

行政院國家科學委員會專題研究計畫 期中進度報告

數位家庭:網路、平台與應用--子計畫六：家庭網路傳輸技術之研究(2/3) 期中進度報告(完整版)

計畫類別：整合型

計畫編號：NSC 95-2219-E-002-019-

執行期間：95年08月01日至96年10月31日

執行單位：國立臺灣大學電信工程學研究所

計畫主持人：許大山

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 96 年 12 月 18 日

行政院國家科學委員會補助專題研究計畫 成果報告

計畫名稱： 數位家庭:網路、平台與應用-
子計畫六：家庭網路傳輸技術之研究(2/3)

計畫類別：☐ 個別型計畫 ☒ 整合型計畫

計畫編號：NSC 95-2219-E-002-019-

執行期間：95 年 08 月 01 日至 96 年 07 月 31 日

計畫主持人：許大山教授

共同主持人：

計畫參與人員：蘇耿逸、林峻安、張正義、王偉震

成果報告類型(依經費核定清單規定繳交)：完整報告

本成果報告包括以下應繳交之附件：

☐ 赴國外出差或研習心得報告一份

☐ 赴大陸地區出差或研習心得報告一份

☒ 出席國際學術會議心得報告及發表之論文各一份

☐ 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

☐ 涉及專利或其他智慧財產權，☐ 一年☐ 二年後可公開查詢

執行單位：國立臺灣大學電信工程學研究所

中 華 民 國 96 年 5 月 25 日

Abstract:

All the digital devices in a digital home need to be connected together. The most convenient form of connectivity is through wireless connections. Compared to a wired link, a wireless link suffers from unpredictability of propagation channel and interference characteristic. For consumer devices suitable for digital home applications, a typical practice is to set the transmission rate low enough that it is practically not possible for transmission to fail. In this manner, there is no need for complicated medium access control (MAC) protocol and network layer protocol to cope with failed transmission. However, this practice is definitely not satisfactory given a relentless thirst for bandwidth to support rich multimedia application in the home.

In this year, we focus on a specific physical layer (PHY) problem in the typical implementation of error control code decoders. For high speed digital wireless data communication within a home, capacity-achieving codes such as the Turbo code and the low density parity check (LDPC) code are popular choices. The decoders for these codes are based on the bit-by-bit maximal a posteriori principle. While these decoders work very well in high SNR regime, their performance greatly suffer in low SNR conditions. From the well-known theory of the sphere packing bound (SPB), we speculate that the performance gap from the ideal maximal-likelihood (ML) decoder in the 10^{-1} to 10^{-2} regime can be more than 3 dB.

In light of this, we focus on establishing the ML performance of Turbo codes. Even though the Turbo code was proposed more than a decade ago, the theoretical ML performance – as well as the method to compute it through simulation – has never been established before. It was generally considered impossible. In this project, we propose the Genetic Decoding Algorithm (GDA) for turbo codes. GDA combines the principles of perturbed decoding and genetic algorithm for excellent decoder performance. In GDA, chromosomes are random additive perturbation noises. A conventional turbo decoder is used to assign fitness values to the chromosomes in the population. After generations of evolution, good chromosomes that correspond to decoded codewords of very good likelihood emerge. GDA can be used as a practical decoder for turbo codes in certain contexts. It is also a natural multiple-output decoder. The most important aspect of GDA, in our opinion, is that one can utilize GDA to empirically determine a lower bound on the error probability with ML decoding. Our results show that, at a word error probability of 10^{-4} , GDA achieves the performance of ML decoding. Using GDA, we establish that an ML decoder only slightly outperforms a MAP-based iterative decoder at this word error probability for the block size we used. However, as the error probability increases, the penalty due to MAP-based decoding starts to appear, especially for code block size lower than 1000.

中文計畫摘要：

無線連線必須承受無法預測的傳遞通道以及干擾特性的變化。對適用於數位家庭的消費產品，一般慣例是將傳輸速率設定得夠低，使得實際上的傳輸不會失敗。在這種情況下，並不需要複雜的媒體存取控制(MAC)與網路層的協議來處理失敗的傳輸。但是，這種作法並無法滿足未來對頻寬的需求。

在此計畫中，我們尋找在高錯誤率區間提升實體層效能的方法。渦輪碼(Turbo code)與低密度奇偶校驗碼(LDPC code)在數位家庭無線傳輸上是熱門的選擇。解碼器則是基於最大事後機率(MAP)的原則。這些解碼器在低錯誤率區間下工作得很好，而在高錯誤率的情況下則承受高效能損失。由理論上的球型包裝界限(SPB)，我們推測在錯誤率 10^{-1} 至 10^{-2} 區間，相對於理想最大概率(ML)解碼的效能損失會大於 2dB。

我們著重於建立渦輪碼之最大概率度解碼效能。儘管渦輪碼已被提出超過十年，最大概率度解碼之效能，以及經由模擬計算此一效能的方法，仍然未被提出。我們針對渦輪碼提出了一個基因解碼演算法(GDA)。基因解碼演算法結合了干擾解碼(perturbed decoding)與基因演算法的原理以提供極佳之解碼器效能。在基因解碼演算法中，染色體是隨機的干擾雜訊；傳統的渦輪解碼器，則被用來為每個染色體評定個別的分數。經過世代的演化後，對應於高概率度的碼字的染色體就會出現。基因解碼演算法在某些狀況下可被用為一個實際可行的渦輪解碼器。我們使用基因解碼演算法，實際得到最大概率度解碼錯誤率效能的下極限。從我們的結果中可知在錯誤率 10^{-4} 附近，基因解碼演算法已經可以達到最大概率度解碼的效能。在這個錯誤率區間，最大概率度解碼器的效能僅比最大事後機率的遞回式解碼器來得好一些。但是，隨著錯誤率增加，最大事後機率的解碼器付出的代價開始出現，尤其是碼區塊大小低於 1000 時。

目錄	頁次
中英文摘要	2
I. Introduction	7
II. Background	8
A. Maximal-a priori (MAP) Iterative Decoding Algorithm	9
B. Genetic Algorithm	10
III. Genetic decoding algorithm	10
A. Perturbed Decoding for Turbo code	10
B. Genetic Decoding for Turbo Codes	11
C. ML bound using GDA	12
D. Complexity Aspects for GDA	12
IV. Simulation Results	12
A. Simulation Setup	12
B. Parameters and Procedures for GDA	12
C. Performance of GDA	13
V. Conclusion	15
VI. Self-estimation	16
VII. Reference	17

I. Introduction

Since its invention [1], turbo codes have received a lot of interest. Turbo codes are known to achieve a performance very close to the channel capacity. Due to this strong performance and manageable decoder complexity, turbo codes have been adopted as options for forward error correction in many recent communications standards such as 3GPP, 3GPP2 and Wi-MAX. Over the years, researchers have answered questions such as error floor and interleaver optimization.

Practical decoders for turbo codes are iterative decoders based on message-passing principles [2]. These decoders are suboptimal from a word error probability perspective. Although yielding the lowest word error probability, ML decoding of turbo codes has been considered unrealistic so far because efficient ML decoders have not been discovered. In the past several years, performance of ML decoding of turbo codes has drawn more attention. In [3] and [4], the union bound on the average error probability over the ensemble of all interleavers was investigated. They are subsequently improved upon [5] [6]. However, these results are valid for the average performance of a set of turbo codes; they cannot be used to establish the performance of a *given* specific turbo code. In [7], a decoding algorithm based on two component list Viterbi decoders was introduced. It could only yield ML decoding of turbo codes for an infinite list size.

In [8], the "perturbed decoding algorithm" (PA) was proposed for a concatenated coded system consisting of an inner error correcting code and an outer error detecting code. In PA, multiple highly likely candidate codewords are obtained by feeding the inner decoder with slightly perturbed versions of the actual received signal. The candidate codewords are subsequently *validated* by an outer decoder. Among the validated codewords, the one with the highest likelihood is declared the final decoder result. As we will demonstrate in this paper, one can adapt the perturbed decoding concept for use with turbo codes. Unfortunately, due to the large code block size used by turbo codes, the direct application of PA to turbo codes so can require an enormous complexity.

Genetic algorithms (GA) constitute a class of optimization techniques that are based on the general laws of natural selection and genetics [9], [10]. It has been applied to numerous intractable optimization problems. In this paper, we present the novel Genetic Decoding Algorithm (GDA) which combines the principles of GA and PA. In GDA, one maintains a pool of additive perturbation noises as the chromosomes. A conventional decoder is used to give scores to these chromosomes. Specifically, one first adds a chromosome to the received signal. The perturbed signal is passed through a conventional decoder to generate a corresponding decoded codeword. The likelihood of the codeword is assigned as the score of the corresponding chromosome. Chromosomes of low scores are eliminated from the population. After several generations of evolution, good chromosomes that correspond to codewords of very good likelihoods emerge.

Although GDA is trivially guaranteed to produce decoded sequences of higher likelihood than a conventional turbo decoder, at its current form the amount of

computation required for GDA is still too large for most practical applications. The critical importance of GDA, in our opinion, lies in the fact that it can be used to *empirically* (i.e., via simulation) establish the fundamental ML performance of a given turbo code. As will be shown later in this paper, for the turbo codes we simulated, GDA can produce ML decoding performance at a word error probability of 10^{-4} and lower. In those applications where the computation load of GDA is acceptable, our results indicate that GDA can outperform conventional MAP decoder by a gain of 0.05 to 0.3 dB.

