

信州大学工学部

学士論文

到着期限のある閉ループ伝送システムにおける
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 07T2010H
氏名 岩井 祐斗

2011年3月10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	到着期限とフィードバック	2
2.3	歪み関数	3
3	閉ループシステム	4
3.1	伝送計画	5
3.2	伝送指針	6
4	CEC アルゴリズム	7
4.1	従来法	7
4.2	提案法	8
5	まとめ	8
	謝辞	9
	参考文献	9
付録 A	証明	9
A.1	修正 CEC アルゴリズムの最適性の証明	9
A.2	伝送指針の実行可能性の証明	14
付録 B	フローチャート	16
B.1	従来法 CEC アルゴリズム	16
B.2	提案法 修正 CEC アルゴリズム	17
付録 C	ソースコード	18
C.1	提案法 修正 CEC アルゴリズムのプログラム	18

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そのコンテンツを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。その中で動画の場合、一連の動きをフレームと呼ばれる複数の静止画に分け、それらを順次表示することで動きを再現する。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、画質がきれいで、フレーム落ちがなくスムーズに再生できることが大事になる。

インターネットを介したこのような動画の再生を、ビデオ・オン・デマンドまたはストリーミングと呼ぶ。動画の再生時間とデータをダウンロードしている時間が等しければ、視聴者は動画が途中で止まることなく最後まで視聴できる。そのためには視聴者が望むフレームが、再生される時間までにダウンロードされていなければならない。視聴者が要請するこの期限を、到着期限と呼ぶ。また、伝送路における誤りを送信側で検知できる場合、到着期限までに繰り返しデータを再送信することで、再生されたフレームの劣化を最小限に抑えることができる。しかし、動画全体としてみた場合、フレームひとつがきれいでも、全体をみて画質が悪ければ視聴者は満足しない。そのため、個々のフレームの詳細な部分を表現するデータを犠牲にすることで、動画全体の概略を表現するデータを到着期限までに送信していくことが大切となる。

本研究では、フレームのデータを送る伝送路には雑音があると仮定し、送信された各パケットの受信内容に誤りが含まれているかどうかを送信側で確認することのできる閉ループシステムを考える。送信機は各パケットの受信状況を確認することによってその後の伝送計画を更新する。この意味で送信機はスケジューラとも呼ばれる。各フレームに対して歪みを定義し、その総和を動画全体の歪みとする。本研究の目的は、全体の歪みを小さくするような伝送手順を見つけることである。

このような問題に対して、従来研究 [1] では確実性等価制御を利用して伝送計画を立てることが提案されている。しかし、提案されたアルゴリズムは到着期限を守らない場合がある。本論文では、従来研究 [1] に沿って問題を説明した後、確実性等価制御に基づく最適な伝送計画を導くアルゴリズムを提案する。

2 伝送システムの定式化

2.1 伝送モデル

本研究の伝送システムの概略図を図 1 に示す。本研究の伝送システムの特徴は、到着期限が設けられていることと、到着期限を守るためにフレームに歪みが生じるのを許容していること

である．以下にそれらを定式化する．また，本論文に登場する主な記号の意味を表 1 にまとめる．

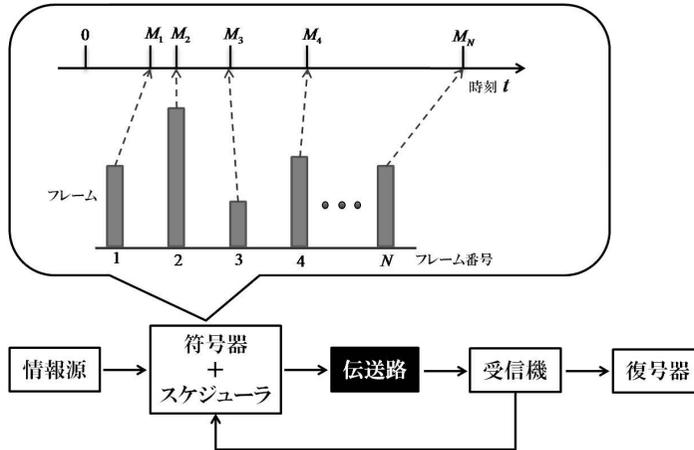


図 1 伝送モデル

表 1 本研究で使われている記号の説明

記号	説明
N	フレームの総数
M_i	フレーム i の到着期限
p	データの伝送成功確率
$d(\cdot)$	歪み関数 (単調減少で狭義に下に凸)
n	伝送アルゴリズム中, 到着期限が残っているフレームの数
b_i	フレーム i に対して, すでに復号器に届いているデータ量
m_i	フレーム i の到着期限になるまでの残り時間
y_i	フレーム i に対する送信データ量 (伝送計画)
B_i	フレーム i に対して, 最終的に届くデータ量 (確率変数)
x_i	フレーム i に対して, 最終的に届くデータ量の期待値

2.2 到着期限とフィードバック

フレームの総数を N とし, それらは最初すべて情報源に存在するとする．各フレームに対して対応する到着期限 M_i ($i = 1, \dots, N$) がある．本研究では問題を一般化し, 到着期限が等間隔であるとは限らない．一般性を損なうことなく, フレームの並び方は到着期限が切れる順とする．すなわち, $M_i \leq M_{i+1}$, $i \in \{1, \dots, N-1\}$ である．

