2, 3$ ). ただし,  $c_1 := 3, c_2 := 4, r_1 := 3, r_2 := 4, r_3 := 5$  とする. このケースでは,

$$C \cap D = \mathbb{R}_+^3 \cap \bigcap_{l=1}^2 C_l \cap \bigcap_{s=1}^3 D_s = \emptyset$$

が成り立たため, 帯域幅割り当て問題の制約集合を,

$$C_\Phi := \left\{ \mathbf{x}^* \in C : \Phi(\mathbf{x}^*) = \min_{\mathbf{y} \in C} \Phi(\mathbf{y}) \right\}, \quad \Phi(\mathbf{x}) := \frac{1}{2} \sum_{s=1}^3 \frac{1}{3} d(\mathbf{x}, D_s)^2 \quad (\mathbf{x} \in \mathbb{R}^3)$$

で定義する. 以上により, 帯域幅割り当て問題は, 以下で与えられる:

$$\mathcal{U}(\mathbf{x}^*) = \max_{\mathbf{y} \in C_\Phi} \mathcal{U}(\mathbf{y}) \text{ となる } \mathbf{x}^* \in C_\Phi \text{ を見つけよ.} \quad (3)$$

問題 (3) を解くための既存手法 [6] は, 以下で与えられる. この手法は, 文献 [13] で提案されたハイブリッド最急降下法 (hybrid steepest descent method (HSDM)) に基づいている<sup>8</sup>.

[HSDM]

$$\mathbf{x}_{n+1} := P_{\mathbb{R}_+^3 \cap C_1} \left[ 0.9P_{C_2} + \sum_{s=1}^3 \frac{0.1}{3} P_{D_s} \right] \left( \mathbf{x}_n + \frac{1}{10^3 \sqrt{n+1}} \nabla \mathcal{U}(\mathbf{x}_n) \right) \quad (n \in \mathbb{N}).$$

HSDM は, 以下で定義される  $C_\Psi$  上の  $\mathcal{U}$  の最大点に収束することが保証される [13, 6].

$$C_\Psi := \left\{ \bar{\mathbf{x}} \in \mathbb{R}_+^3 \cap C_1 : \Psi(\bar{\mathbf{x}}) = \min_{\mathbf{y} \in \mathbb{R}_+^3 \cap C_1} \Psi(\mathbf{y}) \right\},$$

$$\Psi(\mathbf{x}) := \frac{1}{2} \cdot 0.9 d(\mathbf{x}, C_2)^2 + \frac{1}{2} \sum_{s=1}^3 \frac{0.1}{3} d(\mathbf{x}, D_s)^2 \quad (\mathbf{x} \in \mathbb{R}^3).$$

しかしながら, 理論的には, HSDM は問題 (3) の解 ( $C_\Phi$  上の  $\mathcal{U}$  の最大点) に収束することは保証されていない.

問題 (3) に対するアルゴリズム 2.1 は, 以下である:

[アルゴリズム 2.1]

$$\begin{cases} \mathbf{y}_n := P_{\mathbb{R}_+^3} P_{C_1} P_{C_2} \left( \left( 1 - \frac{1}{\sqrt{n+1}} \right) \mathbf{x}_n + \frac{1}{\sqrt{n+1}} \sum_{s=1}^3 \frac{1}{3} P_{D_s}(\mathbf{x}_n) \right), \\ \mathbf{x}_{n+1} := \mathbf{y}_n + \frac{1}{10^3 \sqrt{n+1}} \nabla \mathcal{U}(\mathbf{y}_n) \quad (n \in \mathbb{N}). \end{cases}$$