The rest of this paper is organized as follows. In Section II, we briefly review the conventional iterative decoding algorithms for turbo codes and the fundamental concepts of GA and PA. In Section III, we present in detail the concept and procedure of GDA for turbo codes. In Section IV, our simulation results of ML decoding bounds for turbo codes produced by GDA are presented. The parameters and procedures for GDA are also discussed. Concluding remarks are given in Section V.

II. Background

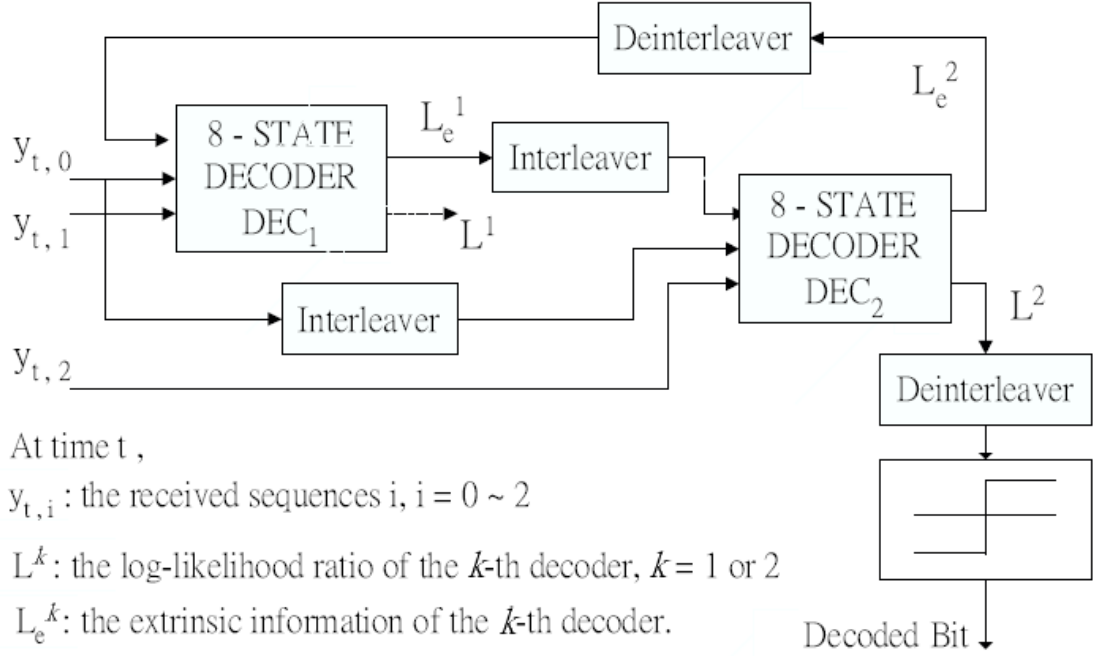


Fig. 1. Conventional iterative decoding algorithm for turbo codes.

A. Maximal-a priori (MAP) Iterative Decoding Algorithm

In this paper, by “turbo codes” we specifically mean parallel concatenated convolutional codes (PCCC). A PCCC is consisted of a parallel concatenation of two constituent recursive convolutional codes and a specific interleaver. For example, a rate 1/3 PCCC employed by WCDMA [11] is consisted of two identical constituent encoders and an interleaver. The two encoders share the same generator polynomial, which is given by

$$g(D) = \left[1 \quad \frac{1 + D + D^3}{1 + D^2 + D^3} \right]. \quad (1)$$

These two constituent encoders are separated by a prunable prime interleaver [12].

As illustrated in Figure 1, a conventional MAP iterative decoder is comprised of two constituent decoders linked by the interleaver. These two constituent decoders pass extrinsic information back and forth. Decoder output is obtained after several iterations. Although many other forms of decoders exist, in this paper we only consider the MAP decoder due to its excellent performance.

B. Genetic Algorithm

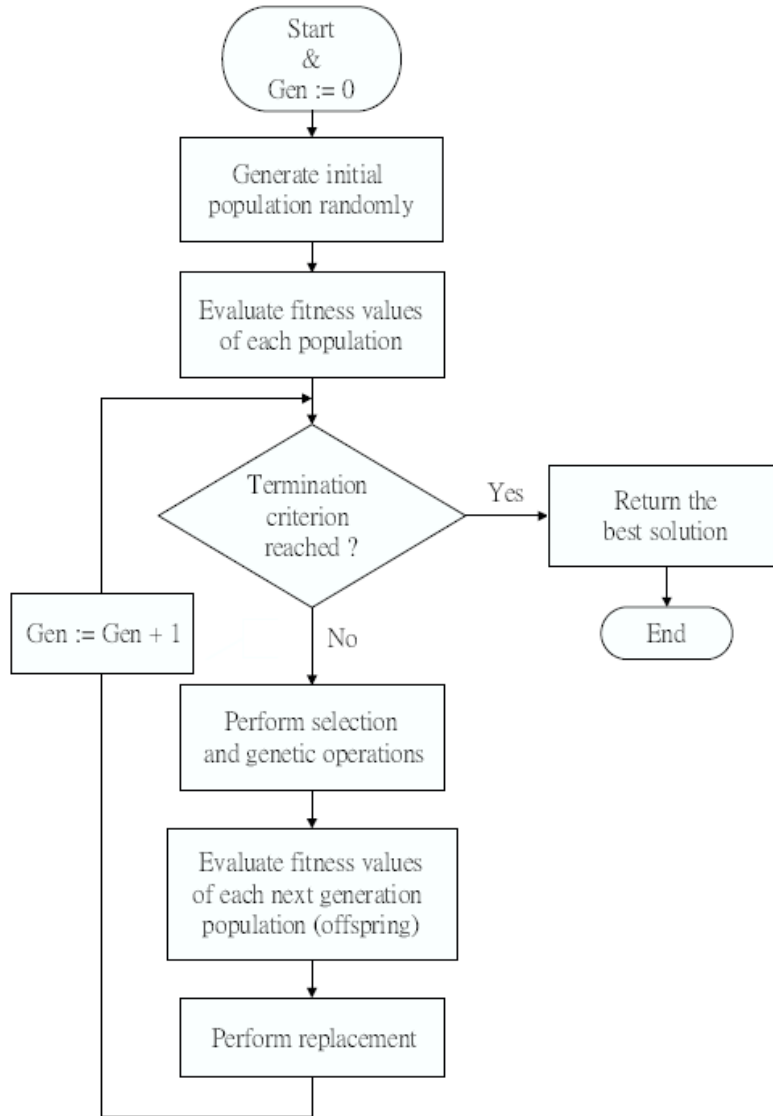


Fig. 2. Flowchart of GA.

To apply GA to an optimization problem, the first and sometimes the most critical step is to formulate a good genetic representation. Once a genetic representation is chosen, a typical genetic algorithm (GA) consists of three fundamental operations: selection, genetic operation, and replacement. The selection step is a process that selects members from a population into the mating pool. The genetic operation includes both crossover and mutation. Crossover combines the subparts of the chromosomes from two parents to produce offsprings. Mutation, on the other hand, introduces novel variations into the chromosomes. Replacement is the step that eliminates weak members in order to remain a population of high fitness. As mentioned in [9], [10], a typical GA procedure can be described as follows:

- 1) Initially, a population is generated randomly.
- 2) Evaluate the fitness values for each individuals of the population by calculating some objective function.
- 3) Select from the population some parents to produce the next generation of

population (offsprings) by the genetic operations.

- 4) Evaluate the fitness values of offsprings in a similar fashion to their parents.
- 5) According to a chosen replacement strategy, some parents and some offsprings survive. The survivors form the population of the subsequent generation.
- 6) Check whether the desired termination criterion is reached or not. If not, return to step 3 and repeat the process. Otherwise, stop and produce the final solution.

We can refer to the flowchart in Figure 2 for the detailed course of the execution of GDA. Hereafter, we will tailor the objective function of the fitness value evaluation block and the termination criterion for the specific goal of decoding a turbo code.

C. Perturbed Decoding Algorithm

Perturbed decoding algorithm (PA) is a decoding algorithm for concatenated error correction and error detection codes. In PA, the original received signal $\mathbf{y}$ is first decoded by the inner decoder. The decoded result is subsequently validated by an outer decoder. If the outer decoder regards the decoded result as invalid, an additive perturbation noise will be randomly generated. By adding the noise to $\mathbf{y}$, one creates a perturbed received signal. This perturbed signal is decoded by the inner decoder. The decoder output, which may be different from the initial decoder output, is again validated by the outer decoder. This process is repeated until either a valid codeword is declared by the outer decoder or when some termination criterion is met. It is a practical strategy to produce multiple decoder outputs with a simple decoder.

