Box-minus operation and application in sum-product algorithm

S Tong  
*Xidian University, sheng@uow.edu.au*

P Wang  
*Xidian University*

D Wang  
*Xidian University*

X Wang  
*Xidian University*

Follow this and additional works at: [https://ro.uow.edu.au/eispapers](https://ro.uow.edu.au/eispapers)

Part of the Engineering Commons, and the Science and Technology Studies Commons

**Recommended Citation**

Tong, S; Wang, P; Wang, D; and Wang, X, "Box-minus operation and application in sum-product algorithm" (2005). *Faculty of Engineering and Information Sciences - Papers: Part A. 1076.*  

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Box-minus operation and application in sum-product algorithm

Abstract
A new formula for box-minus operation, which separates the box-minus operation into sign and reliability operations, was derived. Its application in the sum-product algorithm (SPA) was also investigated. Both box-plus operation and box-minus operation were divided into sign operation and reliability operation. The results show that on using the box-minus operation, SPA can be implemented with less decoding latency and complexity.

Keywords
product, algorithm, sum, application, operation, minus, box

Disciplines
Engineering | Science and Technology Studies

Publication Details

This journal article is available at Research Online: https://ro.uow.edu.au/eispapers/1076
Box-minus operation and application in sum–product algorithm

S. Tong, P. Wang, D. Wang and X. Wang

A new expression for box-minus operation, i.e. the inverse of box-plus operation, is derived, with which the box-minus operation can be implemented by a small look-up table. Its application in the sum–product algorithm is investigated.

Introduction: For the practical application of low density parity check (LDPC) codes [1], the implementation of the associated decoding algorithm, i.e. the sum–product algorithm (SPA), should be carefully considered. As for this problem, there have already been some results [2–5]. An efficient parallel implementation in the log-likelihood ratio (LLR) domain for SPA is proposed in [4] leading to less decoding complexity and latency, which involves two core operations, i.e. box-plus operation [6] and its inverse, called box-minus operation.

In this Letter we derive a new expression for the box-minus operation, which separates the box-minus operation into sign and reliability operations and is more suitable for practical implementation.

Box-plus and box-minus operations: Denote the LLR of a binary random variable \( u \in \{ \pm 1 \} \) as \( L(u) = \log(\Pr(u = +1)/\Pr(u = -1)) \). Then, for two independent binary random variables \( u \) and \( v \), the box-plus operation is defined as follows [6, 4]:

\[
L(u \oplus v) = L(u) \odot L(v) = L(u) \odot L(v) = \log \left( \frac{1 + e^{L(u) + L(v)}}{1 + e^{L(u) - L(v)}} \right)
\]

\[
= \text{sign}(L(u)) \text{sign}(L(v)) \times \min(L(u), L(v)) + \log \left( \frac{1 + e^{-\min(L(u), L(v))}}{1 + e^{-\max(L(u), L(v))}} \right)
\]

(1)

where \( w = u \oplus v \) and \( \oplus \) denotes binary XOR operation. Equation (1) can be further written as

\[
L(u \oplus v) = L(u) \odot L(v) = \text{sign}(L(u)) \text{sign}(L(v)) \times \min(L(u), L(v)) + \log \left( \frac{1 + e^{-\min(L(u), L(v))}}{1 + e^{-\max(L(u), L(v))}} \right)
\]

(2)

In (2), the term next to the multiplication sign is non-negative, and thus the box-plus operation is divided into sign and reliability operations. Then the box-plus operation reduces to the computation of the function \( g(x) = \log(1 + e^{-|x|}) \), which can be implemented by a small look-up table [4].

\[
L(u) = L(v) \odot L(v) = \text{sign}(L(u)) \text{sign}(L(v)) \times \left\lfloor \min(L(u), L(v)) \right\rfloor + \log \left( \frac{1 + e^{-\min(L(u), L(v))}}{1 + e^{-\max(L(u), L(v))}} \right)
\]

(3)

Similar to the derivation of (2), the box-minus operation can be divided into sign and reliability operations as follows:

\[
L(u) = L(v) = \log \left( \frac{1 - e^{L(u) - L(v)}}{1 - e^{L(u) + L(v)}} \right)
\]

(4)

Since \( |L(u)| < |L(v)| \) (see (2)), the term next to the multiplication sign in the above expression is non-negative. Thus, the box-minus operation reduces to the function \( h(x) = \log(1 - e^{-x}), \ x < 0 \). A similar function \( h(x) = \log(e^x - 1) \) is defined in [4] for the implementation of box-minus operation, which is plotted in Fig. 1. It can be seen that as \( x \to -\infty \), \( h(x) \to -\infty \). Hence, \( h(x) \) is not suitable to be implemented with a look-up table. However, \( h(x) \) is only the case of \( h(x) \) when \( x < 0 \), which can be easily implemented by a small look-up table.

Application of box-minus operation in SPA: Consider a check node \( w \) of degree \( n \). Denote its \( n \) neighbouring variable nodes as \( \{u_k\}_{k=1}^n \). In the horizontal step of SPA, the \( n \) LLRs of

\[
\sum_{k=1}^n u_k \quad (i = 1, 2, \ldots, n)
\]

are calculated [2–4]. Obviously, these LLRs can be calculated as follows:

\[
L\left( \sum_{k=1}^n u_k \right) = \sum_{k=1}^n L(u_k) = L(w) \odot L(u_i)
\]

\[
L(w) = L\left( \sum_{k=1}^n u_k \right) = \sum_{k=1}^n L(u_k)
\]

(5)

In the conventional implementation, a forward–backwardward algorithm is used for the horizontal step, which involves \( 3(n - 2) \) box-plus operations [4]. However, by first calculating \( L(w) \) and then box-minusing \( L(u_k) (k = 1, 2, \ldots, n) \), respectively, to get the \( n \) LLRs, the overall computation is \( (n - 1) \) box-plus operations and \( n \) box-minus operations. Since box-minus operation has almost the same computation complexity as box-plus operation, by the above technique the total computation complexity is reduced. In addition, the decoding latency is reduced from \( O(n) \) to \( O(\log(n)) \) box-plus (or box-minus) operations [4]. Here we call this technique the parallel-excluding (PE) technique, which can be viewed as a message-passing schedule [7]. Denote the SPA implemented with PE as PE-SPA. Fig. 2 shows the bit error rate performance of a randomly constructed \((1008,3,6)\) regular LDPC code, assuming an AWGN channel and BPSK modulation. Here, the box-minus operation in PE-SPA is implemented by a 5-bit look-up table.

Fig. 1 Function \( h(x) = \log(e^x - 1) \)

From (1), the box-minus operation can be defined as

\[
L(u) = L(w) \odot L(v) = \text{sign}(L(u)) \text{sign}(L(v)) \times \left\lfloor \min(L(u), L(v)) \right\rfloor + \log \left( \frac{1 - e^{-\min(L(u), L(v))}}{1 - e^{-\max(L(u), L(v))}} \right)
\]

Conclusion: Both box-plus operation and box-minus operation are divided into sign operation and reliability operation. By its new expression, the box-minus operation can be implemented by a small look-up table. Using the box-minus operation, SPA can be implemented with less decoding latency and complexity.
References