The 32nd Symposium on Information Theory
and its Applications (SITA2009)
Yuda-Onsen, Yamaguchi, Japan, Dec. 1–4, 2009

分枝カット法に基づいた線形符号の復号法に関する一考察

A Note on the Branch-and-Cut Approach to Decoding Linear Block Codes

堀井　俊佑* 松嶋　敏泰 * 平澤　茂一 *
Shunsuke Horii Toshiyasu Matsushima Shigeichi Hirasawa

Abstract— Maximum likelihood (ML) decoding of linear block codes can be considered as an integer linear programming (ILP). Since it is an NP-hard problem in general, there are many researches about the algorithms to approximately solve the problem. One of the most popular algorithms is linear programming (LP) decoding. Advanced algorithms for solving ILP (approximately or exactly) include cutting-plane method and branch-and-bound method. As applications of these methods, adaptive LP decoding and branch-and-bound decoding have been proposed.

Another method for solving ILP is the branch-and-cut method, which is a hybrid of cutting-plane and branch-and-bound methods. In this paper, we describe the branch-and-cut based ML decoding algorithm. We construct a generalized framework for the branch-and-cut based ML decoding and compare some algorithms with numerical simulations.

Keywords— Branch-and-Cut method, Maximum Likelihood Decoding, Linear Programming Decoding

1　はじめに

本研究では，無記憶通信路に対する 2 元線形符号の最尤復号の問題を扱う．一般的にこれは NP 困難な問題である為，近似的に最尤復号を行うアルゴリズムに関する研究が数多く行われている．特に確率伝搬法が LDPC 符号に対する復号法として幅広く用いられている．一方で，近年，最尤復号の近似として，線形計画法に基づいて最尤復号を近似的に行う LP 復号法が注目を浴びている [2]．最尤復号の問題は，整数計画問題と見る事ができ，LP 復号では，整数制約を緩和することで得られる線形計画法を解く事で，最尤復号の近似解を得る．LP 復号のアルゴリズムは必ず収束し，整数解に収束した場合は必ず最尤復号の結果と一致することが知られている．この性質は ML Certificate property と呼ばれている．LP 復号法は，このような望ましい性質を持っているが，時間計算量は確率伝搬法に比べて多く，また復号の性能 (復号誤り率) も確率伝搬法に劣ることが多い．その為，LP 復号法の計算量を削減する研究や，復号性能を向上させる研究が数多くされている．

LP 復号法の計算量を削減した復号法の 1 つとして，適応的線形計画 (ALP:Adaptive Linear Programming) 復号法が挙げられる [5]．ALP 復号法では，まず最適化変数に対して 0 以上 1 以下という制約のみを加えた線形計画問題を解き，その最適解が，パリティ検査行列の各行から生成される不等式を満たすかどうかを確認する．満たされない不等式が存在するとき，この不等式はカットもしくは切除平面と呼ばれる．ALP 復号法では，カットが見つかった場合，これを最適化問題に制約として加え，新たに最適化問題を解きなおすというプロセスを，最適解が全ての不等式制約を満たすまで繰り返す．このように，適応的に不等式を加えていく手法は，整数計画法の研究分野では，切除平面法と呼ばれている．ALP 復号法の出力は LP 復号法の出力と一致することが示されおり，また実験的に ALP 復号法の計算量は，LP 復号法よりも少ないことが確認されている．

一方で，LP 復号法の復号性能を向上させた復号法の 1 つとして，分枝限定法に基づいた最尤復号のアルゴリズムが挙げられる [4]．この復号法では，元の線形計画問題に適応的に整数制約を加えて解くというプロセスを繰り返す．これは，木の探索アルゴリズムとして表現することもできる．分枝限定法に基づいた復号法の計算量は，加えられた整数制約に対して指数的に増大するが，その出力は最尤復号と一致することが保証される．