III. Genetic Decoding Algorithm

A. Perturbed Decoding for Turbo Codes

The principle of PA can be extended to decoding turbo codes. Let us consider a BPSK-modulated coded transmission system described as follows. The information bit sequence is $\mathbf{d} = (d_1, d_2, \dots, d_K)$, where $d \in \{0,1\}$. These bits are encoded to produce the transmitted symbols $\mathbf{x} = (x_1, x_2, \dots, x_N)$, where $x_n \in \{0,1\}$. The received signal $\mathbf{y} = (y_1, y_2, \dots, y_N)$ is disturbed by real-valued additive white Gaussian noise of variance $N_0/2$:

$$y_n = x_n + w_n, \quad n = 1, 2, \dots, N, \quad (2)$$

where w_n are i.i.d. $\mathcal{N}(0, N_0)$. Given $\mathbf{y}$, the likelihood of a potential transmitted information sequence $\mathbf{d}'$, whose corresponding coded symbol sequence is $\mathbf{x}(\mathbf{d}')$, is

$$L(\mathbf{d}') = \prod_{n=1}^N \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(y_n - x_n(\mathbf{d}'))^2}{N_0}\right). \quad (3)$$

Consider some decoder for turbo code. It maps the received signal $\mathbf{y}$ into some $\mathbf{d}$. If we perturb the input signal by an additive signal $\mathbf{v}$, the decoder can produce a different output $\mathbf{d}_{\mathbf{v}}$ whose corresponding likelihood is $L(\mathbf{d}_{\mathbf{v}})$. As in the case for PA, by feeding the decoder with randomly perturbed copies of received signal $\mathbf{y}$, one can produce a pool of possible decoded sequences. The highest-likelihood decoded sequence eventually converges to the ML decoder output $\mathbf{d}_{\text{ML}}$, provided that:

- 1) A set of perturbation signals, $\mathcal{V}$, exists such that for any perturbation signal $\mathbf{v} \in \mathcal{V}$ this decoder maps $\mathbf{y} + \mathbf{v}$ into $\mathbf{d}_{\text{ML}}$.
- 2) According to the procedure to generate the perturbation signal $\mathbf{v}$, the probability of selecting a signal $\mathbf{v}$ that belongs to $\mathcal{V}$ is nonzero.

Whether the first condition is met depends on the particular decoder. For most reasonable decoders though the condition is met trivially. Given that this condition is met, if the set $\mathcal{V}$ is not problematic, the second condition can be easily met by choosing i.i.d. white

Gaussian entries as perturbation noise ν_n . Therefore, one expects that, when given a decoder that satisfies the first condition, the performance of ML decoding can be empirically determined by executing the perturbed decoding algorithm. Unfortunately, due to the large block size used by turbo codes, doing so requires an impractically enormous amount of computation!

B. Genetic Decoding for Turbo Codes

In order to avoid the complexity of PA for turbo codes, instead of randomly generating perturbation signals, we wish to extract useful information from prior perturbation-decoding attempts and subsequently apply such information to the generation of new perturbation signal. To achieve this, we combine the genetic algorithm with the PA.

In our proposed GDA for turbo codes, a chromosome is a perturbation signal ν . The genes are the individual additive perturbation noise components. Our goal is to find at least one chromosome which, when added to the receive signal y , induces the decoder to produce the ML output d_{ML} . To this end, the fitness of a chromosome ν is defined to be the likelihood of the corresponding decoded sequence, $d(\nu)$:

$$FV(\nu) = L(d(\nu)), \quad \text{where } y + \nu \xrightarrow{\text{decode}} d(\nu). \quad (4)$$

The above definition of genetic information and fitness evaluation is the core concept of GDA for turbo codes. With them, appropriate definitions can be found for other critical functional blocks in GA: parent selection, cross over, and mutation. The overall procedure of GDA then follows that of a typical GA. In Section IV, we will present our definitions of these functional blocks.

C. ML bound using GDA

GDA is an effective tool to establish a lower and an upper bound of ML decoding even though it itself is not an ML decoder. By definition, the error performance of GDA itself is an upper bound, because no decoder offers better word error probability than ML decoding. When *any* decoder produces an output with a likelihood higher than that of the actually transmitted encoded sequence, the ML decoder must either decode to this erroneous output or some other sequence with even higher likelihood. Thus we can conclude that ML decoder must also make a decoding error on the same received signal. On the other hand, when a decoder produces an output with a likelihood less than or equal to that of the actually transmitted encoded sequence, we can't be certain whether some other sequence with likelihood higher than that of the actually transmitted encoded sequence exists. If we assume an ML decoder will decode correctly over all of these signals, we would have over-estimated the performance of ML decoding. In light of this, a lower bound of ML decoding can be empirically established using a practical decoder to be the probability when the said decoder decodes into a sequence of higher likelihood than that of the actually transmitted encoded sequence.

Note that unless the said decoder does behave very similarly to an ML decoder, this lower bound tends to be very poor. For example, if an MAP decoder is used, the lower bound is entirely useless. With GDA as the decoder, however, the two bounds are very close.

D. Complexity Aspects for GDA

GDA involves repeated uses of a turbo decoder to compute fitness values for individuals in a population. The complexity of GDA can be several orders of magnitude higher than that of a conventional turbo decoder. Nevertheless, in certain context, GDA can be used as a practical decoder. For instances, when a turbo code is concatenated by a CRC, one can first pass received signal through a conventional turbo decoder. GDA is activated only when the conventional turbo decoder fails. The average decoding complexity of GDA is weighted down by the first-pass error probability.

IV. Simulation results

A. Simulation Setup

We use the turbo code defined in the WCDMA specification in our simulation [11]. The interleaver used in the specification is prunable prime interleaver [12]. For a block size of K , the actual code rate is $K/(3K+12)$. Encoded bits are modulated using BPSK modulation and transmitted over an AWGN channel. The “symbol-by-symbol” MAP rule [13] is used for the turbo decoding algorithm that produces the fitness values. Eighteen full iterations are used for the MAP decoder.

B. Parameters and Procedures for GDA

In this paper, our goal is to demonstrate the potential performance of GDA with manageable complexity. We do not claim that the choice of parameters below is optimal in terms of minimizing complexity. Actually, we surmise that our results can be materially improved if we could execute more computation.

From Section III-B, we realize that in order to ensure a nonzero probability of obtaining a perturbation signal that induces the ML-decoding output, it is prudent to allow the perturbation signal to take any values. In light of this, we produce the first generation of perturbation noises as i.i.d. zero mean Gaussian vectors. The signal-to-perturbation noise ratio is defined as

$$SPNR = 10 \log_{10} \left(\frac{E_b}{\sigma_p^2} \right). \quad (5)$$

It is suggested in [8] that as long as the power of the perturbation noise is chosen reasonably, the expected number of attempts until an “ideal noise” occurs is a very weak function of SPNR. Based on empirical observation, we chose 15 dB to be the SPNR used in our simulation.

It is known that the quality of the initial population strongly affects the success of the subsequent evolution. Therefore, we randomly generate 10000 original populations and keep only the fittest 100 individuals to be the initial population. The crossover rate is selected at 80 percent, i.e. a randomly selected 80 percent of the population in a generation participates in the production of new offsprings. When two chromosomes mate, we use uniform crossover to construct offsprings. A gene is copied from each parent with equal probability. On average, each parent donates 50 percent of its genetic material to its offspring. Two offsprings are generated for a pair of parents. Only the two fittest among the four members of a family survive to the next generation. Each gene of a survivor is mutated independently with a small mutation probability of 0.5 percent. Mutation replaces the gene by a randomly generated perturbation noise with the same SPNR. The 20 percent of the population that does not participate in the mating process survives to the next generation by default.

In our simulation, we terminate the GDA if (1) the predefined number of generation

is reached, or (2) a predefined percentage of population converges to one fitness value which is the maximal value among all populations in any generation. We choose these two thresholds at 300 and 20 percent, respectively.

In these results we also present the lower bound for ML decoding computed using the concept of III-C. Recall that the GDA output can be either the originally transmitted codeword with likelihood $L(\mathbf{d})$, a codeword with likelihood lower than $L(\mathbf{d})$, or a codeword with likelihood higher than $L(\mathbf{d})$. In order to produce a tight lower bound, when the first two cases occur, we re-execute GDA for another two rounds.

C. Performance of GDA

The word error probability for turbo codes given information block sizes of 2560 and 5120 bits per block is shown in Figure 3 and Figure 4, respectively. The error performance of conventional decoder, GDA, and the lower bound of ML performance derived from the application of GDA are shown. As can be seen, at a word error probability of 10^{-4} , GDA achieves the error performance of ML decoding. With the help of GDA, here we can determine a very important result empirically: the penalty of using a MAP-based iterative decoder, from a minimization of word error probability perspective, is less than 0.2 dB at a word error probability of 10^{-4} for the turbo codes we consider.

In addition to word error probability, we also show the information bit error probability for the two block sizes in Figure 5 and Figure 6, respectively. Again the error performance of the conventional MAP decoder, GDA, and the lower bound of ML performance derived from the application of GDA are contrasted. One finds that the bit error probability follows that of the word error probability closely. Hence we surmise that the number of bit errors per word error for conventional decoders and for GDA are of approximately the same order.