到着期限が過ぎたフレームに対して、スケジューラはパケットを送信しない。スケジューラは到着期限前のフレームの中からパケットを送信するフレームを選ぶが、送信された各パケットは独立同一分布の確率 p で復号器に届く。パケットの大きさと送信にかかる時間はすべて同一とする。パケットが無事に届いたかどうかはスケジューラにフィードバックされる。

2.3 歪み関数

復号器で復号されるフレームと元のフレームは完全には一致しない。復号器がより多くのデータを受け取れば、復号されたフレームは元のフレームに近づくと考えるのが自然である。本稿では従来研究 [1] に従い、歪みを受け取ったデータ量に関して単調減少で狭義に下に凸な関数であると仮定する (図 2)。単調減少の妥当性は次のように説明できる。送信するデータが画像であったとすると、データ量が多くなるほど画質はよくなる。つまり元の画像により近づくため歪みが小さくなると考えられる。このとき、データ量が大きくなる途中で画質が悪くなるということは起こりにくい。そのため、単調減少になると仮定する。凸性の妥当性は画質の悪い画像と綺麗な画像に同じ量のデータを追加したとすると、その画質を両方ともよくすることができる。しかし悪い画像のほうが画質の改善の程度の変化が大きいと考えられる。そのため、下に凸であるということ仮定する。

届いたデータ量が x のフレームに対し、その歪みを $d(x)$ と表す。

2.3.1 狭義に下に凸な減少関数の性質

狭義に下に凸な減少関数には補題 1 のような性質がある。

補題 1 $d(\cdot)$ が狭義に下に凸な減少関数であったとき、 $a, b, \delta \in \mathbb{R}$, そして $0 < \delta < b - a$ であるとき、次式が成立する。(図 2)

$$d(a + \delta) + d(b - \delta) < d(a) + d(b) \quad (1)$$

証明:

関数が狭義に下に凸であるとは

$$d((1 - \lambda)a + \lambda b) < (1 - \lambda)d(a) + \lambda d(b) \quad (2)$$

を満たすことである。ここで λ は $0 < \lambda < 1$ となる任意の定数である。そのため

$$d(\lambda a + (1 - \lambda)b) < \lambda d(a) + (1 - \lambda)d(b) \quad (3)$$

も成立する。式 (2)(3) を足すと、

$$d(a + (b - a)\lambda) + d(\lambda(a - b) + b) < d(a) + d(b) \quad (4)$$

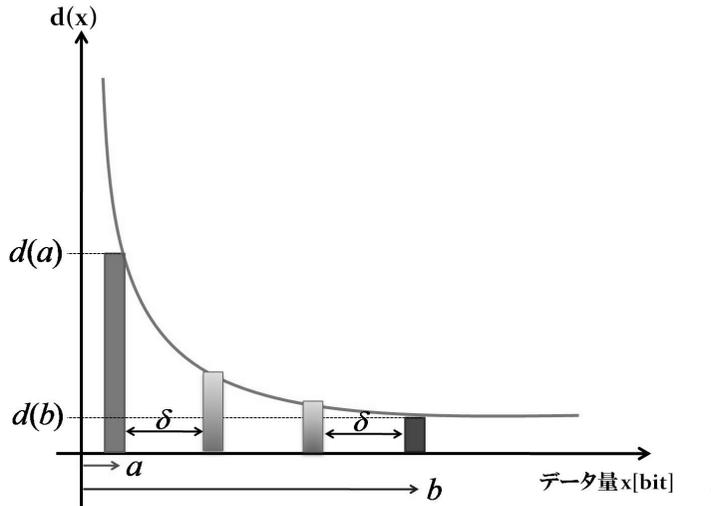


図2 歪み関数 (狭義に下に凸な減少関数)

となる．ここで $\lambda = \frac{\delta}{b-a}$ とおくと，式 (4) は

$$d(a + \delta) + d(b - \delta) < d(a) + d(b) \quad (5)$$

となり，補題 1 が証明された．□

3 閉ループシステム

この節では，フィードバックのある図 1 の伝送モデルを用いて，最終的な全体の歪みを小さくするような伝送を行う方法を考える．伝送路には雑音があるため，データを誤りなくどれだけ送信できるかわからない．そのため事前に最適な伝送計画を立てることはできない．そこで従来研究 [1] では，確実性等価制御 (Certainty Equivalent Controllers, CEC)[2] の考えを用いて伝送計画を随時更新することを提案している．CEC は，観測していないランダムな値の期待値を求め，各段階で最適解を毎回計算する発見的手法である．

本研究の伝送モデルに CEC の手法を用いると，システムの動きは以下ようになる．まず，スケジューラは現在の時点で現状に基づき，歪みの期待値が最小となるように現在以降の伝送計画を立てる．これをもとにフレームを選んでパケットを送信する．次の時刻では，送信したパケットが無事に復号器に届いたかどうかに基づき，スケジューラは把握している現状を更新する．また，時刻が進んでいることから，各フレームの到着期限も更新する．もし到着期限が残っているフレームがあれば，それらに対して再び伝送計画を立てる．これをすべての (最後の) フレームの到着期限が切れるまで繰り返す．

1. 現状に対して最適な伝送計画を見つける .
2. 伝送計画をもとにパケットを送信するフレームを決定する .
3. パケットを送信し , 受信状況を受け取る .
4. 現状を更新する .
5. 到着期限が切れたフレームは対象から取り除く .
6. 到着期限が残っているフレームがあるなら手順 (1) に戻る .
7. 終了 .

