RAIRO-Oper. Res. 50 (2016) 189–209
DOI: 10.1051/ro/2015025

RAIRO Operations Research
www.rairo-ro.org

A NOVEL ANALYTICAL INTEGER OPTIMIZATION METHOD FOR WAVELET BASED SUBBAND CODING

Mahdi Hatam[1] and Mohammad Ali Masnadi-Shirazi[1]

Abstract. In subband coding (SBC), the original signal is decomposed into some frequency subbands and then the total available number of bits is divided between different subbands of the signal. In the most of existing SBC methods, the number of allocated bits can be real and negative, while in practice the number of bits must be integer and nonnegative. In this paper an analytical solution is derived for subband coding with optimum nonnegative integer bit allocation and multi-resolution filter bank (including wavelet filter bank). The analytical solution is applicable for either non-uniform or uniform SBC. A modified discrete bisection algorithm is also proposed which can reduce the computational complexity of searching in a group of discrete functions. The computational complexity of proposed method is lower than the complexity of integer optimization algorithms which are applicable to SBC. Compared to the common SBC algorithms with real-valued bit allocation (in which the number of bits should be rounded), the proposed method has much less quantization error.

Mathematics Subject Classification. 90C10, 91B32, 90B80.

Received July 21, 2014. Accepted June 24, 2015.

1. Introduction

A common data compression approach is to decompose the signal by means of a filter bank and then using the optimal bit allocation, minimize the total quantization mean square error (MSE) under a fixed total bit rate constraint. This approach is known as subband coding (SBC) [1,2,4,6,10,12,19,21,22,24,25]. SBC is widely used for compression and coding of all types of signal such as speech, image and video. SBC can be implemented using any type of filter bank including wavelet filter bank. In the most SBC approaches the bandwidths of subbands are assumed equal (uniform SBC) [1,19,21] while in few approaches the bandwidths of subbands can be different (non-uniform SBC) [4,25]. Since in multi-resolution wavelet transforms bandwidths of subbands are different, only in the second type of SBC approaches multi-resolution wavelet transform can be used.

In many SBC references, direct form relations are derived for the optimal number of bits of each subband in terms of the variance of signal in different subbands [1,4,19,25]. However, in these direct form relations the number of bits can be real and negative, while in practice the number of bits must be integer and nonnegative. Thus, integer optimization techniques should be used for solving the problem of optimal bit allocation in

Keywords. Wavelet filter bank, subband coding, data compression, integer optimization.

[1] Dept. of Communications and Electronics, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran.
masnadi@shirazu.ac.ir

SBC [5, 8, 15, 17, 20, 23]. This is a serious problem which is addressed by many SBC references. For example in [18] Ma and Rajala point out that:

> "Direct forms provide theoretical results but are often limited by practical issues such as non-integer arithmetic, negativity, and/or mismatched bit assignment. The mismatch implies that the calculated results using the direct form, which could be negative, are different from the quantizers bit rate, which are always positive ..."

In another reference, Aas and Mullis state that [1]:

> "The optimal word length (OWL) (9) [direct form relation] will, in general, not give integer values for the bits, but approximating these values by integers is a standard procedure which usually gives satisfactory results [4]. More seriously, channels for which the product $K_{ii}W_{ii}$ is relatively small may be assigned a negative number of bits, which violates the modeling assumptions ..."

Among the existing integer optimization approaches, the algorithm proposed by Fox in [9] can be used for non-uniform and uniform SBC but this algorithm has relatively high computational complexity for SBC and its complexity grows with the total bit rate. Obtaining the integer bit allocation by rounding the real valued direct form solution needs lower computational complexity compared to Fox algorithm but this procedure is heuristic and does not essentially lead to the optimal solution [11].

In this paper at first, using an analytical discrete optimization approach the solution of SBC problem is derived in terms of the Lagrange multiplier, λ. Then, by applying the constraint of problem, the Lagrange multiplier is removed and an analytical solution is derived for SBC with nonnegative integer bit allocation. The solution is valid for both uniform and non-uniform filter banks including multi-resolution wavelet filter banks.

The computational complexity of proposed method is lower than Fox algorithm. Furthermore, unlike the Fox algorithm, the complexity of the proposed method does not grow as the total number of bits increases.

Using computer simulation, the performance of proposed method is compared to the direct form solution (with rounding). The results show that the proposed method is superior. However, the order of computational complexity of proposed method is not increased compared to the direct form solution. Another computer simulation is carried out to compare the running time of proposed method with that of Fox algorithm. The simulation results confirm that the proposed method is faster and its complexity does not depend on the total number of bits.

The proofs of theorems, lemmas and the computational complexity of algorithms are brought in Appendix A.