本研究では，分枝カット法に基づいた最尤復号のアルゴリズムを扱う．分枝カット法は分枝限定法と切除平面法を掛け合わせた技術である [1]．これは，線形計画法を解き，その解に対して切除平面法を行い，得られた解が非整数解であった場合，整数制約を加えた最適化問題を解くというプロセスを，分枝限定法と同様にして繰り返すというアルゴリズムである．分枝カット法は整数計画問題を解くアルゴリズムの 1 つとして幅広く用いられており，特に Feldman らは分枝カット法が組み込まれている整数計画ソルバーパッケージを使用した例を示している [2]．しかし，分枝カット法を問題に適用する際には，木の探索方法や，非整数解が得られた際の，整数制約を加える変数の選択などによって，その性能が大きく変化する．例えば，Draper らによって提案された復号アルゴリズムも分枝カット法の 1 つ見なすことが可能である．線形符号の復号問題に対して，分枝カット法を適用する場合に，アルゴリズムの要素部分をどのようにすれば良いかを検討することは重要である．本研究では，

* 〒 169-8555 東京都新宿区大久保 3-4-1 早稲田大学 Waseda University Okubo, Shinjuku Tokyo 169-8555 Japan.

分枝カット法に基づいた最尤復号アルゴリズムの一般的な枠組みを構築し，結果得られる幾つかのアルゴリズムを数値シミュレーションにより比較する．

本稿は以下のように構成される．第 2 章では最尤復号の問題と LP 復号法のアルゴリズムを概観し，第 3 章では，ALP 復号法と分枝限定法に基づいた復号法に関する記述を行う．第 4 章では，まず分枝カット法に基づく復号法のアルゴリズムの一般的な形式を記述した後に，復号法の性能を向上させる為の幾つかの手段に関する検討を行う．第 5 章で，数値シミュレーションの結果を与え，第 6 章で結論を述べる．

2 最尤復号問題と LP 復号法

$\mathcal{C}$ を 2 元 (n,k) 線形符号とし，符号語 $\boldsymbol{c}\in\mathcal{C}$ が，2 元入力，出力アルファベットが $\mathcal{R}$ の無記憶通信路を通して送信されるものとする．また $\boldsymbol{r}=(r_1,\cdots,r_n)^T\in\mathcal{R}^n$ を受信系列とする．この時，最尤復号の問題は以下のように定式化される．

$$\arg\max_{\boldsymbol{x}\in\mathcal{C}}\Pr(\boldsymbol{r}|\boldsymbol{x})=\arg\max_{\boldsymbol{x}\in\mathcal{C}}\sum_{i=1}^{n}\log\Pr(r_i|x_i). \tag{1}$$

この問題は，以下の線形計画問題と等価である．

$$\begin{aligned}&\text{minimize}\quad \boldsymbol{\gamma}^T\boldsymbol{x}\\ &\text{subject to}\quad \boldsymbol{x}\in\mathrm{conv}(\mathcal{C}),\end{aligned} \tag{2}$$

ここで，$\mathrm{conv}(\mathcal{C})$ は $\mathcal{C}$ の全ての符号語の凸包を表わし，$\boldsymbol{\gamma}=(\gamma_1,\cdots,\gamma_n)^T$ は以下で定義される対数尤度比のベクトルである．

$$\gamma_i=\log\left(\frac{\Pr(r_i|x_i=0)}{\Pr(r_i|x_i=1)}\right), \tag{3}$$

(2) 式は線形計画問題なので，制約式の数に対して多項式時間で解く解法が存在するが，一般的に制約式の数は符号語 n に対して指数のオーダで存在する．最尤復号の近似として，Feldman らは (2) を緩和した問題を解くことを提案した [2]．Feldman らは，パリティ検査行列の各行によって定義されるローカル符号語の凸包を定義した．これらの共通部分は 1 つの多面体 $\mathcal{P}$ を定義し，LP 復号法は，$\boldsymbol{\gamma}^T\boldsymbol{x}$ を $\mathcal{P}$ 上で解くものである．この多面体は，整数ベクトルの頂点と，非整数を含むベクトルの頂点を持つが，整数ベクトルの頂点は $\mathcal{C}$ の符号語と一致する．その為，LP 復号法は，出力が整数解であれば，それは最尤復号の結果と一致するという，ML certificate property という性質を持つ．

