

最適な LDPC 符号探索における MOGA の有効性

石田 由香里[†] 野里 裕高[†] 村川 正宏^{††}
梶谷 勇^{††} 坂無 英徳^{††} 高橋 栄一^{††}
古谷 立美[†] 樋口 哲也^{††}

LDPC 符号 (Low Density Parity Check Code) は、その高い誤り訂正能力により、最近、注目を集めている誤り訂正符号である。しかし、ハードウェア実装を想定した系統的な符号の構成方法が知られていないことが、実用上の問題点の一つになっている。そこで、本研究では昨年、最適な LDPC 符号の探索に多目的遺伝的アルゴリズム (MOGA) を用いた方法を提案した。本稿では、本システムにおける MOGA を他の探索手法や従来の LDPC 符号の構成方法と比較し、その結果、MOGA が最も効率的に優れた LDPC 符号を生成できることを示す。

The Effectiveness of MOGA in Optimal LDPC Code Search

YUKARI ISHIDA,[†] HIROTAKA NOSATO,[†]
MASAHIRO MURAKAWA,^{††} ISAMU KAJITANI,^{††}
HIDENORI SAKANASHI,^{††} EIICHI TAKAHASHI,^{††}
TATSUMI FURUYA[†] and TETSUYA HIGUCHI^{††}

Recently, LDPC (Low Density Parity Check) codes are attracting attention due to the considerable potential for error correction. However, the lack of systematic methods of constructing codes has impeded the implementation. Our study proposed a method to search optimal LDPC codes using MOGA (Multi-Objective Genetic Algorithm) last year. We conclude that the MOGA is more efficient than other approaches in finding 'good' LDPC codes as the result from our comparison.

1. はじめに

近年、誤り訂正符号のなかでも、高い誤り訂正能力をもつ LDPC (Low Density Parity Check) 符号¹⁾ が脚光を浴びつつある²⁾。誤り訂正符号とは、データ伝送系などで発生するデータの誤りを訂正するための符号化技術である。本来のデータに符号化処理を施して送信することで、受信者がその受け取ったデータから送信された本来のデータを高い精度で復元できる。

しかし、実システムへの応用に際して、次の 2 点が問題となっている。(1) 個々の LDPC 符号に対する系統的な構成方法が知られていないため、ユーザの要求仕様を確実に満たす LDPC 符号の設計が保証できない。(2) LDPC 符号のハードウェア実装を想定した場合、回路規模や動作速度といったハードウェア性能と誤り訂正能力がトレードオフの関係にあり、両方の要求を満たすような符号の設計には多大な手間や時間がかかる。

本研究では、これらの問題点を解決するために、多目的遺伝的アルゴリズム (MOGA) を用いた LDPC 符号の最適化設計システムを提案した³⁾。本システムを用いること

で、(1) 高い誤り訂正能力もち、(2) コンパクトで高速なハードウェア実装を想定した、LDPC 符号の自動かつ効率的な探索が実現可能である。

そこで、本稿では、MOGA を用いた本システムの有効性を示すために、従来の LDPC 符号の構成方法や、他の探索手法と、比較検証を行い、最適な LDPC 符号の探索における本システムの有効性を示す。

本稿の構成は次の通りである。2 章で本システムの概要を、3 章で有効性を示すために用いた方式の説明する。4 章で実験の結果を報告し、5 章でまとめを述べる。

2. MOGA を用いた LDPC 符号の探索

本章では、MOGA を用いた LDPC 符号の最適化設計システムについて説明する。

2.1 本システムの概要

個々の LDPC 符号は、検査行列と呼ばれるバイナリ行列 (0 と 1 だけを要素とする行列) で定義されることから、ハードウェア実装を想定した最適な検査行列を効率よく探索するシステムを提案している。

