

信州大学工学部

学士論文

到着期限のあるデータ伝送における
歪み最小化アルゴリズム

指導教員 西新 幹彦 准教授

学科 電気電子工学科
学籍番号 06T2074A
氏名 原 博紀

2010 年 2 月 13 日

目次

1	序章	1
2	伝送システムの定式化	1
2.1	伝送モデル	1
2.2	歪み	2
2.3	関数の凸性	3
2.4	本研究の目的と本論文の構成	3
3	伝送路に雑音がなく、データの上限が有限であり、到着期限が連続の場合	5
4	伝送路に雑音があり、データの上限が無限であり、到着期限が離散の場合	7
5	本論	7
6	まとめ	11
	謝辞	11
	参考文献	11
	付録 A 補題の証明	12
	付録 B ソースコード	17
	B.1 basic-algorithm のプログラム	17

1 序章

今日の社会では、インターネット上での情報交換が日常的に行われている。特に、映画や音楽といった娯楽は、今や店舗に行かずともインターネット上で購入し、その場で楽しむことができる。その中で、映画のように動く動画と、写真のように静止している画像とでは、情報を伝送する方法が異なる。動画の場合、一連の動きを、フレームと呼ばれる複数のコマに分け、それらを順次流すことで、動きを再現する。一方で、写真のような画像は、一つのフレームに描かれた画を伝送し、それを静止画として人は認識する。ここで受け手が求めることは、元となる情報を、より忠実に、再現できるかである。すなわち、動画の場合、コマ遅れやコマ切れがなくスムーズに再生できるか、静止画の場合は画像が欠けることがなく鮮やかに再現できるかである。そして、動画と静止画の伝送を比較した場合、単純に伝送するフレーム数の違いから、元のデータを忠実に再現するのは動画の方が難しいと考えるのが自然である。

このように、インターネットを介した動画の再生を、ビデオ・オン・デマンドまたはストリーミングと呼ぶ。ビデオ・オン・デマンドの場合、元となる動画の再生時間と、視聴者が視聴する時間が等しいことが理想である。この視聴者が要求する時間を、到着期限と呼ぶことにする。従来の研究では、このような情報の伝送について、以下ような問題設定のもと、研究がおこなわれてきた [1]。

- フレームを情報源、視聴者を受け手とし、これらの情報源は伝送路を介して送信される。
- 情報源に存在するデータ量の上限、伝送路の誤りの有無、視聴者が求める到着期限などあらかじめ条件を設定する。
- そして、視聴者が求める理想的な再生を行うために、あらかじめ設定された各条件の下で、最適な伝送方法を求める。

そして、これらの条件の下、導き出された結果を、伝送計画と呼び、これを導くアルゴリズムが求められてきた。本研究は、あらかじめ決めた問題設定のもと、最適な伝送計画を導くアルゴリズムを求めることを目的としてアプローチを行っていく。

2 伝送システムの定式化

2.1 伝送モデル

ビデオ・オン・デマンドのような伝送を単純化し、その単純化した伝送モデルを図 1 に示す。以下では、図 1 を用いて考えていく。

あらかじめエンコードされた N 個のデータが情報源に存在し、それらをソースシンボルと

呼び、伝送路を介して受け手へ伝送される．すなわち、 N 個のソースシンボルを N 個のフレームと考えた場合、この N 個のフレームが一つ一つ順次伝送され、受け手へ届く．ここで、受け手は、 N 個のフレームの伝送が完了しなければならない時間を、自由に要求できるものとし、これを到着期限と呼ぶことにする．この到着期限を、一つ一つのフレームごと定め、それまで送られたフレームの packets 長の合計と比較する．すなわち、すでにエンコードされた、各フレームの有限な packets 長を y_i ($i = 1, \dots, N$)、到着期限を M_i ($i = 1, \dots, N$) すると、

$$\sum_{j=1}^i y_j \leq M_i, \quad (i = 1, \dots, N) \quad (1)$$

を満たす伝送を考える．式 (1) を満たす場合、feasible と呼ぶことにする．本研究では、次のように問題設定を行い、伝送モデルについてアプローチを行っていく．

- あらかじめエンコードされた各フレームの packets 長は、無限である．
- 実際に伝送される各フレームの packets 長を z_i ($i = 1, \dots, N$) とし、この大きさは到着期限 M_i ($i = 1, \dots, N$) にのみ依存し決定される．
- 実際に伝送される packets 長 z_i を N 次元ベクトル z を用いて表し、このベクトル $z = (z_1, \dots, z_N)$ を伝送計画と呼ぶ．
- 伝送路では、誤りは起きないものとする．