図 3 伝送アルゴリズムの概略

この手順をまとめたのが , 図 3 の伝送アルゴリズムである . 現状に対する伝送計画と , パケットを送信するフレームの選び方 (伝送指針) については次節で詳しく説明する . また , 伝送計画を見つけるにあたって , 到着期限が 0 以下となったフレームにはデータを送信することはできないため考える必要はない [1]. しかし , データはパケット単位で送信されるため , 到着期限が 1 パケットを送信する時間に満たない場合 , そのフレームにデータを送信することはできない . そのため , 伝送アルゴリズムの概略の手順 5 の到着期限が切れたフレームとは , 到着期限が 1 パケットを送信する時間に満たないフレームを指す .

3.1 伝送計画

各パケットに対し , すでに届いているデータ量と到着期限までの残り時間をそれぞれ $\mathbf{b} = (b_1, \dots, b_n)$, $\mathbf{m} = (m_1, \dots, m_n)$ とする . これらはスケジューラが把握している現状を表しており , 初期値は $b_i = 0$, $m_i = M_i$, $n = N$ である . 現時点での伝送計画を $\mathbf{y} = (y_1, \dots, y_n)$ と表す .

本来のシステムの目的からすれば , 計画通りに送信する場合に最終的に届くデータ量を確率変数 B_i とおいて , 現状における歪みの条件付き期待値

$$\mathbb{E} \left[\sum_{i=1}^n d(B_i) \middle| \mathbf{b}, \mathbf{y} \right] \quad (6)$$

を最小化する伝送計画 \mathbf{y} を求めなければならない . しかし , このような伝送計画を見つけることは計算量的に高価となる . そこで , (6) の代わりに B_i の条件付き期待値を用いて

$$\sum_{i=1}^n d(\mathbb{E}[B_i | b_i, y_i]) \quad (7)$$

を最小化する伝送計画 \mathbf{y} を用いることにする . 歪み関数の性質を利用すると , (7) を最小化するために歪み関数 $d(\cdot)$ の値を評価する必要がない . したがって計算量的に非常に安価に計算することができるのが利点である . また , (7) を最小化する伝送計画を用いたときの最終的な全体の歪みは , (6) を用いたときの歪みに匹敵することが実験的に示されている [1] .

B_i の条件付き期待値を x_i と表すと

$$x_i = \mathbb{E}[B_i|b_i, y_i] = b_i + py_i \quad (8)$$

が成り立つ．このことから $x = (x_1, \dots, x_n)$ のことも伝送計画と呼ぶ．負のデータ量を送ることはできないので

$$x_i \geq b_i \quad (i = 1, \dots, n) \quad (9)$$

が成り立たなければならない．また，データは到着期限の近い順に送信し，到着期限を越えてデータを送信しないので

$$\sum_{j=1}^i y_j \leq m_i \quad (10)$$

でなくてはならない．ただし，単位時間あたりに送れるデータ量をデータの単位とした．(10) は x_i を用いると

$$\sum_{j=1}^i x_j \leq \sum_{j=1}^i b_j + pm_i \quad (i = 1, \dots, n) \quad (11)$$

と表される．(9) と (11) を満たす伝送計画を実行可能であるという．

(7) より，最適な伝送計画を求める最適化問題を次のように定式化する．

\mathbf{P}_{CEC} :

$$\sum_{i=1}^n d(x_i) \quad (12)$$

を最小化する実行可能な伝送計画 x を求めよ．

3.2 伝送指針

\mathbf{P}_{CEC} の解から (8) によって計算した最適な伝送計画 y は今後どれだけ各フレームにデータを送信すれば歪みを最小化できるかを示しているのであって，今からどのフレームのペケットを送信すればいいかは示していない．伝送計画が導くデータ量は連続量であり，ペケットの大きさは固定されているからである．そのため，ペケットを送信するフレームを決める伝送指針が必要となる．

伝送指針は，ペケットが無事に届いたとしても次の時刻に

$$b_1 \geq \dots \geq b_n \quad (13)$$

が成り立つようにフレームを選択する．この選択の仕方によって，到着期限前の2つのフレームの間の最適性に損失は発生しない．なぜなら，最初のフレームの到着期限は，後ろのフレー

入力: $p, n, \mathbf{b}, \mathbf{m}$

出力: \mathbf{x}

1. $C_j \leftarrow \sum_{i=1}^j b_i + pm_j \ (j = 1, \dots, n)$
2. $l \leftarrow 1$
3. $\mu_j \leftarrow \frac{C_j}{j-l+1} \ (j = l, \dots, n)$
4. $\hat{j} \leftarrow \max \{ \arg \min_{l \leq j \leq n} \{ \mu_j \} \}$
5. $\mu_j(k) \leftarrow \frac{C_j - \sum_{i=l}^k b_i}{j-k} \ (k = l-1, \dots, n)$
6. $\hat{k} \leftarrow \min \{ k \in \{l-1, \dots, N\} | b_{k+1} < \mu_j(k) \}$
7. $x_k \leftarrow \begin{cases} b_k, & l \leq k \leq \hat{k} \\ \mu_j(\hat{k}), & \hat{k} < k \leq \hat{j} \end{cases}$
8. もし $\hat{j} < N$ ならば $l \leftarrow \hat{j} + 1, C_i \leftarrow C_i - C_{\hat{j}}$ とし, 手順 (3) に戻る.
9. 終了.