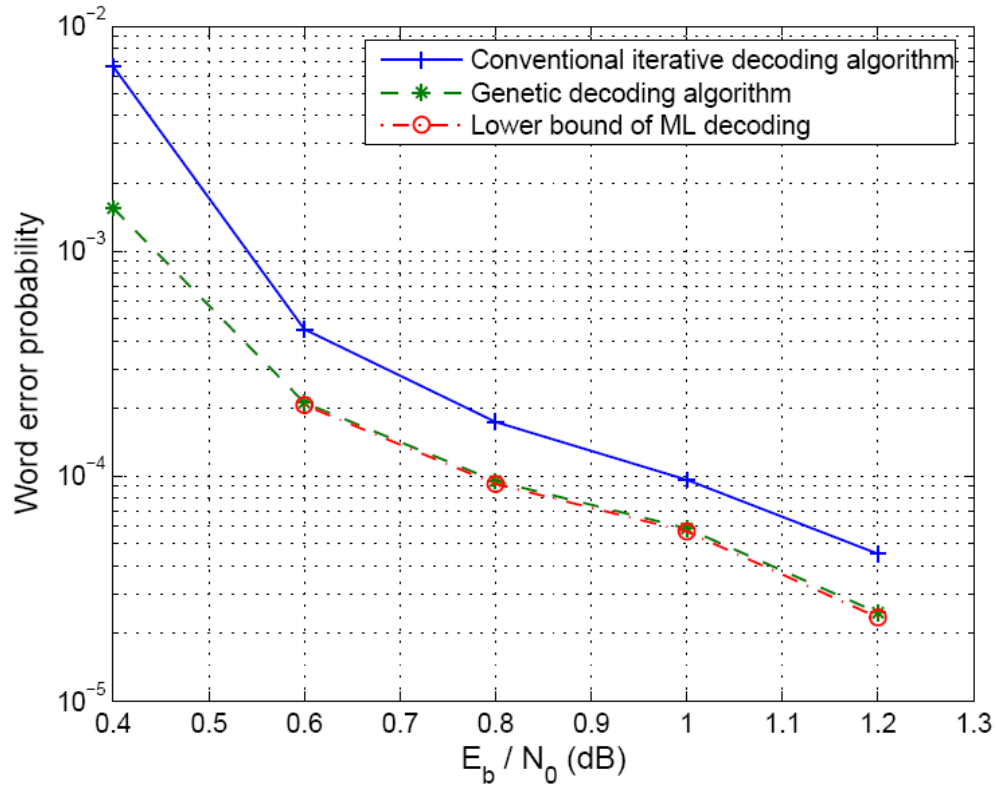


Fig.3. Word error probability for turbo code. The block size is 2560.

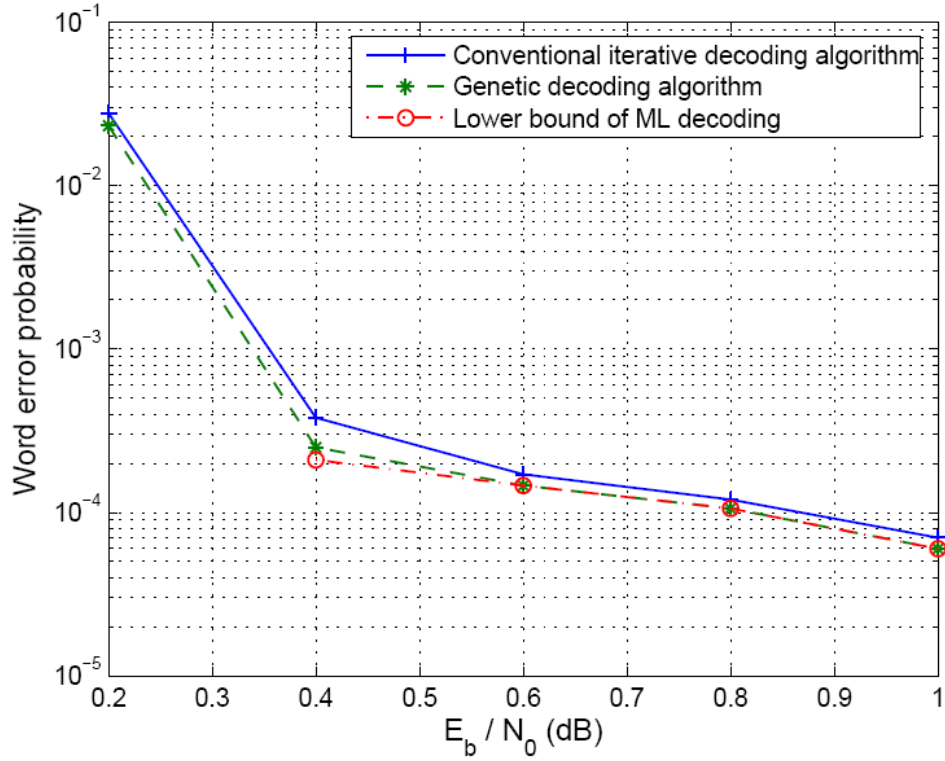


Fig.4. Word error probability for turbo code. The block size is 5120.

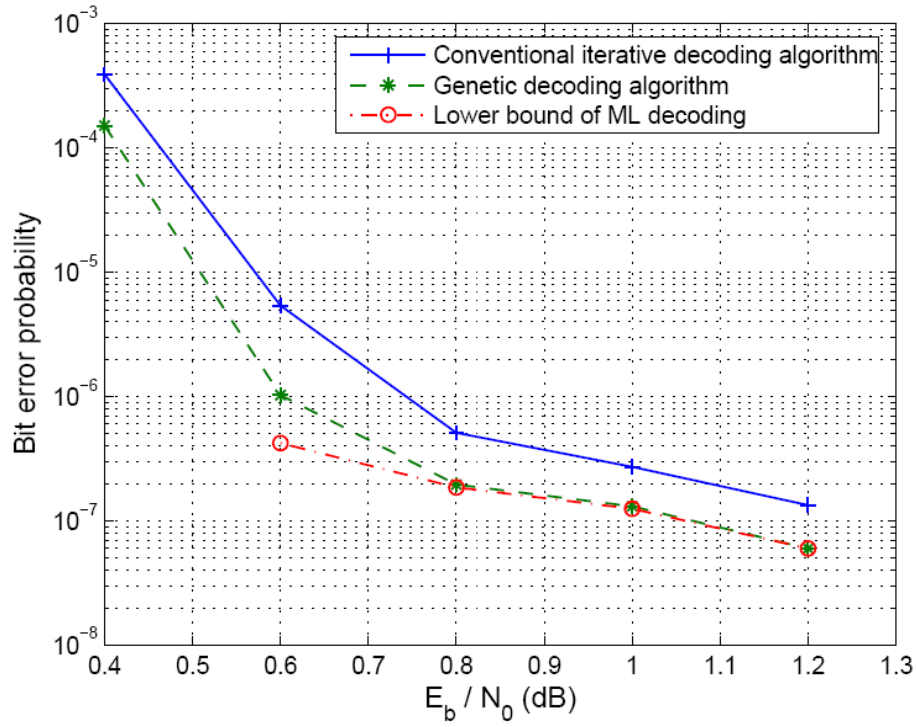


Fig. 5. Bit error probability for turbo code. The block size is 2560.

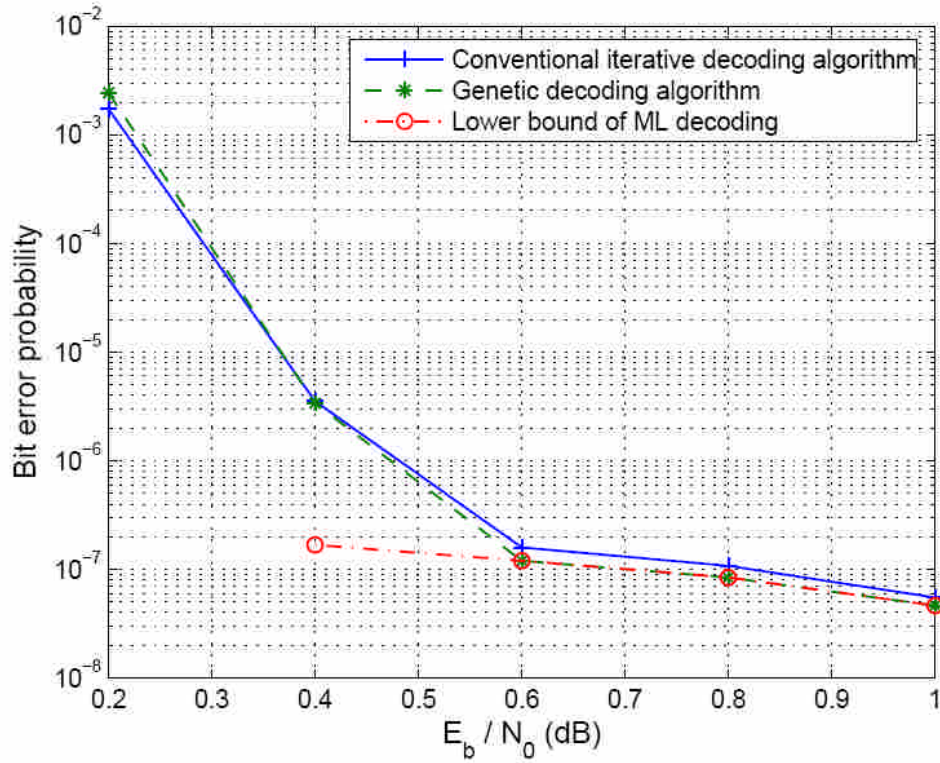


Fig. 6. Bit error probability for turbo code. The block size is 5120.

V. Conclusion

In this report, we present a novel decoding algorithm, the Genetic Decoding Algorithm (GDA), for use with turbo codes.

In GDA, we propose to use random perturbation noises as the genes. A conventional decoder is used to assign fitness values to chromosomes. Specifically, one adds the noise pattern (chromosome) to the received signal first. A well chosen decoder decodes the perturbed signal and generates a corresponding decoded codeword. The likelihood of the codeword is assigned to be the fitness value of the corresponding chromosome. Chromosomes of low fitness values are eliminated from the population. After several generations of evolution, good chromosomes that correspond to codewords of very good likelihoods emerge.