議論を完結にする為，多面体 $\mathcal{P}$ を規定する不等式に関する記述を行う．詳しくは文献 [2] を参照されたい．各パリティ検査行列の行 $j\in\mathcal{J}\in\{1,\cdots,k\}$ と，それに隣接する変数 $\mathcal{N}(j)\subset\{1,\cdots,n\}$ は，要素数が奇数となるような全ての $\mathcal{N}(j)$ の部分集合に対して，以下を満たす必要がある．

$$\sum_{i\in\mathcal{S}}x_i-\sum_{i\in\mathcal{N}(j)\backslash\mathcal{S}}x_i\le|\mathcal{S}|-1. \tag{4}$$

これに加え，以下の $2n$ 個の不等式が必要である．

$$0\le x_i\le 1\quad\text{for all } i\in\{1,\cdots,n\}. \tag{5}$$

以上により，緩和された線形問題は以下のように書ける．

$$\begin{aligned}&\text{minimize}\quad \boldsymbol{\gamma}^T\boldsymbol{x}\\ &\text{subject to}\quad 0\le x_i\le 1\quad\forall i\\ &\qquad\qquad \forall j\in\mathcal{J},\forall\mathcal{N}(j),\forall\mathcal{S}\subseteq\mathcal{N}(j),|\mathcal{S}|\text{ odd}\\ &\qquad\qquad \textstyle\sum_{i\in\mathcal{S}}x_i-\sum_{i\in\mathcal{N}(j)\backslash\mathcal{S}}x_i\le|\mathcal{S}|-1.\end{aligned} \tag{6}$$

3 ALP 復号法と分枝限定法に基づく復号法

3.1 ALP 復号法

全ての符号語 $x\in\mathcal{C}$ に対し，不等式 $\boldsymbol{a}^T\boldsymbol{x}\le b$ が満たされるとき，この不等式は妥当な不等式と呼ぶ．妥当な不等式に対して，$\boldsymbol{x}'$ がこの不等式を満たさないとき，不等式を $\boldsymbol{x}'$ に対するカットもしくは切除平面と呼ぶ．切除平面法は，カットを探索し，見つかったカットを問題に制約として加え，その問題を解きなおすというプロセスを繰り返すものである．文献 [5] で提案されている ALP 復号法は，単純な初期問題を解くところから始め，適応的に不等式制約の集合 (4) からカットを探索するものである．初期問題は以下のように設定されている．

$$\begin{aligned}&0\le x_i\quad\text{if}\quad\gamma_i>0,\\ &x_i\le 1\quad\text{if}\quad\gamma_i<0.\end{aligned} \tag{7}$$

カットを (4) から効率的に探索する方法については [5] を参照されたい．また文献 [5] では，妥当な不等式をカットの探索範囲に追加することで復号誤り率が改善されることが示されている．

3.2 分枝限定法に基づく復号法

分枝限定法は，線形計画問題に適応的に整数制約を加えていくものである．復号アルゴリズムのプロセスは，木の探索として表現することができる．加えられた整数制約が木のパスとして表現され，各ノードにはそれぞれ異なった混合整数計画問題が割り当てられる．深さ 0 の根ノードには元の線形計画問題 (6) が割り当てられる．各ノードは，それぞれ 0 と 1 のラベルが割り当てられた 2 つの枝を持ち，深さが $n-1$ である葉ノードは枝を持たない．各枝は 1 つの整数制約 $x_i=c$ に対応し，c は枝に割り当てられたラベルである．あるノードの子ノードに割り当てられた混合整数計画問題を，子問題と呼ぶ．子問題は親ノードの問題に，枝ノードに対応する整数制約を加えることで得られる．図.1 に例を示す．

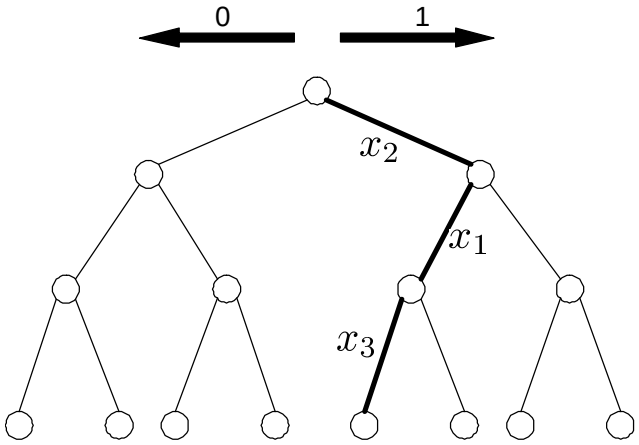

図 1: 整数制約が木のパスとして表現される．1 つの枝が 1 つの等式制約に対応する．太線で描かれたパスは $x_2 = 1, x_1 = 0, x_3 = 0$ という制約に対応.