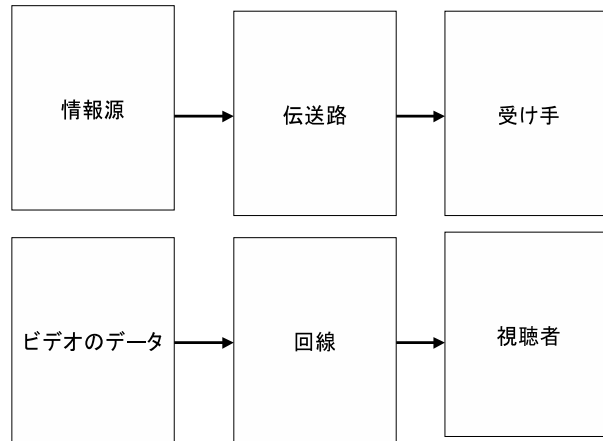


図1 伝送モデル

2.2 歪み

一般に，受け手で復号されるデータと，元のデータは完全には一致しない．受け手で，より多くのデータを受け取れば，復号されたデータは，元のデータに近づくと考えるのが自然である．そこで，元のデータと，実際に受け手で復号されたデータの大きさの違いを測る尺度として，歪みを導入する．本研究では，この歪みを以下のように関数を用いて測り，この関数を歪み関数と呼ぶ事にする．歪み関数 $d(z)$ は，狭義に下に凸でかつ単調減少とする (図 2) ．

伝送されるパケット長 z_i ($i = 1, \dots, N$) に対して， $d(z_i)$ を z_i の歪みと呼ぶ．伝送計画の歪み $D(z)$ を

$$D(z) \triangleq \sum_{i=1}^N d(z_i) \quad (2)$$

と定める．

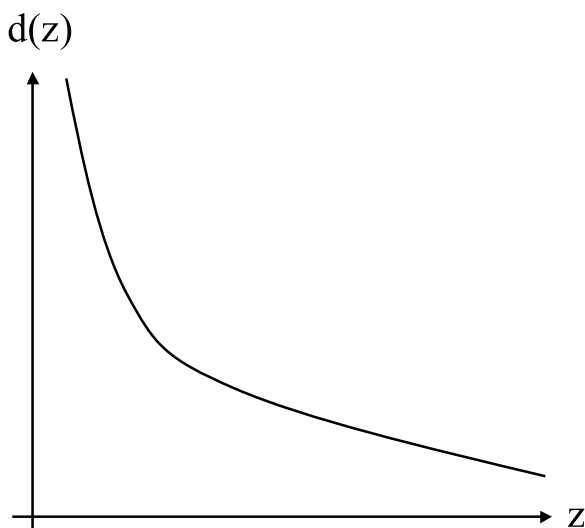


図 2 歪み関数

2.3 関数の凸性

k 次元実数値ベクトル x の実数値関数 $f(x)$ が，定義域に属する任意ベクトル x_1, x_2 と任意の実数 γ_1, γ_2 ($0 \leq \gamma_1, 0 \leq \gamma_2, \gamma_1 + \gamma_2 = 1$) に対して，

$$f(\gamma_1 x_1 + \gamma_2 x_2) \leq \gamma_1 f(x_1) + \gamma_2 f(x_2) \quad (3)$$

が成立するとき, $f(x)$ は下に凸であるという. 特に, $0 < \gamma_1, 0 < \gamma_2$ に対する等号が成立するのが $x_1 = x_2$ の時に限られるとき, $f(x)$ は狭義に下に凸と呼ばれる [2].

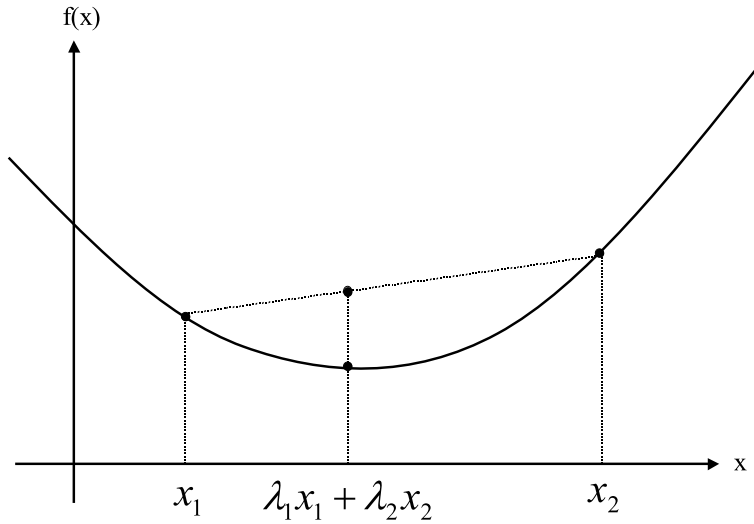


図 3 関数の凸性

2.4 本研究の目的と本論文の構成

本研究では, 従来の研究では行われなかった, 決められた条件下で, 式 (2) を最小にする様な伝送計画 z を導くアルゴリズムを求めることを目的とする. 以下に従来の研究と, 本研究で行う条件を表にして表す (表 1).