本システムは、図 1 に示すように 3 つの機能モジュールで構成されている。それぞれの機能モジュールでは、(a) 評価結果を用いた LDPC 符号の多目的最適化、(b) LDPC 符号のハードウェア実装時の性能評価、(c) LDPC 符号の誤り訂正能力の評価を行う。まず、(a) 多目的最適化モジュール

[†] 東邦大学大学院
Graduate School of Science, Toho University

^{††} 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)

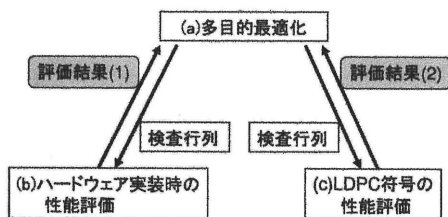


図1 本システムの概要

ルにて、いくつかの検査行列をランダムで生成する。次に、その検査行列を用いて、(b)と(c)の性能評価モジュールにて、性能評価を行い、評価結果(1)と評価結果(2)を出力する。最後に、評価結果(1)と評価結果(2)を用いて、(a)多目的最適化モジュールにて、新たに検査行列を探索する。このような3つの機能モジュールによる一連の処理を繰り返すことで、ハードウェア実装を想定した最適なLDPC符号を定義する検査行列を探索する。

次に、これら3つの機能モジュールについて詳しく述べる。

2.2 多目的最適化の導入方法

ハードウェア実装時の性能とLDPC符号の誤り訂正能力がトレードオフの関係にあることから、多目的最適化手法を導入した。多目的最適化を実現する方法の一つとして、MOGAを用いた。MOGAを用いることで、ハードウェア実装時の性能とLDPC符号の性能の両方を同時に満たす最適な検査行列を効率的に探索することを可能にした。

さらに、効率よく最適化を行うために、本システムでは、MessyGA⁴⁾を基にした可変長染色体GAを用いた。その理由は、LDPC符号の復号アルゴリズムの特性により、繰り返し処理の計算が検査行列の1の場所で行うことが知られているからである。つまり、検査行列内の1のある場所が、LDPC符号の誤り訂正能力に大きく影響を与える。そこで、1のある場所を染色体データとし、多目的最適化を行った。詳細については文献³⁾にて報告済みである。

2.3 ハードウェアの性能評価方法

本システムでは、ハードウェア実装時の回路規模や動作速度といったハードウェア性能評価を「検査行列内の1の個数」で近似する。その理由は、LDPC符号の復号アルゴリズムの特性により、繰り返し処理の計算量が検査行列中の1の個数に依存することが知られている⁵⁾からである。つまり、検査行列内の1の個数が少なくなれば、ハードウェア実装時の回路規模が小さくなり、動作速度も速くなる。このことから、検査行列内の1の個数を評価結果(1)とした。

2.4 LDPC符号の性能評価方法

LDPC符号の誤り訂正能力は、シミュレーションにより評価する。通信路モデルには、標準的な加算性ホワイトガウスノイズ(AWGN)通信を使用し、符号化と復号を行った⁶⁾。実際に通信路シミュレーションを行い、LDPC符号の訂正後のビット誤り率を評価結果(2)とした。

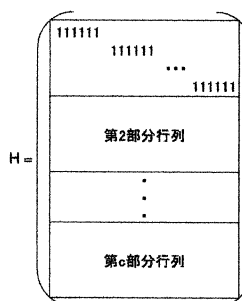


図2 Gallagerによる構成法

3. 従来方式

本章では、従来の構成方法やランダムサーチなどの探索方法について説明する。

3.1 従来の構成方法

LDPC符号の検査行列を生成する方法として、以下に示す構成法がある。

3.1.1 Gallagerによる構成法

まず、乱数を利用して符号アンサンブルから符号を選び出す方法について説明する。ここでは、代表的な方法として、Gallagerによる構成法¹⁾を取り上げる。

Gallagerによる構成法では、検査行列を図2に示すようにc個の部分行列に分ける。このときのcを部分行列数という。まず、第1部分行列の第1行目は連続するr個の1とそれに続く(n-r)個の0からなる。第2行目は第1行目を右にr個、巡回置換したものである。同様に最後まで行って第1部分行列は完成する。そして、第2部分行列以降は第1部分行列に列置換を施した部分行列となる。ただし、列置換はそれぞれの部分行列ごと独立に乱数を用いて選択される。この構成法により各行にr個、各列にc個の1を持つ検査行列が得られる。