分枝限定法に基づく復号法は，以下のように記述される：

Step.1 ノードリスト $\mathcal{L}$ の根ノードを加え，$\boldsymbol{x}^* = \mathbf{0}$, $z^* = \infty$ と初期化する.

Step.2 ノードリスト $\mathcal{L}$ から 1 つノードを取り出し，ノードリストから削除する．ノードリストが空の場合，$\boldsymbol{x}^*$ を最尤符号語として出力.

Step.3 Step.2 で選択されたノードに割り当てられた問題を解く．この問題の最適解を $\tilde{\boldsymbol{x}}$ とする.

- $\tilde{\boldsymbol{x}}$ が非整数かつ $\boldsymbol{\gamma}^T\tilde{\boldsymbol{x}} \leq z^*$ の場合：Step.2 で選ばれたノードの 2 つの子ノードを，ノードリスト $\mathcal{L}$ に加える.
- $\tilde{\boldsymbol{x}}$ が整数かつ $\boldsymbol{\gamma}^T\tilde{\boldsymbol{x}} \leq z^*$ の場合：$\boldsymbol{x}^*$ を $\tilde{\boldsymbol{x}}$ に，z^* を $\boldsymbol{\gamma}^T\tilde{\boldsymbol{x}}$ に更新する.

上記のアルゴリズムでは，ノードリストからどのノードを取り出すかなどが書かれておらず，これらの探索戦略を変更することで，幾つかの異なったアルゴリズムが得られる．Step.2 では，ノードリストからのノードの取り出し方に幾つかの戦略が存在する．幅優先探索，深さ優先探索，評価値優先探索が良く知られている．また，探索アルゴリズムの中で訪れるノードの数は，Step.3 で子問題を生成する際の，非整数変数の選び方にも依存する．文献 [4] で提案されているアルゴリズムは，Spte.2 の探索に幅優先探索，深さ優先探索を用い，Step.3 では $|\tilde{x}_i^{(0)} - 0.5|$ が小さいものを優先して選択するというものである．ここで，$\tilde{x}_i^{(0)}$ は根ノードに割り当てられた LP（元の LP）の最適解の i 番目の要素である.

4 分枝カット法に基づく復号法

分枝カット法は，分枝限定法と切除平面法を掛け合わせた技術である [1]．アルゴリズムは以下のように記述される：

Step.1 根ノードを $\mathcal{L}$ に加え，$\boldsymbol{x}^* = \mathbf{0}$, $z^* = \infty$ と初期化する．見つかったカットの集合 $\mathcal{B}$ をカットプールとし，空集合として初期化する.

Step.2 ノードリスト $\mathcal{L}$ から 1 つノードを取り出し，ノードリストから削除する．ノードリストが空の場合，$\boldsymbol{x}^*$ を最尤符号語として出力.

Step.3 S を Step.2 で得られたノードに割り当てられた混合整数計画問題とする．カットプール $\mathcal{B}$ に含まれる不等式を S に加える.

Step.4 Step.3 で得られた混合整数計画問題を解き，最適解に対して切除平面法を行う．切除平面法で見つかったカットをカットプール $\mathcal{B}$ に加える．$\tilde{\boldsymbol{x}}$ を切除平面法が実行された後の最適解とする.

- $\tilde{\boldsymbol{x}}$ が非整数かつ $\boldsymbol{\gamma}^T\tilde{\boldsymbol{x}} \leq z^*$ の場合：Step.2 で選ばれたノードの 2 つの子ノードを，ノードリスト $\mathcal{L}$ に加える.
- $\tilde{\boldsymbol{x}}$ が整数かつ $\boldsymbol{\gamma}^T\tilde{\boldsymbol{y}} \leq z^*$ の場合：$\boldsymbol{x}^*$ を $\tilde{\boldsymbol{x}}$ に，z^* を $\boldsymbol{\gamma}^T\tilde{\boldsymbol{x}}$ に更新する.