GDA can be used as an empirical tool to establish the ML decoding performance for a given specific turbo code. The performance of GDA itself can serve as an upper bound. The probability where GDA finds a codeword with higher likelihood than the transmitted codeword can be used as a lower bound. These bounds are very tight at lower error probability regime. Our simulation results indicate that these two bounds converge at a word error probability of 10^{-4} . From our results, we conclude that the difference between an ML decoder and a MAP-based iterative decoder is less than 0.2 dB at this word error probability and the block size we simulated. Because GDA produces the lower bound, it achieves the ML error performance in such regime.

VI. Self-estimation

We summarize our research achievement of this sub-project in this year here:

1. A novel decoding algorithm, the Genetic Decoding Algorithm (GDA), for use with turbo codes :

Turbo code is used in several standards, including Wi-MAX. In GDA, we propose to use random perturbation noises as the genes. A conventional decoder is used to assign fitness values to chromosomes. Specifically, one adds the noise pattern (chromosome) to the received signal first. A well chosen decoder decodes the perturbed signal and generates a corresponding decoded codeword. The likelihood of the codeword is assigned to be the fitness value of the corresponding chromosome. Chromosomes of low fitness values are eliminated from the population. After several generations of evolution, good chromosomes that correspond to codewords of very good likelihoods emerge.

2. GDA can be used as an empirical tool to establish the ML decoding performance for a given specific turbo code:

The performance of GDA itself can serve as an upper bound. The probability where GDA finds a codeword with higher likelihood than the transmitted codeword can be used as a lower bound. These bounds are very tight at lower error probability regime. Our simulation results indicate that these two bounds converge at a word error probability of 10^{-4} . From our results, we conclude that the difference between an ML decoder and a MAP-based iterative decoder is less than 0.2 dB at this word error probability and the block size we simulated. Because GDA produces the lower bound, it achieves the ML error performance in such regime.

VII. Reference :

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," *Proc. IEEE Int. Conf. Commun. '93*, pp. 1064-1070, May 1993.
- [2] Shu Lin and Daniel J. Costello, Jr., *Error Control Coding: Fundamentals and applications, Second Edition*, Prentice Hall, Inc., Upper Saddle River, NJ, 2004.
- [3] S. Benedetto and G. Montorsi, "Unveiling Turbo-Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Transactions on Information Theory*. vol. 43, no.2, pp. 409-428, March 1996.
- [4] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," *TDA Progress Report 42-122, April-June 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 44-45. August 15, 1995.
- [5] T. M. Duman and M. Salehi, "New performance bounds for turbo codes," *IEEE Trans. Commun.* vol. 46, pp. 717-723, June 1998.
- [6] I. Sason and S. Shamai, "Improved Upper Bounds on the ML Decoding Error Probability of Parallel and Serial Concatenated Turbo Codes Via Their Ensemble Distance Spectrum," *IEEE Transactions on Information Theory*. vol. 46, no. 1, pp. 24-47, January 2000.
- [7] J. S. Sadowsky, "A Maximum Likelihood Decoding Algorithm for Turbo Codes," in *Proc. IEEE Global Telecommun. Conf.*, vol. 2, Dallas, TX, Nov 1997.
- [8] K. Shih and D. Shiu, "Perturbed Decoding Algorithm for Concatenated Error Correcting and Detecting Codes System," *17th IEEE Intl. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, Sept. 2006.
- [9] H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, The University of Michigan Press, 1975.
- [10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Publishing Company, 1989.
- [11] Third Generation Partnership Project (3GPP), *Technical Specification 25.212: Multiplexing and Channel Coding (Frequency Division Duplex Mode)*, ver. 6.8.0, June, 2006.
- [12] M. Eroz and A. R. Hammons Jr, "On the design of prunable interleavers for turbo codes," *Proc. IEEE 49th VTC*, vol. 2, pp. 1669V1673, July 1999.
- [13] J. Hagenauer, E. Offer, and L. Papke "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Transactions on Information Theory*. vol. 42, no. 2, pp. 429-445, March 1996.

行政院國家科學委員會補助團隊參與國際學術組織會議報告

96 年 7 月 16 日

報告人姓名	許大山	服務機構	台灣大學電信 工程研究所	職稱	副教授
會議時間	自 2007 年 6 月 24 日 至 2007 年 6 月 29 日		地點(國、州、城市)	法國、尼斯	
會議正式 名稱	中文：2007 IEEE 消息理論國際研討會 英文：2007 IEEE International Symposium on Information Theory				
發表論文 題目	中文： 使用基因演算法之渦輪碼最大概度解碼的理論與效能分析 英文： Theory and Performance of ML decoding for Turbo Codes using Genetic Algorithm				

一、參加會議經過

2007 年消息理論國際研討會訂於西元 2007 六月 24 至 29 日在法國南部的尼斯舉

辦，主辦單位為美國電子電機工程師協會(IEEE)，會議討論主題包括：

- | | |
|---------------------------------------|--------------------------------------|
| 1. Coding theory and practice | 2. Multi-terminal information theory |
| 3. Communication theory | 4. Pattern recognition and learning |
| 5. Compression | 6. Quantum information theory |
| 7. Cryptography and data security | 8. Sequences and complexity |
| 9. Detection and estimation | 10. Shannon theory |
| 11. Information theory and statistics | 12. Signal processing |
| 13. Information theory in networks | 14. Source Coding |

這些主題研究內容將於五天內全以專題演講討論方式呈現每位研究者的研究成果，而本次會議除了有如此豐富的專題討論內容外，另外會議期間的每一天都有安排一場精采的 Plenary talk。這些 Plenary talks 內容相當精采，由淺入深為每位在場的研究學者介紹各個主題目前的研究概況與過往的歷史背景，在講演過程中，所有與會的人員也都給與熱烈的迴響，許多相同或不同領域的學者教授都提出個別的問題與評語。這些演講最令人期待與振奮的是 Shlomo Shamai 的演講，Shlomo Shamai 是消息理論界的一位大師，與會者無一不聚精會神的聆聽他的講演，感受他非凡的風采。除了參與 Plenary talks 外，我們也聽取其他先進學者的最新研究成果，主題包括 Multi-User MIMO、Iterative decoding、Relay channels: capacity and side information、Turbo codes... 等等主題。

我們的論文“使用基因演算法之渦輪碼(Turbo codes)最大概度解碼的理論與效能分析”隸屬於 Turbo codes 單元，上台報告時也引起了許多聆聽的研究學者的迴響，會議的主持人就是渦輪碼的發明人 Claude Berrou，與會者紛紛提出自己的問題與想法，希望透過這樣的討論，讓此論文更加完備與豐富。此行我們也獲得了許多新的想法來延續我們這個研究題目，譬如說，我們想到要去設計一個新的 soft turbo encoder 來執行基因解碼演算法中 re-encode 的動作，這個新的方法有可能會讓我們針對渦輪碼所提出的基因解碼演算法的效能會更好，至於這個 soft turbo encoder 的工作原理，我們目前仍在研究與實驗階段，但不失是一個好的嘗試，也是此行重大收穫之一。

二、 與會心得

ISIT 是消息理論領域最大也最具盛名的學術會議，這樣的會議將分屬於世界各地的研究學者聚合在一起討論，並將彼此最新的研究成果與創新想法匯流在一起，也彼此互相挑戰、互相砥礪，增進彼此的研究，滿足每位研究學者的求知渴望。

參加這次會議所獲得的知識與新的研究創意非常豐富，將這些想法與目前自身的研究做一整合與創新的突破，期待能為國內的學術研究更進一份心力。這次的會議讓我們有機會與其他國家學術學者們的交流，也透過這樣的場合，我積極的邀請優秀的學者來台大做專題的演講，渦輪碼的發明人 Claude Berrou 也給予善意回應，並允諾若是時間許可也願意來台大參訪。

三、 考察參觀活動（無是項活動者省略）

無

四、 建議事項

首先感謝行政院國家科學委員會的經費補助，也因有此補助讓我的研究團隊能在經費無慮的情況下參與此次國際性的會議，並將研究團隊的研究成果發表至國際。我的一點建議、也是我的希望是像這樣的團隊參與國際會議計費的補助能持續下去且增加補助額度，這樣也能鼓勵我們國家優秀的研究學者們持續從事創新工作的研究，也能持續不間斷的增加國際性的視野，並藉此培養國家優秀且具有國際觀的人才。

五、攜回資料名稱及內容

1. Proceeding 一本 (會議流程與論文摘要)
2. 會議光碟 一份

六、其他

無

行政院國家科學委員會補助團隊參與國際學術組織會議報告

96 年 7 月 13 日

報告人姓名	薛存志	服務機構	台灣大學電信 工程研究所	職稱	碩二
會議時間	自 2007 年 6 月 24 日 至 2007 年 6 月 29 日		地點（國、州、城市）	法國、尼斯	
會議正式 名稱	中文：2007 IEEE 消息理論國際研討會 英文：2007 IEEE International Symposium on Information Theory				
發表論文 題目	中文： 使用基因演算法之渦輪碼最大概度解碼的理論與效能分析 英文： Theory and Performance of ML decoding for Turbo Codes using Genetic Algorithm				

一、 參加會議經過

本次 IEEE ISIT2007 消息理論國際研討會是在消息理論上最具盛名的年會，主要是由美國電子電機工程師協會(IEEE)所贊助舉辦，我與指導教授許大山老師搭乘不同班飛機於同一日 24 日抵達法國尼斯，法國尼斯是一個聞名的渡假聖地，尼斯是南法著名的蔚藍海岸里維耶拉最重要的觀光城市，市區街道整齊乾淨且房舍建築相當的考究，且市區中的豪華大型飯店林立，如此優美的環境使得與會者可以用輕鬆的心情迎接這五天緊湊的會議行程。