本論文では次のような構成をとる. 3 章では, 伝送モデルで, 伝送路に誤りがなく, データの上限が有限であり, 到着期限が連続の場合について述べる. 4 章では, 伝送路に誤りがあり, データの上限が無限であり, 到着期限が離散の場合について述べる. 5 章では, 本研究で, 伝送路に誤りがなく, データの上限が無限であり, 到着期限が連続値の場合について述べる. 6 章では結論を述べる.

表 1 従来の研究と本研究

データの上限	伝送路の誤り	到着期限	
有限	誤りなし	連続	従来の研究
無限	誤りあり	離散	従来の研究
無限	誤りなし	連続	本研究

3 伝送路に雑音がなく、データの上限が有限であり、到着期限が連続の場合

この章では、データの上限が有限で、伝送路に誤りがなく、到着期限が連続値を取る場合 [1] の、伝送計画について紹介する。図 1 の伝送モデルにおいて、情報源にあらかじめエンコードされた N 個の packets が存在し、各 packet 長を、 γ_i ($i = 1, \dots, N$) と表す。そして、実際に伝送される各 packet 長を y_i ($i = 1, \dots, N$) とする。そのとき、実際に送る packet 長は、あらかじめ用意された packet 長を超えてはならないので、

$$0 \leq y_i \leq \gamma_i \quad (i = 1, \dots, N) \quad (4)$$

という関係式が成り立つ。一方で、各 packet に対して、到着期限 M_i ($i = 1, \dots, N$) が決まっており、各 packet は式 (2) を満たすように、伝送される。このとき、伝送計画 $\mathbf{y} = (y_1, \dots, y_N)$ の歪み $D(\mathbf{y})$ を最小にしかつ feasible であるベクトル $\mathbf{y}$ は、図 3 のアルゴリズムより導かれる。付録に c 言語による実装例を示す。

定理 1([1]) 狭義に凸な関数 $d(y)$ に対して、basic algorithm は、伝送計画 $\mathbf{y}$ の歪み $D(\mathbf{y})$ を最小かつ feasible となるベクトル $\mathbf{y}$ を唯一つ導く。

入力: $\gamma_i, \mu_i = \frac{M_i}{i} \ (i = 1, \dots, N)$

出力: $y_i \ (i = 1, \dots, N)$

1. $\forall j \in \{i\}_{i=1}^N$ に対して, $\mathcal{I}_i = \{1, \dots, j\}$ を定義する.
2. $\mathcal{I} = \mathcal{I}_N, \mu_i = \frac{M_i}{i}, i \in \mathcal{I}$ とする.
3. $z = \min\{\{\mu_i\}_{i \in \mathcal{I}} \cup \{\gamma_i\}\}$ とする.
4. $\forall i \in \{j \in \mathcal{I} | \gamma_j = z\}$ に対して, $y_i^* = z$ とする.
5. $\hat{j} = \max\{j \in \mathcal{I} | \mu_j = z\}$ とする.
6. $\forall i \in \mathcal{I}_{\hat{j}} \cap \mathcal{I}, y_i^* = z$ と決める.
7. $\mathcal{I}_i = \mathcal{I} - \{j | y_j^* = z\}$ とおく.
8. $\forall i \in \mathcal{I}, \mu_i = \frac{M_i - \sum_{j \in \mathcal{I}_i - \mathcal{I}} \mu_j}{|\mathcal{I}_i \cap \mathcal{I}|}$ とする.
9. もし, $\mathcal{I} \neq \emptyset$ ならステップ 3 の戻り, そうでなければ終了.

図 4 basic algorithm

4 伝送路に雑音があり、データの上限が無限であり、到着期限が離散の場合

この章では、データの上限が無限で、伝送路に雑音があり、到着期限が離散値を取る場合 [1] の、伝送計画について紹介する. 図 1 の伝送モデルにおいて、情報源にあらかじめエンコードされた N 個の packets が存在している. ここで、各 packet 長 $z_i \ (i = 1, \dots, N)$ は無限であるものとする. 伝送路では、誤りがあり、情報源から受け手で正しく受け取ったビット数を $b_i \ (i = 1, \dots, N)$ とする. ここで、誤りを次のように決める.