分枝限定法の場合と同様に，アルゴリズムの要素部分の選択により，幾つかの復号アルゴリズムが得られる．文献 [3] で Draper らによって提案されているアルゴリズムは，以下のようにして上記の分枝カット法の一例として導出可能である.

- 初期の LP を (7) のみを制約として構築し，この問題を根ノードに割り当てる.
- Step.2 のノードの選択では幅優先探索を採用する.
- 切除平面法でカットを探索する範囲は (4) とする.
- Step.3 の非整数の選択では，親ノードの最適解が 0.5 に一番近いところを選択する.

次に，アルゴリズムの計算量を削減する方法に関する検討を行う．文献 [3][4] では，探索の戦略として，幅優先探索もしくは深さ優先探索のみが採用されているが，一般的にも評価値優先探索は強力な探索の戦略である．評価値優先探索を採用した場合，Step.2 においてノードリストからノードを選択する際に，親ノードに割り当てられた問題の目的関数値が最も小さいものが選択される.

また，文献 [3][4] では，Step.3 の非整数変数の選択は，最適化変数の値のみに注目して決定されているが，各情報シンボルの信頼度を利用することでアルゴリズムの計算量を削減できる可能性がある．文献 [6] においても，木の探索に基づいた復号法が提案されているが，実験的に，より信頼度の高い情報シンボルに整数制約を入れること

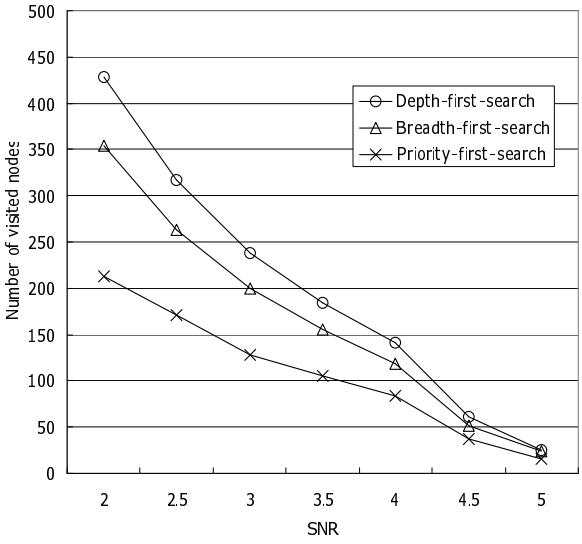

図 2: 探索戦略の違いによる探索ノード数の差

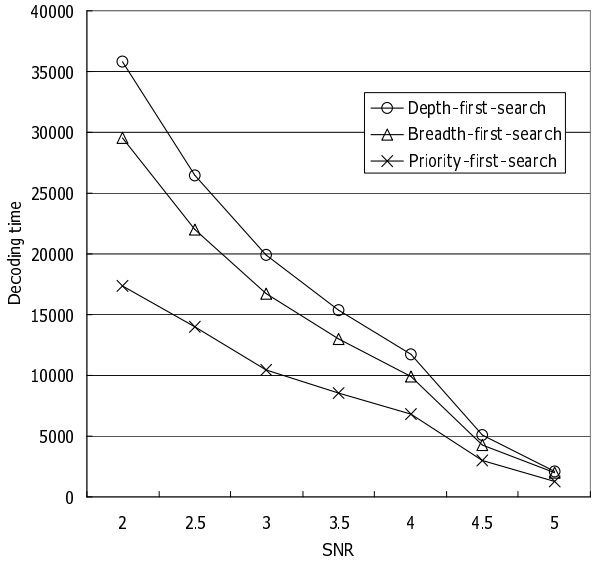

図 3: 探索戦略の違いによる復号時間の差

で，早い収束が見込めることが示されている．そこで親ノードの最適解 $\tilde{\boldsymbol{x}}$ に対して，インデックス i^* を以下のように定義する．

$$i^* = \arg\max_{i \in I_{\tilde{\boldsymbol{x}}}} |\gamma_i|, \tag{8}$$

ここで，$I_{\tilde{\boldsymbol{x}}}$ は，$\tilde{\boldsymbol{x}}$ が非整数であるインデックスの集合である．言い換えると，i^* は，非整数変数の中で対数尤度比 γ_i の絶対値が最も大きい位置 i である．