図 4 CEC アルゴリズム

μ の到着期限より前のため, 始めのフレームよりも後ろのフレームにデータを多く送っても歪みを小さくすることはできないからである. (13) は以降の証明の中で断りなく使われる.

このような伝送指針 ψ の例として次がある [1].

$$\psi(\mathbf{y}) = \min \left\{ \arg \max_{1 \leq i \leq j^*} \{ y_i \} \right\} \quad (14)$$

$$j^* = \min \left\{ j \mid \sum_{i=1}^j y_i \geq 1 \right\} \quad (15)$$

この伝送指針が選んだフレームは, データが送信された後でも, (13) を満たすことを付録で証明する.

4 CEC アルゴリズム

P_{CEC} を解く試みとして CEC アルゴリズム [1] がある. 以下では CEC アルゴリズムの反例を示し, P_{CEC} の最適解を与える修正 CEC アルゴリズムを提案する.

4.1 従来法

CEC アルゴリズムを図 4 に示す. CEC アルゴリズムは実行可能でない伝送計画を導く場合がある. その反例は $p = 1/2, n = 4, \mathbf{b} = \{2, 0, 0, 0\}, \mathbf{m} = \{2, 2, 2, 4\}$ である. これに対する CEC アルゴリズムの出力は $\mathbf{x} = \{2, 2/3, 2/3, 2/3\}$ となる. このとき $\sum_{i=1}^3 x_i = 10/3 > \sum_{i=1}^3 b_i + pm_3 = 3$ となり (11) を満たさないので実行可能ではない.

このように, CEC アルゴリズムは未来の時刻に到着期限を越えるような伝送計画を導く場合がある. しかし, パケットを送信するにあたって到着期限に間に合うフレームを選択するのは伝送指針の責任なので, 伝送指針が正しければ送信されたパケットが到着期限を越えること

入力: $p, n, \mathbf{b}, \mathbf{m}$

出力: \mathbf{x}

1. $C_j \leftarrow \sum_{i=1}^j b_i + pm_j$ ($j = 1, \dots, n$)
2. $l \leftarrow 1$
3. $\mu_j \leftarrow \frac{C_j - \sum_{i=1}^{l-1} x_i}{j-l+1}$ ($j = 1, \dots, n$)
4. $\hat{j} \leftarrow \max \{ \arg \min_{l \leq j \leq N} \{ \mu_j \} \}$
5. もし $b_l > \mu_{\hat{j}}$ ならば (5.1), そうでないなら (5.2) を実行する.
(5.1) $x_l \leftarrow b_l$ として, $l \leftarrow l+1$
(5.2) $x_i \leftarrow \mu_{\hat{j}}$ ($i = l, \dots, \hat{j}$) として, $l \leftarrow \hat{j}+1$
6. もし $l \leq N$ であったら, 手順 (3) に戻る.
7. 終了.

図 5 修正 CEC アルゴリズム

はない。また、よく似た伝送計画に対しては、伝送指針は同じフレームを選択すると考えられるので、最終的な歪みを小さくするという目的から大きく外れることはないと考えられる。

4.2 提案法

本稿で提案する修正 CEC アルゴリズムを図 5 に示す。このアルゴリズムは P_{CEC} の最適解を与える。最適性の証明は付録に記載する。

定理 1 単調減少で狭義に下に凸な歪み関数 $d(\cdot)$ のもとで、修正 CEC アルゴリズムの出力は P_{CEC} の最適解である。

5 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を、狭義に下に凸な減少関数である歪みとした。そして、歪みを最小とするためのアルゴリズムを提案した。

情報源から送られる N 個のフレームに対する送信データ量を、 N 次元ベクトルを用いて表し、これを伝送計画と呼んだ。受け手が要請する時間を到着期限とし、伝送計画があらかじめ決められた到着期限を守る場合を実行可能であるとした。従来研究 [1] では伝送路に誤りのあるなかで、歪みを最小とする最適な伝送計画を導くアルゴリズムを紹介されている。しかしこのアルゴリズムには、実行可能でない伝送計画を立ててしまう場合がある。本研究では、常に実行可能な伝送計画を立てることができるアルゴリズムを提案し、その最適性を証明した。今後の課題として、送信データ量の上限を考えた場合や、フィードバックに遅れや誤りを考慮した場合など、より一般的な条件のもとで最適な伝送計画を導くアルゴリズムを考える課題が

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へ感謝の意を表す。

参考文献

- [1] Azadeh Faridi, Anthony Ephremides, "Distortion Control For Delay-sensitive Sources" IEEE Transaction on information, Vol.54, No.8 pp.3399-3406 August 2008.
- [2] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scentific, 2005, vol,I.