- 各スロットごと、確率 p で、1 ビット packet の伝送に成功し、 $1 - p$ で失敗する.

したがって、伝送する packet 長 z_i と、受け手で正しく受け取られたビット数 b_i の関係は、パラメータ z_i の、二項分布を用いて

$$\Pr\{b_i = k\} = \binom{z_i}{k} p^k (1 - p)^{z_i - k} \quad (5)$$

と表すことができる. それゆえ、歪み関数と期待値を用いて、 $\sum_{i=1}^N E[d(b_i)]$ が最小かつ feasible となる伝送計画を導くことが課題となる. ここで、

$$g(z_i) = E[d(b_i)] \quad (6)$$

とおけば,

$$G(z) \triangleq \sum_{i=1}^N g(z_i)$$

このとき, $G(z)$ が最小かつ feasible となるベクトル $z = (z_1, \dots, z_N)$ を求めればよい.

伝送計画の歪み $G(z)$ が最小かつ feasible な伝送計画 z は, 図 5 の open-loop algorithm より導かれる.

入力: M_i ($i = 1, \dots, N$)

出力: z_i ($i = 1, \dots, N$)

1. $\hat{j} = \max \left\{ \arg \min_j \left\{ \left\lfloor \frac{M_j}{j} \right\rfloor \right\}_{j=1}^N \right\}, k = M_{\hat{j}} - \hat{j} \left\lfloor \frac{M_{\hat{j}}}{\hat{j}} \right\rfloor$
- 2.

$$z_j^* = \begin{cases} \left\lfloor \frac{M_{\hat{j}}}{\hat{j}} \right\rfloor & j = 1, \dots, \hat{j} - k \\ \left\lfloor \frac{M_{\hat{j}}}{\hat{j}} \right\rfloor + 1 & j = \hat{j} - k + 1, \dots, \hat{j} \end{cases}$$

3. もし, $\hat{j} < N$ なら, $\{M_j\}_{j=1}^{\hat{j}}$ を取り除き, $j > \hat{j}$ に対して, $j = j - \hat{j}$ をセットし, M_j の残りを更新して, ステップ 1 に戻る. そうでなければ終了.

図 5 open-loop algorithm

定理 2([1]) 狭義に下に凸な関数 $d(z)$ に対して, open-loop algorithm は, 伝送計画の歪み $G(z)$ を最小かつ feasible となるベクトル z を導く.

5 本論

本研究では, 伝送モデル図 1 で, データの上限が無限であり, 伝送路に誤りがなく, 到着期限が連続値を取る場合の, 伝送計画について考える. 本章で述べる結果が本研究の成果である. 伝送されるパケット長 z_i ($i = 1, \dots, N$) は, 到着期限 M_i ($i = 1, \dots, N$) にのみ依存し, 伝送計画 $z = (z_1, \dots, z_N)$ の歪みが最小かつ feasible なベクトル z は図 6 のアルゴリズムより導かれる.

定理 3 狭義に下に凸な関数 $d(z)$ に対して, 最適な伝送計画を導くアルゴリズムは, あらかじめ決められた feasible な伝送計画 z の歪み $D(z)$ よりも, 歪みが小さくかつ feasible である最適な伝送計画 z' を導く.

入力: M_i ($i = 1, \dots, N$)

出力: z_i ($i = 1, \dots, N$)

1. $i = 0, j_0 = 0, M_0 = 0$ とする.
2. $j_{i+1} = \max\{\arg \min_{j_i < j \leq N} \{\frac{M_j - M_{j_i}}{j - j_i}\}\}$
3. $z_j = \frac{M_j - M_{j_0}}{j - j_0}, j = j_0 + 1, \dots, j_{i+1}$
4. $j_{i+1} < N$ ならば, $i = i + 1$ として 2. に戻る. そうでなければ終了.

図 6 最適な伝送計画を導くアルゴリズム

この定理は以下に述べる補題 1 から補題 3 を用いて表すことができる. 補題 1 では, あらかじめ決められた伝送計画 z の歪み $D(z)$ より, 歪みが小さくかつ feasible な伝送計画が存在する場合, その新たな伝送計画 z' を導く. また, 補題 3 では, 補題 1 より求められた伝送計画 z' を, 最適な伝送計画へと導く. 補題 2 は補題 3 で必要な条件を証明している. 各補題の証明については付録に掲載する.