3.1.2 擬巡回符号に基づく構成法

次に、代数的手法などを利用する確定的な構成方法について説明する。ここでは、代表的な方法として、擬巡回符号に基づく構成法⁷⁾を取り上げる。

擬巡回符号を用いた検査行列Hは、式1のような代数的な構造を持つ。

$$H = \begin{pmatrix} I & I & \cdots & I \\ I & \alpha & \cdots & \alpha^{(r-1)} \\ I & \alpha^2 & \cdots & \alpha^{(r-1) \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ I & \alpha^{(c-1)} & \cdots & \alpha^{(r-1) \times (c-1)} \end{pmatrix} \quad (1)$$

ここでIは単位行列であり、 α は $\rho \times \rho$ (ρ は素数)巡回置換行列である。巡回置換行列とは、単位行列を巡回させた行列である。この構成法により各行にr個、各列にc個の1を持つ、大きさ $(\rho \times r) \times (\rho \times c)$ の検査行列Hが得られる。

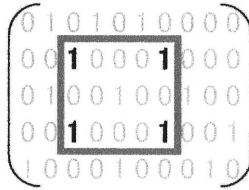


図 3 ループ

表 1 本システムのパラメータ

パラメータ名	値
検査行列の大きさ	1000×500
初期の 1 の割合	2%
SNR	2.0dB
交叉率	50%
突然変異率	100%
個体数	100 個体
評価世代数	100 世代
評価回数	10 試行

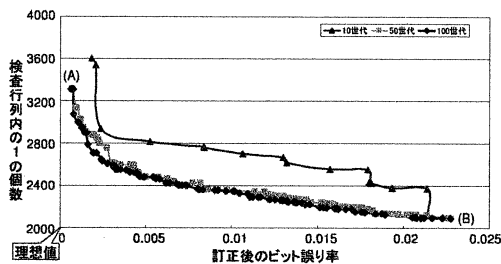


図 4 各世代のパレート最適解

3.2 ランダムサーチ

次に、探索法としてランダムサーチを説明する。これは、検査行列を探索空間からランダムに探索する。ただし、探索した検査行列に MOGA でも行っている以下の処理を施す。(1) 各行各列に 1 が 2 個以上存在するように、足りない行や列の要素が 0 の場所をランダムに選んで 1 に変更する。(2) ループが発生していたら、ループを構成している 4 つの要素からランダムに 1 つ選んで 0 に変更する。ループとは、図 3 のように検査行列中の 4 つの“1”が長方形又は、正方形を作っている状態のことである。この理由は、LDPC 符号の復元アルゴリズムの特性により、これらの条件に反する LDPC 符号は誤り訂正能力が低いことが分かっている⁶⁾。

4. 本システムの有効性検証結果

本章では、2 章で説明した本システムの有効性を検証するために、3 章で説明した方法との比較検証の結果を報告する。

4.1 本システムの評価結果

2 章で説明した本システムを用いて検証を行った。その時に使ったパラメータは、表 1 である。SNR は通信品質に影響を与えるノイズ量である。今回、評価時間が短いという理由から、ノイズの多い状況を選んだ。