付録 A 証明

ここでは，証明をわかりやすくするため次のような用語を導入する．修正 CEC アルゴリズムの手順 (5) で $b_l > \mu_j$ が成り立ち，手順 (5.1) で $x_l \leftarrow b_l$ が実行されたフレーム l を「リッチである」という．また手順 (5) で， $b_l \leq \mu_j$ となり，手順 (5.2) で $x_i \leftarrow \mu_j$ が実行された一連のフレーム $i = l, \dots, \hat{j}$ に対し，それらは「同じグループに属している」という．

A.1 修正 CEC アルゴリズムの最適性の証明

図 5 の修正 CEC アルゴリズムの最適性を証明する．

補題 2 修正 CEC アルゴリズムの出力 x は (9) を満たす．

証明: リッチなフレーム j に対しては $x_i = b_i$ となる．グループに属するフレーム j に対してはグループの先頭と末尾のフレーム番号をそれぞれ l, \hat{j} とおくと $x_j = \mu_j \geq b_l \geq b_j$ となる．

□

補題 3 修正 CEC アルゴリズムの結果にしたがってフレームの並びを見ると，ひとつのグループの後ろには別のグループが続く．

証明:

手順 (5) で $b_l \leq \mu_j$ が成立しているときの l, \hat{j} をそれぞれ l_1, \hat{j}_1 とおく．このとき，

$$x_i = \mu_{\hat{j}_1}, \quad i = l_1, \dots, \hat{j}_1$$

となる．さらに， $l_2 = \hat{j}_1 + 1$ として

$$\hat{j}_2 \triangleq \max \left\{ \arg \min_{l_2 \leq j \leq n} \left\{ \frac{C_j - \sum_{i=1}^{l_2-1} x_i}{j - l_2 + 1} \right\} \right\} \quad (16)$$

$$\mu_{\hat{j}_2} \triangleq \frac{C_{\hat{j}_2} - \sum_{i=1}^{l_2-1} x_i}{\hat{j}_2 - l_2 + 1} \quad (17)$$

とする．このとき $\hat{j}_1 < \hat{j}_2$ であるから

$$\mu_{\hat{j}_1} < \frac{C_{\hat{j}_2} - \sum_{i=1}^{l_1-1} x_i}{\hat{j}_2 - l_1 + 1} \quad (18)$$

$$= \frac{C_{\hat{j}_2} - \left(\sum_{i=1}^{l_2-1} x_i - \sum_{i=l_1}^{l_2-1} x_i \right)}{\hat{j}_2 - l_1 + 1} \quad (19)$$

$$= \frac{C_{\hat{j}_2} - \sum_{i=1}^{l_2-1} x_i + (l_2 - l_1)\mu_{\hat{j}_1}}{\hat{j}_2 - l_1 + 1} \quad (20)$$

となり， $\mu_{\hat{j}_1}$ について整理すると

$$\mu_{\hat{j}_1} < \frac{C_{\hat{j}_2} - \sum_{i=1}^{l_2-1} x_i}{\hat{j}_2 - l_2 + 1} = \mu_{\hat{j}_2} \quad (21)$$

を得る．よって $b_{l_2} \leq b_{l_1} \leq \mu_{\hat{j}_1}$ であることから $b_{l_2} < \mu_{\hat{j}_2}$ となる．

以上より， $b_l \leq \mu_{\hat{j}}$ が成立したとき，次のループでも， $b_l \leq \mu_{\hat{j}}$ が成立する．□

補題 4 修正 CEC アルゴリズムの出力は (11) を満たす．

証明:

まず，フレーム j をリッチであるとする．このとき補題 3 より，1 から j まですべてリッチなので

$$\sum_{i=1}^j x_i = \sum_{i=1}^j b_i \leq C_j \quad (22)$$

となる．

次に， j がグループに属しているとする．グループの先頭と末尾をそれぞれ l, \hat{j} とおくと

$$\mu_{\hat{j}} = \min_{l \leq k \leq n} \frac{C_k - \sum_{i=1}^{l-1} x_i}{k - l + 1} \leq \frac{C_j - \sum_{i=1}^{l-1} x_i}{j - l + 1} \quad (23)$$

が成り立つので

$$\sum_{i=1}^j x_i = \sum_{i=1}^{l-1} x_i + \sum_{i=l}^j x_i \quad (24)$$

$$= \sum_{i=1}^{l-1} x_i + (j-l+1)\mu_{\hat{j}} \quad (25)$$

$$\leq \sum_{i=1}^{l-1} x_i + C_j - \sum_{i=1}^{l-1} x_i = C_j \quad (26)$$

となる .

以上より , 修正 CEC アルゴリズムの出力 x は (11) を満たす . \square

補題 5 フレーム n はグループに属する .

証明:

背理法の仮定として , フレーム n がリッチであったとする . このとき補題 3 より , すべてのフレームはリッチである . すなわち $x_i = b_i, i = 1, \dots, n$ となる . また , 最後のループで $l = \hat{j} = n$ となっていなければならない . すると

$$b_n > \mu_n = \frac{C_n - \sum_{i=1}^{n-1} x_i}{n - n + 1} = C_n - \sum_{i=1}^{n-1} b_i \quad (27)$$

すなわち $\sum_{i=1}^n b_i > C_n$ となり , (11) に矛盾する . よって , 最後のフレームはグループに属する . \square

補題 6 グループ内の最後のフレーム \hat{j} に対し $\sum_{i=1}^{\hat{j}} x_i = C_{\hat{j}}$ が成り立つ .