補題 1 条件

$$\sum_{i=1}^N z_i = M_N \quad (7)$$

のもとで, 不等式

$$\sum_{i=1}^j z_i \leq M_j, \quad (j = 1, \dots, N) \quad (8)$$

を満たす N 次元ベクトル z の要素のうち $j = k+1, \dots, k+l$ 番目の要素が $z_{k+1} = \dots = z_{k+l}$ でないとする. また,

$$\frac{m}{M_{k+m} - M_k} \leq \frac{l}{M_{k+l} - M_k}, \quad (m = 1, \dots, l) \quad (9)$$

とする. このもとで,

$$a \triangleq \frac{1}{l} \sum_{i=k+1}^{k+l} z_i, \quad (10)$$

$$z'_j \triangleq \begin{cases} a & (j = k+1, \dots, k+l) \\ z_j & \text{otherwise} \end{cases} \quad (11)$$

とする.

このとき ,

$$G(z) - G(z') > 0 \quad (12)$$

$$\sum_{i=1}^j z'_i \leq M_j, \quad (j = 1, \dots, N) \quad (13)$$

が成立する.

補題 2 $g(\cdot)$ を凸関数とする.

$$\begin{cases} x_1 < x_2 \\ 0 < \delta < x_2 - x_1 \\ 0 < \mu < 1 \\ x'_1 \triangleq x_1 + (1 - \mu)\delta \\ x'_2 \triangleq x_2 - \mu\delta \end{cases} \quad (14)$$

のとき ,

$$\mu g(x_1) + (1 - \mu)g(x_2) > \mu g(x'_1) + (1 - \mu)g(x'_2) \quad (15)$$

が成立する.

補題 3 条件

$$\sum_{i=1}^N z_i = M_N \quad (16)$$

のもとで , 不等式

$$\sum_{i=1}^j \leq M_j, \quad (j = 1, \dots, N) \quad (17)$$

を満たす N 次元ベクトル $z = (z_1, \dots, z_N)$ の要素のうち ,

$$z_1 = \dots = z_{j_1} < z_{j_1+1} = \dots = z_{j_2} < z_{j_2+1} = \dots = z_{j_3} < \dots = z_{j_K} \quad (18)$$

を満たす $j_1, j_2, j_3, \dots, j_K$ を定める.

このとき ,

$$z_1 = \dots = z_{j_1} = a \quad (19)$$

$$z_{j_{K-1}+1} = \dots = z_{j_K} = b \quad (20)$$

さらに ,

$$\delta_\alpha \triangleq M_{j_\alpha} - \sum_{i=1}^{j_\alpha} z_i, \quad (\alpha = 1, \dots, K-1) \quad (21)$$

$$k = \arg \min_{\alpha} \delta_\alpha \quad (22)$$

と仮定する. このもとで,

$$a' = a + \frac{\delta_k}{n_a} \quad (23)$$

$$b' = b - \frac{\delta_k}{n_b} \quad (24)$$

とすると, 新たなベクトル z' が以下のように表せる.

$$z' = (z'_1, \dots, z'_{j'_1}, z_{j_1+1}, \dots, z_{j_{K-1}}, z'_{j_{K-1}+1}, \dots, z'_{j_K}) \quad (25)$$

このとき,

$$G(z) - G(z') > 0 \quad (26)$$

かつ feasible である.

定理の証明: 補題 2 より補題 3 の最適性が証明される. したがって, 補題 1 と補題 3 より, 定理 3 は明らかである.

6 まとめ

ビデオ・オン・デマンドやストリーミングと呼ばれる, 動画の再生では, 元のデータの情報量と受け手で受信されるデータの情報量は完全には一致しない.

そこで, 本研究では, 簡単な伝送モデルを用いて, 元のデータ量と受け手で受け取られる情報量の大きさの違いを表す尺度として, 狭義に下に凸な減少関数である, 歪み関数を導入した. 問題設定として, 情報源のデータの上限, 伝送路の誤りの有無, 到着期限なる受けで要求する時間を設定し, 情報源から送られる N 個のフレームの各パケット長を, N 次元ベクトルを用いて表し, これを伝送計画と呼んだ. この伝送計画が, あらかじめ決められた到着期限を守る場合, feasible であると呼び, 伝送計画が, feasible かつ歪みが最小となる, 最適な伝送計画を導くアルゴリズムを求めることを目的とした.