這次的會議主要的討論主題包括下列幾項：

- | | |
|---------------------------------------|--------------------------------------|
| 1. Coding theory and practice | 2. Multi-terminal information theory |
| 3. Communication theory | 4. Pattern recognition and learning |
| 5. Compression | 6. Quantum information theory |
| 7. Cryptography and data security | 8. Sequences and complexity |
| 9. Detection and estimation | 10. Shannon theory |
| 11. Information theory and statistics | 12. Signal processing |
| 13. Information theory in networks | 14. Source Coding |

本次年會中的第一場 plenary talk 是由 *Michelle Effros* 來進行報告，報告的題目是：*Network Source Coding: Pipe Dream or Promise?* 此一演講的內容得到大家的迴響，許多相同或不同領域的學者教授都提出個別的意見與評語。今日有許多主題在今天於會場不同的會議室進行研討，而我的論文也是在今天上台發表，報告隸屬於 Turbo Code 專題，這次的報告由我的指導老師主講。

第二天的 plenary talk 是由 *Shlomo Shamai* 主講，報告的題目是：*Reflections on the Gaussian Broadcast Channel: Progress and Challenges*，*Shlomo Shamai* 是目前消息理論的一位大師，風采非凡。接下來的陸續幾天也有個別的、精采的 plenary talk 引起大家的迴響，這些 talk 幫大家由淺入深的介紹數個領域的研究概況，增長了我們不少的知識，了解了世界各地的學者教授在做些什麼主題的研究，也砥礪了我們的研究。

二、與會心得

學生本次參加 ISIT2007 獲益匪淺，除了可以再討論過程中交流各研究者的觀念與想法外，也可檢視自身研究的廣度與深度，是否能有在進一步的空間。本次大會辦得非常成功，也傳達了許多重要研究主題與未來的研究方向，對於能參與此會議並且發

表論文學生覺得非常榮幸。

這也是我第一次出國參加國際會議，結束後才發現有許多準備不足需要改進的地方。首先會議前對行程的規劃不夠詳細，再者我個人沒有主動與其他演講者、研究者進行交流十分可惜。有了這次的經驗，下次在參加國際會議前，會先利用年會的網站進行詳細的行程規劃，並積極主動的向與會者請益和學術討論，以及與各地的研究生討論，交換彼此的研究生活，拓展自己的國際觀，也作為將來生涯規劃的參考。此外，英語是學術研究共通的語言，我需要在更加強我的英語能力，才能夠更清楚明白的表達自己的意見。

三、考察參觀活動（無是項活動者省略）

無

四、建議事項

感謝學校與老師給我這個機會完成這次的國外研討會之行，讓我於活動過程中學習到許多相關的專業知識與增廣見聞，並在經費上給我許多補助。對於出國補助的部分，學生希望國科會和學校能夠持續協助支持研究所碩博士班學生積極參與國際研討會，這樣有助於培養研究所學生的國際觀與語文能溝通能力。

五、攜回資料名稱及內容

這次研討會學生共帶回以下資料：

1. Proceeding 一本（會議流程與論文摘要）

2. 會議光碟 一份

六、其他

無

Theory and Performance of ML Decoding for Turbo Codes using Genetic Algorithm

Tsun-chih Hsueh and Da-shan Shiu
 Graduate Institute of Communication Engineering
 Department of Electrical Engineering
 National Taiwan University
 Taipei, Taiwan

Email: r94942036@ntu.edu.tw, dsshiue@cc.ee.ntu.edu.tw

Abstract—Although yielding the lowest error probability, ML decoding of turbo codes has been considered unrealistic so far because efficient ML decoders have not been discovered. In this paper, we propose the Genetic Decoding Algorithm (GDA) for turbo codes. GDA combines the principles of perturbed decoding and genetic algorithm. In GDA, chromosomes are random additive perturbation noises. A conventional turbo decoder is used to assign fitness values to the chromosomes in the population. After generations of evolution, good chromosomes that correspond to decoded codewords of very good likelihood emerge. GDA can be used as a practical decoder for turbo codes in certain contexts. It is also a natural multiple-output decoder. The most important aspect of GDA, in our opinion, is that one can utilize GDA to empirically determine a lower bound on the error probability with ML decoding. Our results show that, at a word error probability of 10^{-4} , GDA achieves the performance of ML decoding. Using GDA, we establish that an ML decoder only slightly outperforms a MAP-based iterative decoder at this word error probability for the block size we used and the turbo code defined for WCDMA.

I. INTRODUCTION

Since its invention [1], turbo codes have received a lot of interest. Turbo codes are known to achieve a performance very close to the channel capacity. Due to this strong performance and manageable decoder complexity, turbo codes have been adopted as options for forward error correction in many recent communications standards such as 3GPP and 3GPP2. Over the years, researchers have answered questions such as error floor and interleaver optimization.

Practical decoders for turbo codes are iterative decoders based on message-passing principles [2]. These decoders are suboptimal from a word error probability perspective. Although yielding the lowest word error probability, ML decoding of turbo codes has been considered unrealistic so far because efficient ML decoders have not been discovered. In the past several years, performance of ML decoding of turbo codes has drawn more attention. In [3] and [4], the union bound on the average error probability over the ensemble of all interleavers was investigated. They are subsequently improved upon [5] [6]. However, these results are valid for the average performance of a set of turbo codes; they cannot be used to establish the performance of a *given* specific turbo code. In

[7], a decoding algorithm based on two component list Viterbi decoders was introduced. It could only yield ML decoding of turbo codes for an infinite list size.

In [8], the "perturbed decoding algorithm" (PA) was proposed for a concatenated coded system consisting of an inner error correcting code and an outer error detecting code. In PA, multiple highly likely candidate codewords are obtained by feeding the inner decoder with slightly perturbed versions of the actual received signal. The candidate codewords are subsequently *validated* by an outer decoder. Among the validated codewords, the one with the highest likelihood is declared the final decoder result. As we will demonstrate in this paper, one can adapt the perturbed decoding concept for use with turbo codes. Unfortunately, due to the large code block size used by turbo codes, the direct application of PA to turbo codes so can require an enormous complexity.

Genetic algorithms (GA) constitute a class of optimization techniques that are based on the general laws of natural selection and genetics [9], [10]. It has been applied to numerous intractable optimization problems. In this paper, we present the novel Genetic Decoding Algorithm (GDA) which combines the principles of GA and PA. In GDA, one maintains a pool of additive perturbation noises as the chromosomes. A conventional decoder is used to give scores to these chromosomes. Specifically, one first adds a chromosome to the received signal. The perturbed signal is passed through a conventional decoder to generate a corresponding decoded codeword. The likelihood of the codeword is assigned as the score of the corresponding chromosome. Chromosomes of low scores are eliminated from the population. After several generations of evolution, good chromosomes that correspond to codewords of very good likelihoods emerge.

Although GDA is trivially guaranteed to produce decoded sequences of higher likelihood than a conventional turbo decoder, at its current form the amount of computation required for GDA is still too large for most practical applications. The critical importance of GDA, in our opinion, lies in the fact that it can be used to *empirically* (i.e., via simulation) establish the fundamental ML performance of a given turbo code. As will be shown later in this paper, for the turbo codes we simulated, GDA can produce ML decoding performance at a word error probability of 10^{-4} and lower. In those applications where the computation load of GDA is acceptable, our results indicate

† This work is supported by the National Science Council of Taiwan under Grant 95-2219-E-002-019.

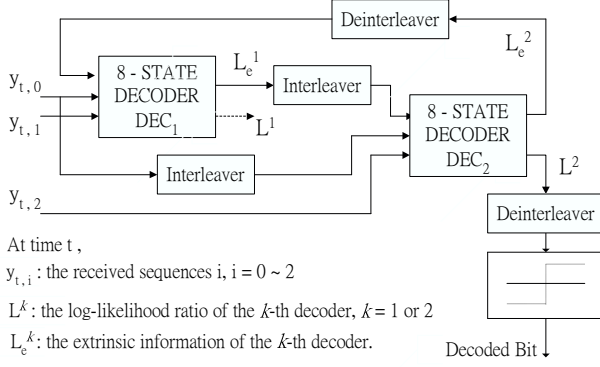


Fig. 1. Conventional iterative decoding algorithm for turbo codes.

that GDA can outperform conventional MAP decoder by a gain of 0.05 to 0.3 dB.

The rest of this paper is organized as follows. In Section II, we briefly review the conventional iterative decoding algorithms for turbo codes and the fundamental concepts of GA and PA. In Section III, we present in detail the concept and procedure of GDA for turbo codes. In Section IV, our simulation results of ML decoding bounds for turbo codes produced by GDA are presented. The parameters and procedures for GDA are also discussed. Concluding remarks are given in Section V.

II. BACKGROUND

A. Maximal-a priori (MAP) Iterative Decoding Algorithm

In this paper, by “turbo codes” we specifically mean parallel concatenated convolutional codes (PCCC). A PCCC is consisted of a parallel concatenation of two constituent recursive convolutional codes and a specific interleaver. For example, a rate 1/3 PCCC employed by WCDMA [11] is consisted of two identical constituent encoders and an interleaver. The two encoders share the same generator polynomial, which is given by

$$g(D) = \begin{bmatrix} 1 & \frac{1 + D + D^3}{1 + D^2 + D^3} \end{bmatrix}. \quad (1)$$

These two constituent encoders are separated by a prunable prime interleaver [12].