証明:

グループの先頭のフレーム番号を l とすると

$$x_i = \mu_{\hat{j}} = \frac{C_{\hat{j}} - \sum_{i=1}^{l-1} x_i}{\hat{j} - l + 1} \quad (i = l, \dots, \hat{j}) \quad (28)$$

が成り立つ . よって

$$\sum_{i=1}^{\hat{j}} x_i = \sum_{i=1}^{l-1} x_i + \sum_{i=l}^{\hat{j}} x_i \quad (29)$$

$$= \sum_{i=1}^{l-1} x_i + (\hat{j} - l + 1)\mu_{\hat{j}} \quad (30)$$

$$= \sum_{i=1}^{l-1} x_i + C_{\hat{j}} - \sum_{i=1}^{l-1} x_i = C_{\hat{j}} \quad (31)$$

となる . \square

補題 7 フレーム \hat{i} がリッチであり、フレーム i が最初のグループに属するならば、 $x_{\hat{i}} > x_i$ である。

証明: 最後のリッチなフレームの番号を l_1 とおく。このとき

$$x_{\hat{i}} = b_{\hat{i}} \geq b_{l_1} = x_{l_1} \quad (32)$$

である。ここで

$$\hat{j}_1 \triangleq \max \left\{ \arg \min_{l_1 \leq j \leq n} \left\{ \frac{C_j - \sum_{i=1}^{l_1-1} x_i}{j - l_1 + 1} \right\} \right\} \quad (33)$$

とおく。すると $\hat{j}_1 > l_1$ となる。なぜなら、もし $\hat{j}_1 = l_1$ ならば、 l_1 がリッチであることから、

$$b_{l_1} > \mu_{\hat{j}_1} = \frac{C_{\hat{j}_1} - \sum_{i=1}^{l_1-1} x_i}{\hat{j}_1 - l_1 + 1} = C_{l_1} - \sum_{i=1}^{l_1-1} b_i \quad (34)$$

より $\sum_{i=1}^{l_1} b_i > C_{l_1}$ となって (11) に矛盾するからである。さらに $l_2 \triangleq l_1 + 1$ とし

$$\hat{j}_2 \triangleq \max \left\{ \arg \min_{l_2 \leq j \leq n} \left\{ \frac{C_j - \sum_{i=1}^{l_2-1} x_i}{j - l_2 + 1} \right\} \right\} \quad (35)$$

とする。 l_2 は最初のグループに属し、 $\hat{j}_1 \geq l_2$ であるため

$$x_i = x_{l_2} = \mu_{\hat{j}_2} \quad (36)$$

$$= \min_{l_2 \leq j \leq n} \left\{ \frac{C_j - \sum_{i=1}^{l_2-1} x_i}{j - l_2 + 1} \right\} \quad (37)$$

$$\leq \frac{C_{\hat{j}_1} - \sum_{i=1}^{l_2-1} x_i}{\hat{j}_1 - l_2 + 1} \quad (38)$$

$$= \frac{C_{\hat{j}_1} - \sum_{i=1}^{l_1-1} x_i - b_{l_1}}{\hat{j}_1 - l_1} \quad (39)$$

$$< \frac{1}{\hat{j}_1 - l_1} \left(1 - \frac{1}{\hat{j}_1 - l_1 + 1} \right) \left(C_{\hat{j}_1} - \sum_{i=1}^{l_1-1} x_i \right) \quad (40)$$

$$= \frac{C_{\hat{j}_1} - \sum_{i=1}^{l_1-1} x_i}{\hat{j}_1 - l_1 + 1} \quad (41)$$

$$= \mu_{\hat{j}_1} < b_{l_1} = x_{l_1} \leq x_i \quad (42)$$

となる。□

定理 1 単調減少で狭義に下に凸な歪み関数 $d(\cdot)$ のもとで、修正 CEC アルゴリズムの出力は P_{CEC} の最適解である。

証明: 補題 2 と補題 3 より、修正 CEC アルゴリズムの出力は実行可能である。最適性を以下に示す。

背理法の仮定として、修正 CEC アルゴリズムの出力 x が最適解でないとする。最適解を x' とおく。これは $\sum_{i=1}^n d(x'_i) < \sum_{i=1}^n d(x_i)$ を意味する。したがってこのとき、あるフレーム \hat{i} に対して $d(x'_i) < d(x_i)$ すなわち $x'_i > x_i$ となっているはずである。フレーム \hat{i} は、リッチであるかまたはグループに属するかどうかである。

まず、フレーム \hat{i} がリッチであるとする。このとき、最初のグループの中のある \tilde{i} に対して、 $x'_i < x_i$ が成立する。なぜなら、さもなければ最初のグループの末尾 \hat{j} で $\sum_{i=1}^{\hat{j}} x'_i > \sum_{i=1}^{\hat{j}} x_i$ となるが、補題 6 より $\sum_{i=1}^{\hat{j}} x_i = C_{\hat{j}}$ であるので x' が実行可能であることに矛盾する。リッチなフレーム i に対して $x'_i \geq x_i = b_i$ でなければならないことに注意されたい。また補題 7 より $x_i > x_i$ である。したがって $x'_i > x_i > x_i > x'_i$ が成立する。ここで

$$\delta \triangleq \min \{x'_i - x_i, x_i - x'_i\} > 0 \quad (43)$$

を用いて x'' を

$$x''_i \triangleq \begin{cases} x'_i - \delta, & i = \hat{i} \\ x'_i + \delta, & i = \tilde{i} \\ x'_i, & \text{otherwise} \end{cases} \quad (44)$$

のように定義する。すると補題 1 より

$$\sum_{i=1}^n d(x''_i) < \sum_{i=1}^n d(x'_i)$$

となるため x' の最適性に矛盾する。

次に、フレーム \hat{i} がグループに属しているとする。補題 6 より、このとき同じグループのあるフレーム \tilde{i} ($\neq \hat{i}$) に対して $x'_i < x_i$ となる。よって $x'_i > x_i = x_i > x'_i$ が成立する。ここで (43), (44) のように x'' を定義すれば矛盾が導かれる。