本研究では, データの上限が無限であり, 伝送路で誤りはなく, 受け手で要求される到着期限は, 連続な正の実数値を取る, という条件の下, 伝送計画が feasible かつ歪みが最小となる N 次元ベクトルを導く, アルゴリズムを求めることを目的とした. その結果, 最適な伝送計画を導くアルゴリズムにより, 最適な伝送計画を導くことができるということが明らかになった. 今後の課題としては, まだ行われていない組み合わせの条件下で, 最適な伝送計画を導くアルゴリズムはどのように表せるか. さらに, 符号化や復号化を考慮した場合, どのように問題設定をするか, また, そのときの最適な伝送計画を導くアルゴリズムは, どのように表すことができるかなど, さまざまな条件を検証する必要がある.

謝辞

本研究を行うにあたり，数多くの助言，指導をしていただいた指導教員西新幹彦准教授，また西新研究室の皆様に感謝の意を表する．

参考文献

- [1] Azadeh Faridi, Anythony Ephremides, “Distortion Control For Delay-sensitive Sources” IEEE Transaction on information, Vol.54, No.8 pp.3399-3406 August 2008.
- [2] 韓太瞬, 小林欣吾, 情報と符号化の数理, 培風館,1999.

付録 A 補題の証明

補題 1 条件

$$\sum_{i=1}^N z_i = M_N \quad (27)$$

のもとで，不等式

$$\sum_{i=1}^j z_i \leq M_j, \quad (j = 1, \dots, N) \quad (28)$$

を満たす N 次元ベクトル z の要素のうち $j = k+1, \dots, k+l$ 番目の要素が $z_{k+1} = \dots = z_{k+l}$ でないとする。また，

$$\frac{m}{M_{k+m} - M_k} \leq \frac{l}{M_{k+l} - M_k}, \quad (m = 1, \dots, l) \quad (29)$$

とする。このもとで，

$$a \triangleq \frac{1}{l} \sum_{i=k+1}^{k+l} z_i, \quad (30)$$

$$z'_j \triangleq \begin{cases} a & (j = k+1, \dots, k+l) \\ z_j & \text{otherwise} \end{cases} \quad (31)$$

とする。

このとき，

$$G(z) - G(z') > 0 \quad (32)$$

$$\sum_{i=1}^j z'_i \leq M_j, \quad (j = 1, \dots, N) \quad (33)$$

が成立する。

証明；

$$\begin{aligned} G(z) - G(z') &= g(z_{k+1}) + g(z_{k+2}) + \dots + g(z_{k+l}) - g(z'_{k+1}) - g(z'_{k+2}) \dots - g(z'_{k+l}) \\ &= g(z_{k+1}) + g(z_{k+2}) + \dots + g(z_{k+l}) - lg(a) \\ &> 0 \end{aligned} \quad (34)$$

ただし，最後の不等式は Jensen の不等式を用いた。

故に,

$$G(z) - G(z') > 0$$

定理の前半が証明された.

次に, ベクトル z' で, 式 (39),(42) より

$$\sum_{i=1}^j z'_i \leq M_j, \quad (j = 1, \dots, k) \quad (35)$$

を満たしている.

ここで, z'_{k+m} は,

$$\begin{aligned} \sum_{i=1}^{k+m} z'_i &= \sum_{i=1}^k z'_i + \sum_{i=k+1}^{k+m} z'_i \\ &\stackrel{(39)}{=} \sum_{z_i}^k z_i + ma \\ &= \frac{l-m}{l} \sum_{i=1}^k z_i + \frac{m}{l} \left(\sum_{i=1}^k z_i + la \right) \\ &\stackrel{(41)}{=} \frac{l-m}{l} \sum_{i=1}^k z_i + \frac{m}{l} \left(\sum_{i=1}^k z_i + \sum_{i=k+1}^{k+l} z_i \right) \\ &= \frac{l-m}{l} \sum_{i=1}^k z_i + \frac{m}{l} \sum_{i=1}^{k+l} z_i \\ &\stackrel{(39)}{\leq} M_{k+m} \end{aligned} \quad (36)$$

一方で, z'_N は,

$$\begin{aligned} \sum_{i=1}^N z'_i &= \sum_{i=1}^{k+l} z'_i + \sum_{i=k+l+1}^N z'_i \\ &\stackrel{(45)}{\leq} M_{k+l} + \sum_{i=k+l+1}^N z'_i \\ &\stackrel{(41)}{=} M_{k+l} + \sum_{i=k+l+1}^N z_i \\ &\stackrel{(38)}{\leq} M_{k+l} + M_N - M_{k+l} \\ &= M_N \end{aligned} \quad (37)$$

以上より，ベクトル z' は，feasible である．□

補題 2 $g(\cdot)$ を凸関数とする．

$$\begin{cases} x_1 < x_2 \\ 0 < \delta < x_2 - x_1 \\ 0 < \mu < 1 \\ x'_1 \triangleq x_1 + (1 - \mu)\delta \\ x'_2 \triangleq x_2 - \mu\delta \end{cases} \quad (38)$$