5 数値シミュレーション

ここでは，幾つかの復号アルゴリズムを数値シミュレーションにより比較を行う．実験は全て，正則 (60-30)LDPC 符号を用い，通信路には加法的白色ガウス通信路を仮定している．結果は 10000 回の実験結果の平均である．

5.1 探索戦略による違い

まずは，Step.2 における探索戦略の違いによる差を比較する．ここでは 3 つのアルゴリズムを比較する．アルゴリズムは全て，分枝限定法を用い，Step.3 における非整数変数の選択は文献 [4] のものと同じものとする．Step.2 における探索戦略は，幅優先探索，深さ優先探索，評価値優先探索の 3 つで比較を行う．各アルゴリズムの探索ノード数を図 2 に示す．SNR が大きい所では，各アルゴリズムに差は無いが，SNR が小さいときは，評価値優先探索がその他のアルゴリズムに比べて探索ノード数が少なくて済むことが確認できる．

各アルゴリズムでは探索以外にも操作が存在する．評価値優先探索を採用した場合，ノードリスト内のソートの操作がこれに含まれる．図 3 は各アルゴリズムの復号時間をプログラム上で計測した結果である．数値シミュレーションは 2.4GHz Pentium IV プロセッサを用いて行った．図 2 と図 3 の曲線形状が殆ど同じであることが確認できる．これより，混合整数計画問題を解く時間に比べて，ソートなどにかかる計算は殆ど無視できるものであると考えられる．

5.2 非整数変数の選択法による違い

次に，アルゴリズムの Step.3 における非整数変数の選択において，情報シンボルの信頼度を利用して計算量が削減できるかどうかを確認する．ここでは，2 つのアルゴリズムを比較する．いずれのアルゴリズムも分枝限定法に基づいたアルゴリズムとし，Step.2 における探索方法は評価値優先探索を採用する．Step.3 における非整数変数の選択は，一方は [4] のものと同じものとし，もう一方は (8) を元に選択を行う．各アルゴリズムの探索ノード数を図 4 に示す．情報シンボルの信頼度を利用することで探索ノード数を削減できていることが確認できる．

5.3 切除平面法の効果

最後に，分枝限定法と分枝カット法の比較を行う．いずれのアルゴリズムも評価値優先探索を採用し，非整数変数の選択は (8) に基づいて決定されるものとする．分枝カット法においては，初期の LP は (7) に基づいて構築されるものとし，カットの探索範囲は (4) の中から探索を行うものとする．図 5 は，各アルゴリズムの復号時間を比較したものである．分枝カット法が分枝限定法に比べて，復号にかかる計算時間が約半分となっていることが確認できる．