以上より、 x は最適解である。□

A.2 伝送指針の実行可能性の証明

1 パケット送信するのに 1 秒かかるとする．これより，図 3 の伝送アルゴリズムの手順 5 「到着期限がきれたフレームは対象から取り除く」を数式を用いて表すと，

$$u = \min\{i | m_i \geq 1\} \quad (45)$$

となるフレーム番号 u より小さいフレームは取り除く，となる．よって，今後使われる m_i はすべて 1 以上となる．

定理 2 伝送指針 [1]

$$\psi(\mathbf{y}) = \min \left\{ \arg \max_{1 \leq i \leq j^*} \{y_i\} \right\} \quad (46)$$

$$j^* = \min \left\{ j \left| \sum_{i=1}^j y_i \geq 1 \right. \right\} \quad (47)$$

によって決められるフレームは，データが送信された後でも

$$b_1 \geq \dots \geq b_n \quad (48)$$

が成立する．

証明: ここで時刻 t の時点で，各フレームに届いているデータ量を b ，時刻 $t+1$ の時点で，各フレームに届いているデータ量を b' とする．ただし， b' は伝送アルゴリズムの手順 5 で到着期限の切れたフレームを取り除く前の値である．また， b, b' は整数ベクトルである．

これらをもとに，数学的帰納法によって証明していく．まず， $t=0$ の時， $b = (0, \dots, 0)$ であるから，すべての i に対して (48) が成立する．次に， b が，(48) を満たしていると仮定する．

時刻 t でのデータ伝送に失敗した場合，

$$b' = b \quad (49)$$

となり (48) が成立する．

次に，時刻 t でのデータ伝送に成功した場合，

$$b'_i = \begin{cases} b_i + 1 & (i = \psi(\mathbf{y})) \\ b_i & (otherwise) \end{cases} \quad (50)$$

となる．この式が，(48) を満たすことを示すには，

$$b_{\psi(\mathbf{y})-1} \geq b_{\psi(\mathbf{y})} + 1 \quad (51)$$

となることを示せばよい．ここで修正 CEC アルゴリズムの結果に対して，最初のグループの先頭と末尾を l, \hat{j}_1 とおく．補題 6 より，

$$\sum_{i=1}^{\hat{j}_1} y_i = \frac{\sum_{i=1}^{\hat{j}_1} x_i - \sum_{i=1}^{\hat{j}_1} b_i}{p} \quad (52)$$

$$= \frac{C_{\hat{j}_1} - \sum_{i=1}^{\hat{j}_1} b_i}{p} \quad (53)$$

$$= \frac{pm_{\hat{j}_1}}{p} = m_{\hat{j}_1} \geq 1 \quad (54)$$

となるため，(47) より， $j^* \leq \hat{j}_1$ となる．したがって，アルゴリズムの結果を y_i を用いて表すと，

$$y_i = \begin{cases} 0 & (1 \leq i \leq l-1) \\ \frac{\mu_{\hat{j}_1} - b_i}{p} & (l \leq i \leq j^*) \end{cases} \quad (55)$$

となる．これより，(46) は

$$\psi(\mathbf{y}) = \min \left\{ \arg \min_{l \leq i \leq j^*} \{b_i\} \right\} \quad (56)$$

となる．ここで $\psi(\mathbf{y})$ について場合分けをおこなう．まず $\psi(\mathbf{y}) = l$ の場合，補題 7 より，

$$b_{\psi(\mathbf{y})} = b_l \leq \mu_{\hat{j}_1} = x_l < x_{l-1} = b_{l-1} = b_{\psi(\mathbf{y})-1} \quad (57)$$

となる． b は整数ベクトルであるため (51) が成立する．

次に， $\psi(\mathbf{y}) \geq l+1$ のとき， $l \leq \psi(\mathbf{y}) - 1$ より，

$$b_{\psi(\mathbf{y})} = \min_{l \leq i \leq j^*} \{b_i\} < b_{\psi(\mathbf{y})-1} \quad (58)$$