のとき，

$$\mu g(x_1) + (1 - \mu)g(x_2) > \mu g(x'_1) + (1 - \mu)g(x'_2) \quad (39)$$

が成立する．

証明; f は狭義に凸な関数であり，また $x_1 < x_2$ ， $0 < \lambda < 1$ のとき，

$$f(\lambda x_1 + (1 - \lambda)x_2) > \lambda f(x_1) + (1 - \lambda)f(x_2) \quad (40)$$

が成り立つ．

(1) $x_1 < x'_1 < x'_2 < x_2$ なる $x'_1 < x_2$ を考える．

このとき，適当な λ を用いて，

$$x'_1 = (1 - \lambda_1)x_1 + \lambda_1 x_2 \quad (41)$$

$$x'_2 = \lambda_2 x_1 + (1 - \lambda_2)x_2 \quad (42)$$

と書けるので，

$$f(x'_1) < (1 - \lambda_1)f(x_1) + \lambda_1 f(x_2) \quad (43)$$

$$f(x'_2) < \lambda_2 f(x_1) + (1 - \lambda_2)f(x_2) \quad (44)$$

任意の $0 < \lambda_3 < 1$ を満たす λ_3 と式 (43),(44) より，

$$\lambda_3 f(x'_1) + (1 - \lambda_3)f(x'_2) < (\lambda_3(1 - \lambda_1) + (1 - \lambda_3)\lambda_2)f(x_1) + (\lambda_3\lambda_1 + (1 - \lambda_3)(1 - \lambda_2))f(x_2) \quad (45)$$

ここで，

$$\lambda_3 = \lambda_3(1 - \lambda_1) + (1 - \lambda_3)\lambda_2 \quad (46)$$

を λ_3 について解くと

$$\lambda_3 = \frac{\lambda_2}{\lambda_1 + \lambda_2} \quad (47)$$

このとき,

$$\begin{aligned} \lambda_3 \lambda_1 + (1 - \lambda_3)(1 - \lambda_2) &= \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} + \frac{\lambda_1}{\lambda_1 + \lambda_2} (1 - \lambda_2) \\ &= \frac{\lambda_1}{\lambda_1 + \lambda_2} \end{aligned} \quad (48)$$

以上より, 式 (45) は,

$$\frac{\lambda_2}{\lambda_1 + \lambda_2} f(x'_1) + \frac{\lambda_1}{\lambda_1 + \lambda_2} f(x'_2) < \frac{\lambda_2}{\lambda_1 + \lambda_2} f(x_1) + \frac{\lambda_1}{\lambda \lambda_1 + \lambda_2} f(x_2) \quad (49)$$

と表すことができる.

(2) $x_1 < x_2$ とする. $0 < \delta < x_2 - x_1$, $0 < u < 1$ に対して,

$$x'_1 \triangleq x_1 + (1 - \mu)\delta x'_2 \quad (50)$$

$$x'_2 \triangleq x_2 - \mu\delta \quad (51)$$

とおく. このとき, x'_1, x'_2 はそれぞれ,

$$\begin{aligned} x'_1 &= \frac{x_2 - x_1 - \delta + \mu\delta}{x_2 - x_1} x_1 + \frac{(1 - \mu)\delta}{x_2 - x_1} x_2 \\ &= \frac{x_2 - x_1 - (1 - \mu)\delta}{x_2 - x_1} x_1 + \frac{(1 - \mu)\delta}{x_2 - x_1} x_2 \end{aligned} \quad (52)$$

$$\begin{aligned} x'_2 &= \frac{\mu\delta}{x_2 - x_1} x_1 + \frac{x_2 - x_1 - \delta + (1 - \mu)\delta}{x_2 - x_1} x_2 \\ &= \frac{\mu\delta}{x_2 - x_1} x_1 + \frac{x_2 - x_1 - \mu\delta}{x_2 - x_1} x_2 \end{aligned} \quad (53)$$

ここで,

$$\lambda_1 = \frac{(1 - \mu)\delta}{x_2 - x_1} \quad (54)$$

$$\lambda_2 = \frac{\mu\delta}{x_2 - x_1} \quad (55)$$

とおくと,

$$\frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{\mu\delta}{(1 - \mu)\delta + \mu\delta} = \mu \quad (56)$$

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{(1 - \mu)\delta}{(1 - \mu)\delta + \mu\delta} = 1 - \mu \quad (57)$$