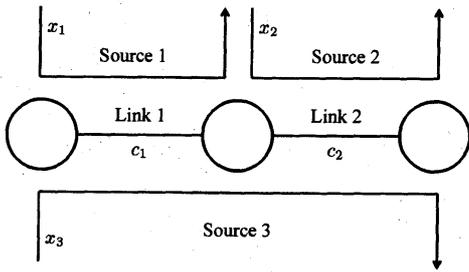


図 1:  $c_1 := 3, c_2 := 4, r_1 := 3, r_2 := 4, r_3 := 5$  での 2 リンク, 3 送信者のネットワーク

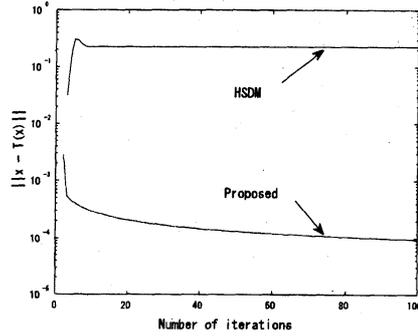


図 2: HSDM 及びアルゴリズム 2.1 に対する  $\|x_n - T(x_n)\| = \|x_n - P_{\mathbb{R}_+^3} P_{C_1} P_{C_2}(x_n)\|$  の挙動

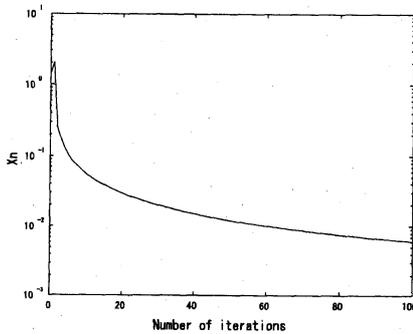


図 3: アルゴリズム 2.1 で生成された  $(X_n)_{n=1}^{100} := (\sqrt{n+1} \|x_n - y_n\|)_{n=1}^{100}$  の挙動

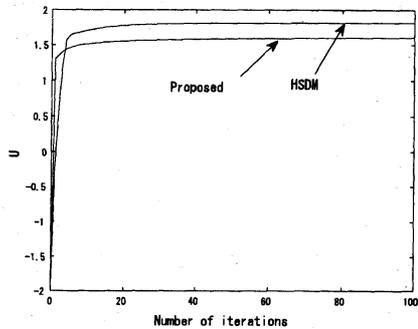


図 4: HSDM 及びアルゴリズム 2.1 に対する  $u(x) := \sum_{s=1}^3 \log x_s$  の挙動 (アルゴリズム 2.1 での  $x_{100}$  を計算するのに必要な CPU タイムは約 0.01 秒.)

HSDM 及びアルゴリズム 2.1 が絶対集合  $C := \mathbb{R}_+^3 \cap C_1 \cap C_2$  に収束するかどうかを確認するために、評価指標

$$\|x - T(x)\| := \|x - P_{\mathbb{R}_+^3} P_{C_1} P_{C_2}(x)\| \quad (x \in \mathbb{R}^3)$$

を用いる。  $C := \mathbb{R}_+^3 \cap C_1 \cap C_2 \neq \emptyset$  に対して、  $x \in \mathbb{R}^3$  が  $\|x - T(x)\| = 0$  を満たす必要十分条件は、  $x \in C$  である。 図 2 は、 HSDM とアルゴリズム 2.1 (Proposed) に対する  $\|x_n - T(x_n)\|$  ( $n = 1, 2, \dots, 100$ ) の挙動を表す。 この図は、 HSDM で生成された  $(\|x_n - T(x_n)\|)_{n \in \mathbb{N}}$  が 0 に収束しない、つまり、 HSDM は  $C$  にさえ収束しないことを示している<sup>9</sup>。 一方で、 アルゴリズム 2.1 で得られた  $(\|x_n - T(x_n)\|)_{n \in \mathbb{N}}$  は 0 に収束している、つまり、 アルゴリズム 2.1 は  $C = \text{Fix}(T)$  に収束することが言える。

図 3 は、 アルゴリズム 2.1 で生成された

$$(X_n)_{n=1}^{100} := \left( \sqrt{n+1} \|x_n - y_n\| \right)_{n=1}^{100}$$

<sup>8</sup>ここで紹介する HSDM は、  $C$  に出来るだけ収束させるようにパラメータ  $v_2 := 0.9, u := u_s = 0.1/3$  ( $s = 1, 2, 3$ ) を用いている。 詳細については、 文献 [6, 2] を参照せよ。

<sup>9</sup>パラメータ  $v_2$  が 0.99 のとき、 HSDM が  $C$  に収束することを確認した。

の挙動を示しており,  $(X_n)_{n \in \mathbb{N}}$  が 0 に収束することが言える. これは, アルゴリズム 2.1 が収束条件  $\|x_n - y_n\| = o(1/\sqrt{n+1})$  を満たすことを意味する<sup>10</sup>. よって, 定理 2.1 (c) により, アルゴリズム 2.1 は問題 (3) の一意解に収束することが保証される.