となる． b は整数ベクトルであるため (51) は成立する．これらより，時刻 t でのデータ伝送に成功した場合でも (48) は成立する．□

付録 B フローチャート

CEC アルゴリズム [1] のフローチャートと、本論文で提案した修正 CEC アルゴリズムのフローチャートを示す。

B.1 従来法 CEC アルゴリズム

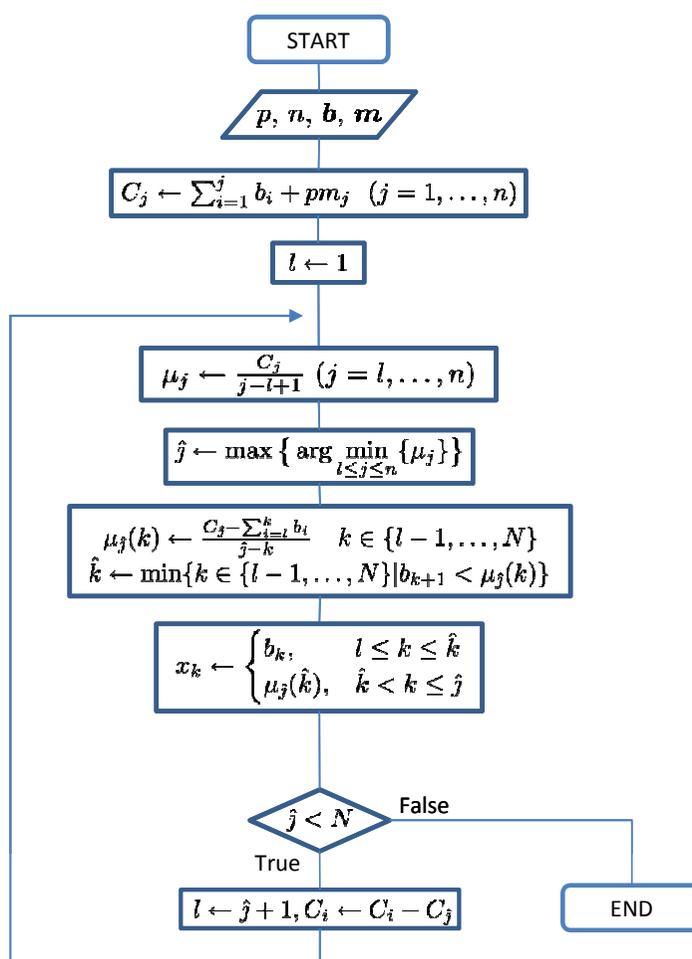


図 6 CEC アルゴリズムのフローチャート

B.2 提案法 修正 CEC アルゴリズム

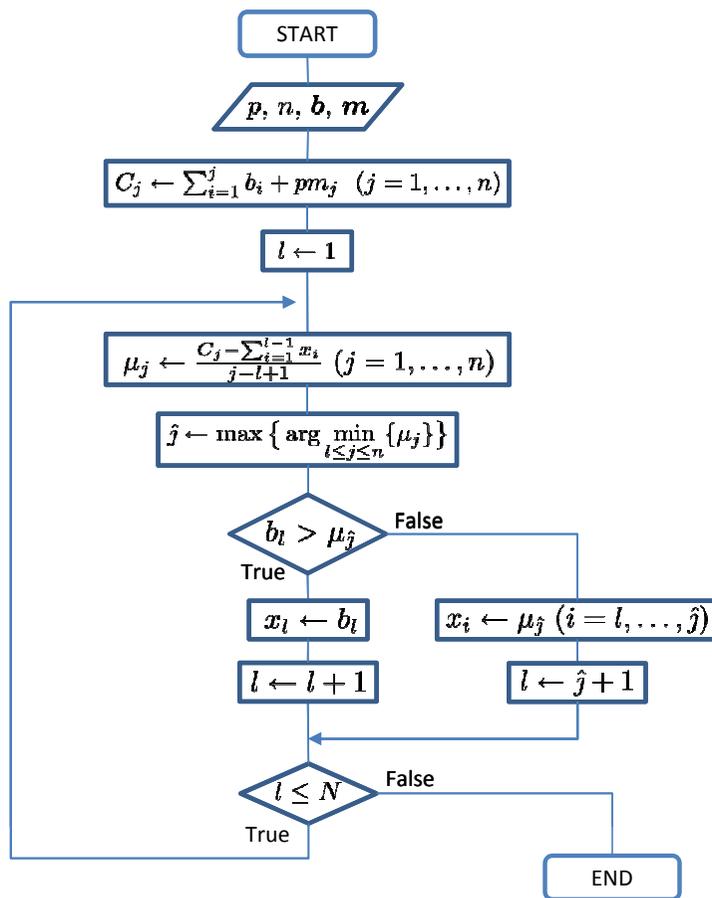


図7 修正 CEC アルゴリズムのフローチャート

付録 C ソースコード

ここでは、4章で提案したアルゴリズムのプログラムを記載する。

C.1 提案法 修正 CEC アルゴリズムのプログラム

```
#include <stdio.h>

#define FRAMEBOUND 100

int n;
double b[FRAMEBOUND];
double pm[FRAMEBOUND];
double x[FRAMEBOUND];

void
cec_proc(void)
{
    double c[FRAMEBOUND];
    double b_cum;
    double x_cum;
    double mu_min;
    int ell;
    int i;
    int j;

    for (b_cum = 0, i = 0; i < n; i++){
        b_cum += b[i];
        c[i] = b_cum + pm[i];
    }

    ell = 0;
    x_cum = 0;

    while (ell < n){

        double mu[FRAMEBOUND];

        for (i = ell; i < n; i++){
            mu[i] = (c[i] - x_cum) / (i - ell + 1);
        }

        mu_min = mu[ell];
        j = ell;

        for (i = ell + 1; i < n; i++){

            if (mu[i] <= mu_min){
                mu_min = mu[i];
                j = i;
            }
        }

        if (b[ell] > mu[j]){
            x[ell] = b[ell];
            x_cum += x[ell];
            ell++;
        } else {
            for (i = ell; i <= j; i++){
                x[i] = mu[j];
                x_cum += x[i];
            }
            ell = j + 1;
        }
    }
}

return;
}

int
main()
```

```
{
    int i;
    scanf("%d", &n);

    if (n <= 0 || n > FRAMEBOUND){
        goto ERROR;
    }

    for (i = 0; i < n; i++){
        scanf("%lf", b + i);
    }

    for (i = 0; i < n; i++){
        scanf("%lf", pm + i);
    }

    cec_proc();

    for (i = 0; i < n; i++){
        printf(" %f", x[i]);
    }
    printf("\n");

return(0);
ERROR:
return(-1);
}
```