よって, 式 (49), (56), (57) より,

$$\mu f(x'_1) + (1 - \mu)f(x'_2) < \mu f(x_1) + (1 - \mu)f(x_2) \quad \square$$

補題 3 条件

$$\sum_{i=1}^N z_i = M_N \quad (58)$$

のもとで, 不等式

$$\sum_{i=1}^j z_i \leq M_j, \quad (j = 1, \dots, N) \quad (59)$$

を満たす N 次元ベクトル $z = (z_1, \dots, z_N)$ の要素のうち,

$$z_1 = \dots = z_{j_1} < z_{j_1+1} = \dots = z_{j_2} < z_{j_2+1} = \dots = z_{j_3} < \dots = z_{j_K} \quad (60)$$

を満たす $j_1, j_2, j_3, \dots, j_K$ を定める.

このとき,

$$z_1 = \dots = z_{j_1} = a \quad (61)$$

$$z_{j_{K-1}+1} = \dots = z_{j_K} = b \quad (62)$$

さらに,

$$\delta_\alpha \triangleq M_{j_\alpha} - \sum_{i=1}^{j_\alpha} z_i, \quad (\alpha = 1, \dots, K-1) \quad (63)$$

$$k = \arg \min_{\alpha} \delta_\alpha \quad (64)$$

と仮定する. このもとで,

$$a' = a + \frac{\delta_k}{n_a} \quad (65)$$

$$b' = b - \frac{\delta_k}{n_b} \quad (66)$$

とすると, 新たなベクトル z' が以下のように表せる.

$$z' = (z'_1, \dots, z'_{j'_1}, z_{j_1+1}, \dots, z_{j_{K-1}}, z'_{j_{K-1}+1}, \dots, z'_{j_K}) \quad (67)$$

このとき,

$$G(z) - G(z') > 0 \quad (68)$$

かつ feasible である.

証明:

$$G(z) - G(z') = \{n_a g(a) + n_b g(b)\} - \{n_a g(a') + n_b g(b')\} \quad (69)$$

$$= (n_a + n_b) \left\{ \frac{n_a}{n_a + n_b} g(a) + \frac{n_b}{n_a + n_b} g(b) \right\} - (n_a + n_b) \left\{ \frac{n_a}{n_a + n_b} g(a') + \frac{n_b}{n_a + n_b} g(b') \right\}$$

$$= (n_a + n_b) \{ \mu g(a) + (1 - \mu) g(b) - \mu g(a') - (1 - \mu) g(b') \} \quad (70)$$

$$\stackrel{\text{(補題 2 より)}}{>} 0 \quad (71)$$

ただし, $\frac{n_a}{n_a + n_b} = \mu$ として補題 2 を適用した.□

付録 B ソースコード

B.1 basic-algorithm のプログラム

```
#include <stdio.h>

#define ARRAYMAX 100

double
arraymin(int size, double v[], int s[])
{
    int i;
    double minimum = -1.0;

    for (i = 0; i < size; i++){
        if(s[i] >= 0){
            if (minimum > v[i] || minimum < 0){
                minimum = v[i];
            }
        }
    }
    return(minimum);
}

void
faridi_base(int size, double r[], double m[], double k[])
{
    int s[ARRAYMAX];
    double u[ARRAYMAX];
    int rest;
    int i, j;
    double q;
    double z, z1, z2;

    for (i = 0; i < size; i++){
        s[i] = i;
    }
    rest = size;

    do {
        j = 0;
        q = 0.0;
        for (i = 0; i < size; i++){
            if (s[i] < 0){
                q += k[i];
            }
        }
    } while (q > 0);
}
```

```

        } else {
            u[i] = (m[i] - q) / (++j);
        }
    }

    z1 = arraymin(size, u, s);
    z2 = arraymin(size, r, s);
    z = (z1 < z2)? z1: z2;

    for (i = 0; i < size; i++){
        if (s[i] >= 0){
            if (r[i] == z){
                k[i] = z;
                s[i] = -1;
                rest--;
            }
        }
    }

    for (i = size - 1; i >= 0; i--){
        if (s[i] >= 0){
            if (u[i] == z){
                for (; i >= 0; i--){
                    if (s[i] >= 0){
                        k[i] = z;
                        s[i] = -1;
                        rest--;
                    }
                }
            }
        }
    }

    } while (rest > 0);

    return;
}

int
main()
{
    double r[5] = {4, 3, 6, 2, 5};
    double m[5] = {2.5, 3, 10, 10, 12};
    double k[5];
    int i;

    faridi_base(5, r, m, k);

    for (i = 0; i < 5; i++){
        printf(" %f", k[i]);
    }

    return(0);
}

```