2. Problem formulation

Assume that using a filter bank, the original signal is decomposed to p subbands and the signal of ith subband is quantized with b_i bits per sample. In SBC, the aim is to find the optimal number of bits per sample for each subband under a maximum bit rate constraint. The total quantization MSE is obtained from [4, 25]

$$MSE = \sum_{i=1}^{p} \frac{\varepsilon_i \sigma_i^2}{\tau_i m_i} 4^{-b_i}, \tag{2.1}$$

where τ_i is the masking threshold coefficient of ith subband, m_i is the decimation factor of ith subband, σ_i^2 is the variance of signal in ith subband and ε_i is a constant depending on the statistical properties of the signal of ith subband. The problem constraint is

$$\sum_{i=1}^{p} \frac{b_i}{m_i} \leq R,$$

where R is the maximum allowable value of average number of bits per sample. Similar to wavelet and wavelet packet filter banks, we assume that bandwidths of all subbands are integer multiples of the minimum bandwidth of subbands. On the other hand the decimation factor is proportional to the inverse of bandwidth *i.e.* $m_i \propto 1/\Delta_i$

where Δ_i is the bandwidth of ith subband. Thus, the constraint can be written as

$$\sum_{i=1}^{p} (\Delta_i/\Delta_{\min}) \frac{b_i}{m_{\max}} \leq R,$$

$$\sum_{i=1}^{p} k_i b_i \leq R m_{\max},$$

where $\Delta_{\min} = \min_i \{\Delta_i\}$, $m_{\max}$ is the decimation factor of the subband with bandwidth $\Delta_{\min}$ and $k_i = \Delta_i/\Delta_{\min}$ is an integer number. Since k_is and b_is are integer, the constraint can be modified as $\sum_{i=1}^{p} k_i b_i \leq B$ and the optimization problem is formulated as

$$\underset{\{b_1,b_2,\ldots,b_p\}}{\text{Minimize}} \sum_{i=1}^{p} \frac{\varepsilon_i \sigma_i^2}{\tau_i m_i} 4^{-b_i} \quad \text{Subject to:} \quad \sum_{i=1}^{p} k_i b_i \leq B, \tag{2.2}$$

where $B \triangleq \lfloor R m_{\max} \rfloor$ is an integer value.

In the common SBC direct form relations, the optimal real-valued number of bits of ith subband is obtained from the following relation [4, 25]

$$b_i = R + \frac{1}{2} \log_4 \frac{\varepsilon_i \sigma_i^2 / (\tau_i m_i)}{\prod_{j=1}^{p} \left(\varepsilon_j \sigma_j^2 / (\tau_j m_j)\right)^{1/m_j}}. \tag{2.3}$$

In practice, the number of bits cannot be non-integer and the above value should be replaced by an integer near it. Rounding the value obtained from (2.3) to the nearest integer value may cause exceeding the maximum bit rate, R. Thus, to implement the common direct form relations, one should round the value obtained from direct form relation to the nearest integer value smaller than it.

3. Optimization tools

3.1. Analytical discrete optimization approach

Here, we introduce an analytical approach for finding the local minima and maxima of discrete functions. We have presented the general approach in [14] but here we bring a part of this approach which is for one-dimensional discrete functions.

Some Definitions:
Assume that $f[n]$ is a discrete function defined on $\mathbf{D} \subset \mathbb{Z}$.

Define n^* as a local minimum of the discrete function $f[n]$ if n^*, $n^* - 1$ and $n^* + 1$ are in $\mathbf{D}$ and we have $f[n^*] \leq f[n^* - 1]$ and $f[n^*] \leq f[n^* + 1]$.

Define n^* as a local maximum for the discrete function $f[n]$ if n^*, $n^* - 1$ and $n^* + 1$ are in $\mathbf{D}$ and we have $f[n^*] \geq f[n^* - 1]$ and $f[n^*] \geq f[n^* + 1]$.

Denote the floor of real number x by $\lfloor x \rfloor$ which is defined as the largest integer less than or equal to x.

Theorem 3.1. *Assume that $f[n]$ is a discrete function defined on the integer interval $I \subset \mathbb{Z}$ and $f_r(x)$ is a continuous function defined on the connected interval $I_r \subset \mathbb{R}$ so that $I \subset I_r$ and at each integer point in I_r such as n' we have $f_r(n') = f[n']$. Let $x_1, x_2, \ldots, x_m$ be all the solutions of the equation $f_r(x) - f_r(x-1) = 0$. The set $\mathbf{A}$ consisting of the floors of all x_is $(i = 1, 2, \ldots, m)$ plus integer $(x_i - 1)$s (if any) contains all the local minima and maxima of $f[n]$.* ‖

The proof of Theorem 3.1 can be found in Appendix A.1

Remark 3.2. In Theorem 3.1 if some of x_is are integer we have

$$f_r(x_i) = f_r(x_i - 1)$$

$$f[x_i] = f[x_i - 1].$$

Thus, in this case if x_i is a local maximum (minimum) of $f[n]$, then $x_i - 1$ is another local maximum (minimum) of $f[n]$ with the same value.‖

3.2. Lagrange analysis for SBC

Lagrange multiplier technique can be used for integer optimization problems, but there is more limited conditions for optimality of this technique in integer optimization compared to continuous optimization problems [7, 13]. Theorem 3.3 introduces a sufficient condition for optimality of Lagrange multiplier approach.