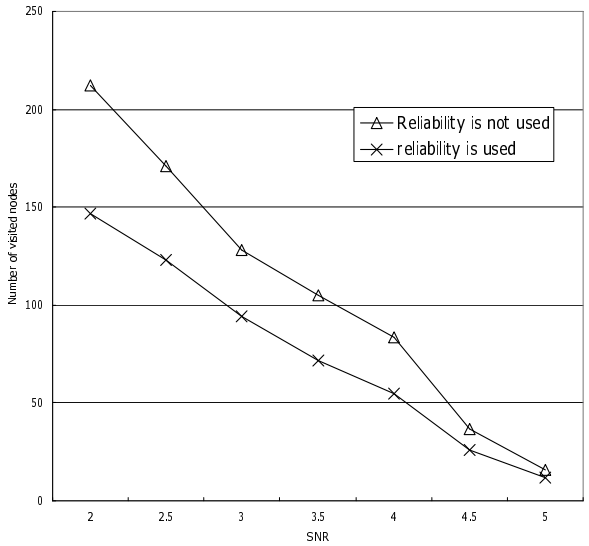

図 4: 情報シンボルの信頼度の利用による探索ノード数の差

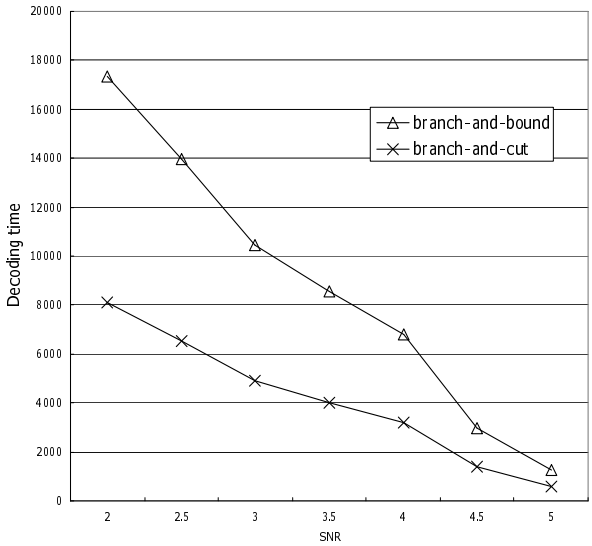

図 5: 分枝限定法と分枝カット法の復号時間の差

6 むすび

本稿では，分枝カット法に基づいた 2 元線形ブロック符号の復号法を提案した．分枝カット法に基づいた復号という枠組みで，複数のアルゴリズムが得られることを示した．例として，従来の復号法が分枝カット法に基づいた復号に含まれることを示し，評価値優先探索を利用したり，情報シンボルの信頼度を利用することで復号にかかる計算量を削減できることを示した．本稿で示したアルゴリズムのほかにも様々なアルゴリズムが分枝カット法に基づいて構築が可能である．例として，本稿では (4) のみをカットの探索範囲としたが，文献 [5] でも示されている通り，冗長なパリティ検査式を作ることでカットの探索範囲を広げることが可能である．分枝カット法においてカットの探索範囲を広げた場合，Step.4 にかかる計算量が増えるが，その代わりにアルゴリズム全体での探索ノード数の減少が見込める．アルゴリズム全体の復号時間を小さくする為に，カットの探索範囲をどのように取れば良いか，また探索範囲を広げた場合にどのようにカットを探索すれば効率的に探索できるかは，実験による検証が必要である．これは今後の課題としたい．

7 謝辞

本研究を行うにあたり，数多くの御助言を賜りました松嶋研究室の各氏に感謝致します．なお本研究の一部は日本学術振興会科学研究費新学術領域（No.20200044）及び早稲田大学特定課題研究（No.2009A-058）による．

参考文献

[1] A. Martin, ”General mixed integer programming : Computational issues for branch-and-cut algorithms,” Computational Combinatorial Optimization, Lecture Notes in Computer Science, Vol. 2241, pp. 1-25, 2001.

[2] J. Feldman, M. J. Wainwright, and D. Karger, ”Using linear programming to decoding binary linear codes,” IEEE Trans. Inf. Theory, vol. 51, no. 3, pp. 954-972, Mar. 2005.

[3] S. C. Draper, J. S. Yedidia, and Y. Wang, ”ML decoding via mixed-integer adaptive linear programming,” In Proc. Int. Symp. Inf. Theory, pp. 1656-1660, Nice, France, June 2007.

[4] K. Yang, J. Feldman, and X. Wang, ”Nonlinear programming approaches to decoding low-density parity-check codes,” IEEE J. Selected Areas Commun., vol. 24, no. 8, AUG. 2006

[5] M. H. Taghavi N. and P. H. Siegel, ”Adaptive linear programming decoding,” In Proc. Int. Symp. Inf. Theory, pp. 1374-1378, Seatle, USA, July 2006

[6] Y. S. Han, C. R. P. Hartmann, and C.-C. Chen, ”Efficient priority-first search maximum-likelihood soft-decision decoding of linear block codes,” IEEE Trans. Inf. Theory, vol. 39, no. 5, pp. 1514-1523, Sept. 1993.