アルゴリズム 2.1 に必要な反復回数や CPU タイムは図 4 に示す. アルゴリズム 2.1 は  $n \geq 10$  で安定になり,  $x_{100}$  を計算するのに必要な CPU タイムは約 0.01 秒であった. 図 2-4 及び定理 2.1 から, アルゴリズム 2.1 は最適な帯域幅 ( $x_1^* \approx 1.387$ ,  $x_2^* \approx 2.387$ ,  $x_3^* \approx 1.613$ ) を見つけることができる. 一方で, HSDM は  $n \geq 20$  で安定するが, 図 2 にも見られるように, HSDM は  $C$  に収束しないので, 問題 (3) を解くことはできない.

### 謝辞

本研究の一部は, 日本学術振興会における科学研究費補助金基盤研究 (C)(課題番号 23500090) 及び若手研究 (B)(課題番号 23760077) による支援を受けている. ここに記し謝意を表す.

### 参考文献

- [1] P. L. COMBETTES AND P. BONDON, *Hard-constrained inconsistent signal feasibility problems*, IEEE Transactions on Signal Processing, 47 (1999), pp. 2460-2468.
- [2] H. IIDUKA, *Fixed point optimization algorithm and its application to network bandwidth allocation*, Journal of Computational and Applied Mathematics, doi:10.1016/j.cam.2011.10.004 (to appear).
- [3] H. IIDUKA, *Fixed point optimization algorithm and its application to power control in CDMA data networks*, Mathematical Programming, doi:10.1007/s10107-010-0427-x (to appear).
- [4] H. IIDUKA, *Iterative algorithm for solving triple-hierarchical constrained optimization problem*, Journal of Optimization Theory and Applications, 148 (2011), pp. 580-592.
- [5] H. IIDUKA, *Iterative algorithm for triple-hierarchical constrained non-convex optimization problem and its application to network bandwidth allocation*, submitted.
- [6] H. IIDUKA AND M. UCHIDA, *Fixed point optimization algorithms for network bandwidth allocation problems with compoundable constraints*, IEEE Communications Letters, 16 (2011), pp. 596-598.

<sup>10</sup> $((n+1)^a \|x_n - y_n\|)_{n \in \mathbb{N}}$  ( $a = 2, 3$ ) は, 0 に収束しなかったことも確認した. このことから, ゆっくり 0 に収束するステップサイズ  $(\lambda_n)_{n \in \mathbb{N}}$  (例えば,  $\lambda_n := 1/(n+1)^a$  ( $a \in (0, 1)$ )) を使用することが問題 (3) を解く上で, 望ましいと言えよう.

- [7] 飯塚 秀明, 内田 真人, 不動点最適化手法とネットワーク資源割り当て問題への応用, 応用数理, 21 (3) (2011), pp. 6-21.
- [8] H. IIDUKA AND I. YAMADA, *A use of conjugate gradient direction for the convex optimization problem over the fixed point set of a nonexpansive mapping*, SIAM Journal on Optimization, 19 (2009), pp. 1881-1893.
- [9] H. IIDUKA AND I. YAMADA, *Computational method for solving stochastic linear-quadratic control problem given an unsolvable stochastic algebraic Riccati equation*, submitted.
- [10] F. P. KELLY, *Charging and rate control for elastic traffic*, European Transactions on Telecommunications, 8 (1997), pp. 33-37.
- [11] J. MO AND J. WALRAND, *Fair end-to-end window-based congestion control*, IEEE/ACM Transactions on Networking, 8 (2000), pp. 556-567.
- [12] R. SRIKANT, *Mathematics of Internet Congestion Control*, Birkhauser, 2004.
- [13] I. YAMADA, *The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings*, In: D. Butnariu, Y. Censor, S. Reich (eds.) *Inherently Parallel Algorithms for Feasibility and Optimization and Their Applications*, Elsevier (2001), pp. 473-504.