表 2 Gallager による構成法のパラメータ

パラメータ名	値
検査行列の大きさ	1016×508
SNR	2.0dB
個体数	100 個体

表 3 擬巡回符号に基づく構成法のパラメータ

パラメータ名	値
α の大きさ	127×127
検査行列の大きさ	1016×508
SNR	2.0dB
個体数	126 個体

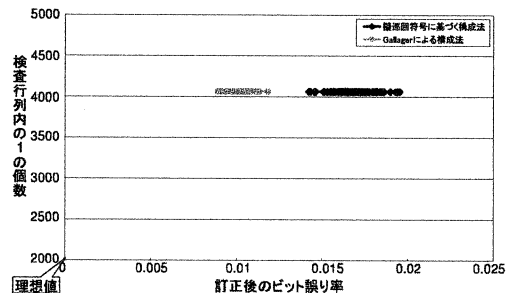


図 5 従来の構成方法を用いた検査行列の評価結果

ある試行による各世代のパレート最適解を図 4 に示す。図 4 は検査行列内の 1 の個数 (評価結果 (1)) を縦軸にとり、訂正後のビット誤り率 (評価結果 (2)) を横軸にとった。訂正後のビット誤り率の値が大きい理由は、検査行列の大きさが小さいことと、ノイズの多い状況で評価を行ったからである。図 4 は第 10 世代、第 50 世代、第 100 世代のパレート最適解を示す。この結果から、世代を迫うごとに理想値に近付いている、すなわち、最適化が進んでいることがわかる。

さらに、図 4 の第 100 世代のパレート最適解を見ると、訂正後のビット誤り率を優先するときは (A) 検査行列内の 1 の個数が 3312、訂正後のビット誤り率が 0.000625 を、検査行列内の 1 の個数の少なさを優先するときは (B) 検査行列内の 1 の個数が 2093、訂正後のビット誤り率が 0.022767 を、というようにユーザが目的に合った選好解を選ぶことができる。つまり、実装状況に合わせて、LDPC 符号を選択できる。

4.2 従来の構成方法の評価結果と比較

4.2.1 2 つの従来の構成方法の評価結果

ここでは、Gallager による構成法と擬巡回符号に基づく構成法を比較した。今回、検証に使ったパラメータは表 2 と表 3 である。擬巡回符号に基づく構成法の p は素数であることから今回 127 とした。さらに、 $r = 8$ 、 $c = 4$ と設定したので、大きさは 1016×508 の検査行列を生成した。正しく比較するために、Gallager による構成法も $r = 8$ 、 $c = 4$ となる大きさは 1016×508 の検査行列を生成した。

この 2 つの構成法の結果を図 5 に示す。図 5 は検査行列内の 1 の個数を縦軸にとり、訂正後のビット誤り率を横

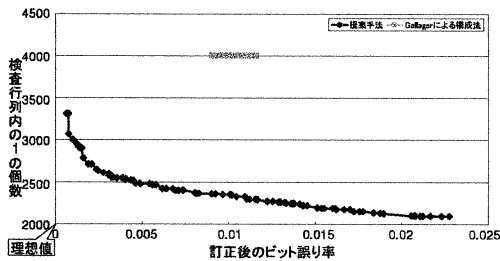


図 6 従来の構成方法と本システムの性能比較

表 4 ランダムサーチのパラメータ

パラメータ名	値
検査行列の大きさ	1000×500
初期の 1 の割合	0.5%
SNR	2.0dB
個体数	10100 個体

軸にとった。図 5 から、擬巡回符号に基づく構成法より、Gallager による構成法のほうが良い結果を得ることができた。

4.2.2 従来の構成方法と本システムの比較

ここでは、従来の構成方法と本システムを比較した。従来の構成方法は、4.2.1 にて Gallager による構成法のほうが良い結果が出たので、そちらを使用する。今回、Gallager による構成法で用いたパラメータは、検査行列の大きさを本システムと同じにするため 1000×500 に設定し、各行に 1 の個数が 8 個、各列に 1 の個数が 4 個になるようにした。それ以外は、表 2 と同じである。