As illustrated in Figure 1, a conventional MAP iterative decoder is comprised of two constituent decoders linked by the interleaver. These two constituent decoders pass extrinsic information back and forth. Decoder output is obtained after several iterations. Although many other forms of decoders exist, in this paper we only consider the MAP decoder due to its excellent performance.

B. Genetic Algorithm

To apply GA to an optimization problem, the first and sometimes the most critical step is to formulate a good genetic representation. Once a genetic representation is chosen, a typical genetic algorithm (GA) consists of three fundamental operations: selection, genetic operation, and replacement. The

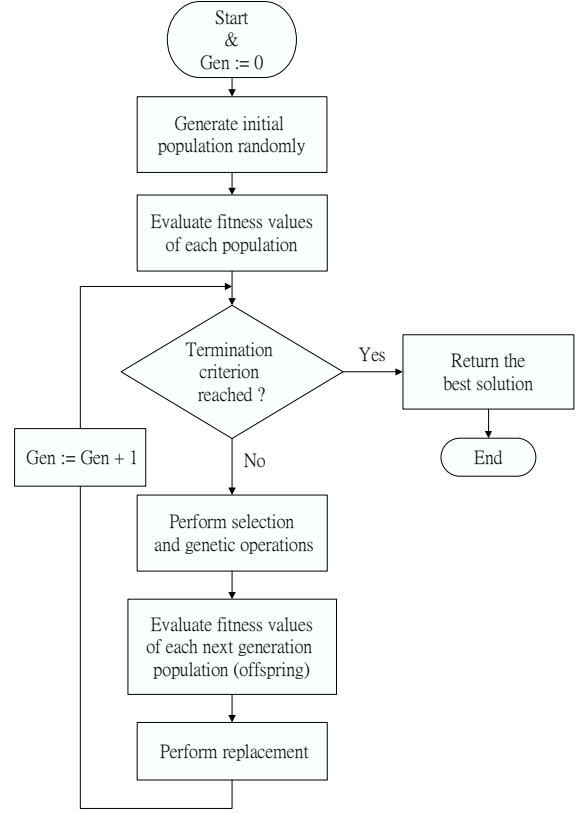


Fig. 2. Flowchart of GA.

selection step is a process that selects members from a population into the mating pool. The genetic operation includes both crossover and mutation. Crossover combines the subparts of the chromosomes from two parents to produce offsprings. Mutation, on the other hand, introduces novel variations into the chromosomes. Replacement is the step that eliminates weak members in order to remain a population of high fitness. As mentioned in [9], [10], a typical GA procedure can be described as follows:

- 1) Initially, a population is generated randomly.
- 2) Evaluate the fitness values for each individuals of the population by calculating some objective function.
- 3) Select from the population some parents to produce the next generation of population (offsprings) by the genetic operations.
- 4) Evaluate the fitness values of offsprings in a similar fashion to their parents.
- 5) According to a chosen replacement strategy, some parents and some offsprings survive. The survivors form the population of the subsequent generation.
- 6) Check whether the desired termination criterion is reached or not. If not, return to step 3 and repeat the process. Otherwise, stop and produce the final solution.

We can refer to the flowchart in Figure 2 for the detailed course of the execution of GDA. Hereafter, we will tailor the objective function of the fitness value evaluation block and the termination criterion for the specific goal of decoding a turbo code.

C. Perturbed Decoding Algorithm

Perturbed decoding algorithm (PA) is a decoding algorithm for concatenated error correction and error detection codes. In PA, the original received signal $\mathbf{y}$ is first decoded by the inner decoder. The decoded result is subsequently validated by an outer decoder. If the outer decoder regards the decoded result as invalid, an additive perturbation noise will be randomly generated. By adding the noise to $\mathbf{y}$, one creates a perturbed received signal. This perturbed signal is decoded by the inner decoder. The decoder output, which may be different from the initial decoder output, is again validated by the outer decoder. This process is repeated until either a valid codeword is declared by the outer decoder or when some termination criterion is met. It is a practical strategy to produce multiple decoder outputs with a simple decoder.

III. GENETIC DECODING ALGORITHM

A. Perturbed Decoding for Turbo Codes

The principle of PA can be extended to decoding turbo codes. Let us consider a BPSK-modulated coded transmission system described as follows. The information bit sequence is $\mathbf{d} = (d_1, d_2, \dots, d_K)$, where $d_k \in \{0, 1\}$. These bits are encoded to produce the transmitted symbols $\mathbf{x} = (x_1, x_2, \dots, x_N)$, where $x_n \in \{1, -1\}$. The received signal $\mathbf{y} = (y_1, y_2, \dots, y_N)$ is disturbed by real-valued additive white Gaussian noise of variance $N_0/2$:

$$y_n = x_n + w_n, \quad n = 1, 2, \dots, N, \quad (2)$$

where w_n are i.i.d. $\mathcal{N}(0, N_0/2)$. Given $\mathbf{y}$, the likelihood of a potential transmitted information sequence $\mathbf{d}'$, whose corresponding coded symbol sequence is $\mathbf{x}(\mathbf{d}')$, is

$$L(\mathbf{d}') = \prod_{n=1}^N \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(y_n - x_n(\mathbf{d}'))^2}{N_0}\right). \quad (3)$$

Consider some decoder for turbo code. It maps the received signal $\mathbf{y}$ into some $\mathbf{d}$. If we perturb the input signal by an additive signal ν , the decoder can produce a different output $\mathbf{d}_\nu$ whose corresponding likelihood is $L(\mathbf{d}_\nu)$. As in the case for PA, by feeding the decoder with randomly perturbed copies of received signal $\mathbf{y}$, one can produce a pool of possible decoded sequences. The highest-likelihood decoded sequence eventually converges to the ML decoder output $\mathbf{d}_{ML}$, provided that:

- 1) A set of perturbation signals, $\mathcal{V}$, exists such that for any perturbation signal $\nu \in \mathcal{V}$, this decoder maps $\mathbf{y} + \nu$ into $\mathbf{d}_{ML}$.
- 2) According to the procedure to generate the perturbation signal ν , the probability of selecting a signal ν that belongs to $\mathcal{V}$ is nonzero.

Whether the first condition is met depends on the particular decoder. For most reasonable decoders though the condition is met trivially. Given that this condition is met, if the set $\mathcal{V}$ is not problematic, the second condition can be easily met by choosing i.i.d. white Gaussian entries as perturbation noise ν_n .

Therefore, one expects that, when given a decoder that satisfies the first condition, the performance of ML decoding

can be empirically determined by executing the perturbed decoding algorithm. Unfortunately, due to the large block size used by turbo codes, doing so requires an impractically enormous amount of computation!

B. Genetic Decoding for Turbo Codes

In order to avoid the complexity of PA for turbo codes, instead of randomly generating perturbation signals, we wish to extract useful information from prior perturbation-decoding attempts and subsequently apply such information to the generation of new perturbation signal. To achieve this, we combine the genetic algorithm with the PA.

In our proposed GDA for turbo codes, a chromosome is a perturbation signal ν . The genes are the individual additive perturbation noise components. Our goal is to find at least one chromosome which, when added to the receive signal $\mathbf{y}$, induces the decoder to produce the ML output $\mathbf{d}_{ML}$. To this end, the fitness of a chromosome ν is defined to be the likelihood of the corresponding decoded sequence, $d(\nu)$:

$$FV(\nu) = L(\mathbf{d}(\nu)), \quad \text{where } \mathbf{y} + \nu \xrightarrow{\text{decode}} \mathbf{d}(\nu). \quad (4)$$

The above definition of genetic information and fitness evaluation is the core concept of GDA for turbo codes. With them, appropriate definitions can be found for other critical functional blocks in GA: parent selection, cross over, and mutation. The overall procedure of GDA then follows that of a typical GA. In Section IV, we will present our definitions of these functional blocks.

C. ML bound using GDA

GDA is an effective tool to establish a lower and an upper bound of ML decoding even though it itself is not an ML decoder. By definition, the error performance of GDA itself is an upper bound, because no decoder offers better word error probability than ML decoding.

When any decoder produces an output with a likelihood higher than that of the actually transmitted encoded sequence, the ML decoder must either decode to this erroneous output or some other sequence with even higher likelihood. Thus we can conclude that ML decoder must also make a decoding error on the same received signal. On the other hand, when a decoder produces an output with a likelihood less than or equal to that of the actually transmitted encoded sequence, we can't be certain whether some other sequence with likelihood higher than that of the actually transmitted encoded sequence exists. If we assume an ML decoder will decode correctly over all of these signals, we would have over-estimated the performance of ML decoding. In light of this, a lower bound of ML decoding can be empirically established using a practical decoder to be the probability when the said decoder decodes into a sequence of higher likelihood than that of the actually transmitted encoded sequence.

Note that unless the said decoder does behave very similarly to an ML decoder, this lower bound tends to be very poor. For example, if an MAP decoder is used, the lower bound is entirely useless. With GDA as the decoder, however, the two bounds are very close.

D. Complexity Aspects for GDA

GDA involves repeated uses of a turbo decoder to compute fitness values for individuals in a population. The complexity of GDA can be several orders of magnitude higher than that of a conventional turbo decoder. Nevertheless, in certain context, GDA can be used as a practical decoder. For instances, when a turbo code is concatenated by a CRC, one can first pass received signal through a conventional turbo decoder. GDA is activated only when the conventional turbo decoder fails. The average decoding complexity of GDA is weighted down by the first-pass error probability.