本システムと従来の構成方法との比較結果を図 6 に示す。図 6 は検査行列内の 1 の個数を縦軸にとり、訂正後のビット誤り率を横軸にとった。図 6 から、従来の構成方法より、本システムの方が理想値に近い、つまり、性能が良い検査行列を探索することができた。一般に、検査行列の各行の 1 の個数がすべて等しく、各列の 1 の個数がすべて等しい場合より、異なっているほうが優れた誤り訂正能力を持つことが知られている⁶⁾が、本実験結果もそれを裏付けた。

4.3 ランダムサーチを用いた方法と本システムの性能比較

ここでは、ランダムサーチと本システムを比較した。今回、ランダムサーチの検証に使ったパラメータは表 4 である。

その結果を図 7 に示す。図 7 は検査行列内の 1 の個数を縦軸にとり、訂正後のビット誤り率を横軸にとった。図 7 から、ランダムサーチより、本システムの方が理想値に近い、つまり、性能が良い検査行列を探索することができた。さらに、ランダムサーチだと解がある部分に集まるが、本システムだと解に多様性があるので、ユーザの目的に合った選好解を選ぶことができる。

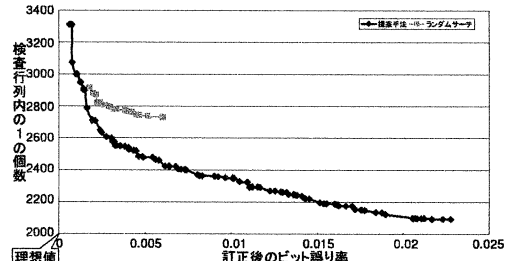


図 7 ランダムサーチとの比較

5. ま と め

本研究では、MOGA を Gallager による構成法や擬巡回符号に基づく構成法といった従来の LDPC 符号の構成方法や、ランダムサーチといった他の探索手法と比較した。その結果、MOGA が、それらの方法より、優れた性能を持つ LDPC 符号を探索することができた。つまり、本システムは優れた LDPC 符号の探索に有効である。

最後に今後の展望について説明する。まず、より効率的なパレート最適解を求めるには、もっと個体に多様性を持たせる必要がある。今後は NSGA-II⁸⁾などのより効率的な方法による実験を進める予定である。次に、より検査行列が大きい LDPC 符号を扱えるようにシステムを拡張する。さらに、探索空間も大きくなると同時に符号評価に要する時間も増大することから、クラスタシステムと MPI(Message Passing Interface) ライブラリ等を用いた並列化やハードウェアによる高速化などを導入する。

参 考 文 献

- 1) R. Gallager. Low-density parity-check codes. *M.I.T. Press, Cambridge, MA*, 1963.
- 2) Shu Lin. and Daniel J. Costello. *Error control coding second edition*. PEARSON Prentice Hall, 2004.
- 3) 石田由香里, 野里裕高, 飯島洋祐, 村川正宏, 梶谷勇, 高橋栄一, 古谷立美, 樋口哲也. 可変長染色体 GA を用いた LDPC 符号の最適化設計システムの提案. 情報処理学会数値モデル化と問題解決研究会, Vol. 2008, No. 17, 2008.
- 4) David E. Goldberg, Bradley Korb, and Kalyanmoy Deb. Messy Genetic Algorithms: Motivation, Analysis, and First Results. *Complex Systems*, Vol. 3, pp. 493–530, 1989.
- 5) 松嶋敏泰. ターボ符号・LDPC 符号とその復号法の概要. 電子情報通信学会誌, Vol. 88, No. 4, pp. 244–248, 4 2005.
- 6) 和田山正. 低密度パリティ検査符号とその復号法. 株式会社トリケップス, 2002.
- 7) J.L.Fan. *Constrained Coding and Soft Iterative Decoding*. Kluwer Academic Publishers, 2001.
- 8) Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A fast elitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA-II. *Parallel Problem Solving from Nature VI Conference*, pp. 849–858, 2000.