IV. SIMULATION RESULTS

A. Simulation Setup

We use the turbo code defined in the WCDMA specification in our simulation [11]. The interleaver used in the specification is prunable prime interleaver [12]. For a block size of K , the actual code rate is $K/(3K + 12)$. Encoded bits are modulated using BPSK modulation and transmitted over an AWGN channel. The “symbol-by-symbol” MAP rule [13] is used for the turbo decoding algorithm that produces the fitness values. Eighteen full iterations are used for the MAP decoder.

B. Parameters and Procedures for GDA

In this paper, our goal is to demonstrate the potential performance of GDA with manageable complexity. We do not claim that the choice of parameters below are optimal in terms of minimizing complexity. Actually, we surmise that our results can be materially improved if we could execute more computation.

From Section III-B, we realize that in order to ensure a nonzero probability of obtaining a perturbation signal that induces the ML-decoding output, it is prudent to allow the perturbation signal to take any values. In light of this, we produce the first generation of perturbation noises as i.i.d. zero mean Gaussian vectors. The signal-to-perturbation noise ratio is defined as

$$SPNR = 10 \log_{10} \left(\frac{E_b}{\sigma_p^2} \right). \quad (5)$$

It is suggested in [8] that as long as the power of the perturbation noise is chosen reasonably, the expected number of attempts until an “ideal noise” occurs is a very weak function of SPNR. Based on empirical observation, we chose 15 dB to be the SPNR used in our simulation.

It is known that the quality of the initial population strongly affects the success of the subsequent evolution. Therefore, we randomly generate 10000 original populations and keep only the fittest 100 individuals to be the initial population. The crossover rate is selected at 80 percent, i.e. a randomly selected 80 percent of the population in a generation participates in the production of new offsprings. When two chromosomes mate, we use uniform crossover to construct offsprings. A gene is copied from each parent with equal probability. On average, each parent donates 50 percent of its genetic material to its offspring. Two offsprings are generated for a pair of parents. Only the two fittest among the four members of a family

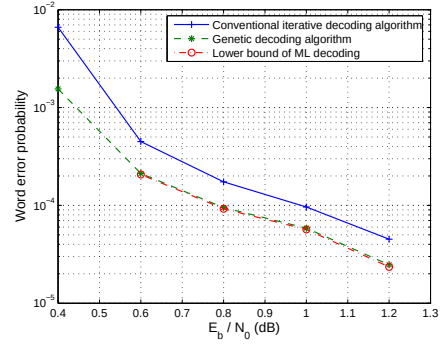


Fig. 3. Word error probability for turbo code. The block size is 2560.

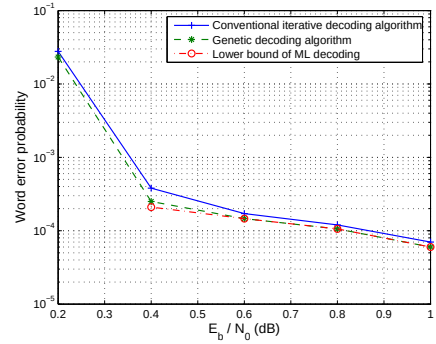


Fig. 4. Word error probability for turbo code. The block size is 5120.

survive to the next generation. Each gene of a survivor is mutated independently with a small mutation probability of 0.5 percent. Mutation replaces the gene by a randomly generated perturbation noise with the same SPNR. The 20 percent of the population that does not participate in the mating process survives to the next generation by default.

In our simulation, we terminate the GDA if (1) the predefined number of generation is reached, or (2) a predefined percentage of population converges to one fitness value which is the maximal value among all populations in any generation. We choose these two thresholds at 300 and 20 percent, respectively.

In these results we also present the lower bound for ML decoding computed using the concept of III-C. Recall that the GDA output can be either the originally transmitted codeword with likelihood $L(\mathbf{d})$, a codeword with likelihood lower than $L(\mathbf{d})$, or a codeword with likelihood higher than $L(\mathbf{d})$. In order to produce a tight lower bound, when the first two cases occur, we re-execute GDA for another two rounds.

C. Performance of GDA

The word error probability for turbo codes given information block sizes of 2560 and 5120 bits per block is shown in Figure 3 and Figure 4, respectively. The error performance of conventional decoder, GDA, and the lower bound of ML performance derived from the application of GDA are shown. As can be seen, at a word error probability of 10^{-4} , GDA achieves the error performance of ML decoding. With the help

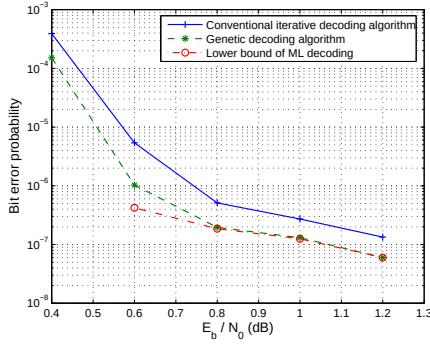


Fig. 5. Bit error probability for turbo code. The block size is 2560.

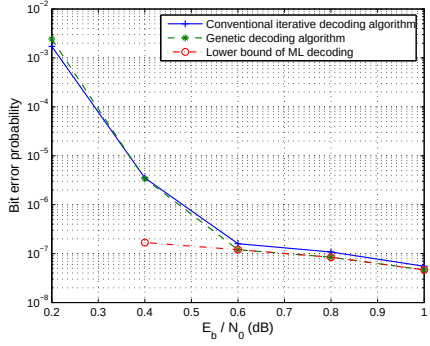


Fig. 6. Bit error probability for turbo code. The block size is 5120.

of GDA, here we can determine a very important result empirically: the penalty of using a MAP-based iterative decoder, from a minimization of word error probability perspective, is less than 0.2 dB at a word error probability of 10^{-4} for the turbo codes we consider.

In addition to word error probability, we also show the information bit error probability for the two block sizes in Figure 5 and Figure 6, respectively. Again the error performance of the conventional MAP decoder, GDA, and the lower bound of ML performance derived from the application of GDA are contrasted. One finds that the bit error probability follows that of the word error probability closely. Hence we surmise that the number of bit errors per word error for conventional decoders and for GDA are of approximately the same order.

V. CONCLUSION

In this paper, we present a novel decoding algorithm, the Genetic Decoding Algorithm (GDA), for use with turbo codes.

In GDA, we propose to use random perturbation noises as the genes. A conventional decoder is used to assign fitness values to chromosomes. Specifically, one adds the noise pattern (chromosome) to the received signal first. A well chosen decoder decodes the perturbed signal and generates a corresponding decoded codeword. The likelihood of the codeword is assigned to be the fitness value of the corresponding chromosome. Chromosomes of low fitness values are eliminated from the population. After several generations of evolution, good chromosomes that correspond to codewords of very good likelihoods emerge.

GDA can be used as an empirical tool to establish the ML decoding performance for a given specific turbo code. The performance of GDA itself can serve as an upper bound. The probability where GDA finds a codeword with higher likelihood than the transmitted codeword can be used as a lower bound. These bounds are very tight at lower error probability regime. Our simulation results indicate that these two bounds converge at a word error probability of 10^{-4} . From our results, we conclude that the difference between an ML decoder and a MAP-based iterative decoder is less than 0.2 dB at this word error probability and the block size we simulated. Because GDA produces the lower bound, it achieves the ML error performance in such regime.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," *Proc. IEEE Int. Conf. Commun.* '93., pp. 1064-1070, May 1993.
- [2] Shu Lin and Daniel J. Costello, Jr., *Error Control Coding: Fundamentals and applications, Second Edition*, Prentice Hall, Inc., Upper Saddle River, NJ, 2004.
- [3] S. Benedetto and G. Montorsi, "Unveiling Turbo-Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Transactions on Information Theory*, vol. 43, no.2, pp. 409-428, March 1996.
- [4] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," *TDA Progress Report 42-122, April-June 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 44-45, August 15, 1995.
- [5] T. M. Duman and M. Salehi, "New performance bounds for turbo codes," *IEEE Trans. Commun.* vol. 46, pp. 717-723, June 1998.
- [6] I. Sason and S. Shamai, "Improved Upper Bounds on the ML Decoding Error Probability of Parallel and Serial Concatenated Turbo Codes Via Their Ensemble Distance Spectrum," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 24-47, January 2000.
- [7] J. S. Sadowsky, "A Maximum Likelihood Decoding Algorithm for Turbo Codes," in *Proc. IEEE Global Telecommun. Conf.*, vol. 2, Dallas, TX, Nov 1997.
- [8] K. Shih and D. Shiu, "Perturbed Decoding Algorithm for Concatenated Error Correcting and Detecting Codes System," *17th IEEE Intl. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, Sept. 2006.
- [9] H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, The University of Michigan Press, 1975.
- [10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Publishing Company, 1989.
- [11] Third Generation Partnership Project (3GPP), *Technical Specification 25.212: Multiplexing and Channel Coding (Frequency Division Duplex Mode)*, ver. 6.8.0, June, 2006.
- [12] M. Eroz and A. R. Hammons Jr, "On the design of prunable interleavers for turbo codes," *Proc. IEEE 49th VTC*, vol. 2, pp. 1669V1673, July 1999.
- [13] J. Hagenauer, E. Offer, and L. Papke "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, March 1996.