Theorem 3.3 (from [7]). *Consider the optimization problem:*

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbf{D}}{\text{Minimize}} \ \{ G(\mathbf{x}) \} \\ &\text{Subject to} \ : \ C(\mathbf{x}) \le c, \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ *is a d-element vector and* $G(\mathbf{x})$ *and* $C(\mathbf{x})$ *are real valued functions defined on* $\mathbf{D}$. *Assume that* $\mathbf{x}^*(\lambda)$ *is the solution of the problem*

$$\underset{\mathbf{x}\in D}{Minimize} \ \{ G(\mathbf{x}) + \lambda C(\mathbf{x}).\}$$

If a $\lambda^* \ge 0$ *is found that satisfies the equality* $C(\mathbf{x}^*(\lambda^*)) = c$ *then* $\mathbf{x}^*(\lambda^*)$ *is the solution of the problem* (3.1) *(see the proof in* [7]*).*‖

According to Theorem 3.3, to solve the SBC problem (2.2), at first we should find

$$\begin{aligned} &\min_{b_1,b_2,\ldots,b_p} \left\{ \sum_{i=1}^{p} \frac{\varepsilon_i \sigma_i^2}{\tau_i m_i} 4^{-b_i} + \lambda \sum_{i=1}^{p} k_i b_i \right\}. \\ &b_i \in \mathbf{D}, i = 1, \ldots, p \end{aligned} \tag{3.2}$$

Denote the optimum value of b_i in (3.2) by the set function $\{b_i^*(\lambda)\}$ which is related to the bits of ith subband. The reason of defining $\{b_i^*(\lambda)\}$ as a set function is the fact that the problem (3.2) may have more than one solution at each λ (see Rem. 3.2).

When $\mathbf{D} \subset \mathbb{Z}$, depending on the values of k_is, in some cases the relation $\sum_{i=1}^{p} k_i b_i = B$ cannot be satisfied. Let B_1 be the maximum integer value smaller than or equal to B for which the relation $\sum_{i=1}^{p} k_i b_i = B_1$ can be satisfied. Since B_1 is the maximum accessible number of bits, the constraint $\sum_{i=1}^{p} k_i b_i \le B$ can be changed as $\sum_{i=1}^{p} k_i b_i \le B_1$ and hence according to Theorem 3.3, λ^* can be obtained from

$$\sum_{i=1}^{p} k_i \{b_i^*(\lambda^*)\} = B_1.$$

In [9], Fox showed that the solution obtained from his algorithm is equal to the solution of Lagrange unconstrained problem ($\{b_i^*(\lambda)\}$s) for some λ. He also showed that $\{b_i^*(\lambda)\}$s are efficient solution of the main problem which means that $\{b_i^*(\lambda)\}$s are optimal solution of main problem for $B = \sum_{i=1}^{p} k_i \{b_i^*(\lambda)\}$. Furthermore, he showed that in the case $k_1 = k_2 = \ldots = k_p$ (uniform SBC in our application) $\{b_i^*(\lambda^*)\}$s are essentially the optimal solution of main problem.

3.3. Selection algorithm

Selection is the algorithm of finding the member with rank i in a set of N numbers. Best selection algorithms are $O(N)$ [3, 16]. Using the selection algorithm we can also find the subset of i largest (or smallest) members of a set of N numbers with the complexity $O(N)$ (for example by comparing all members to the member with rank i).

Some Definitions:

Define $\Psi(i\,,\,\mathbf{T})$ as the member with rank i (ith large number) in the set $\mathbf{T}$.

Define $\mathbf{\Phi}(i\,,\,\mathbf{T})$ as the subset of i largest members of the set $\mathbf{T}$ and define $\mathbf{\Phi}'(i\,,\,\mathbf{T})$ as the subset of i smallest members of the set $\mathbf{T}$.

Define $\bar{\mathbf{\Phi}}(i_1, i_2, \mathbf{T})$ as the set of members with ranks between i_1 to i_2 in the set $\mathbf{T}$. We have:

$$\bar{\mathbf{\Phi}}(i_1, i_2, \mathbf{T}) = \mathbf{\Phi}(i_2, \mathbf{T}) - \mathbf{\Phi}(i_1 - 1, \mathbf{T})\,, \qquad i_2 > i_1$$

Define $\mathcal{R}(x\,,\,\mathbf{T})$ as the rank of the number x in the set $\mathbf{T}$ (in descending order).

Denote the number of members of any set like $\mathbf{T}$ by $|\mathbf{T}|$.

3.4. Discrete bisection algorithm

Bisection algorithm is a well-known algorithm for finding the roots of continuous functions. Here we present a discrete version of this algorithm. Consider the non-increasing discrete function $f[n]$ and assume that $f[n_1] > 0$ and $f[n_2] \leq 0$ where $n_2 - n_1 = N > 0$. In the discrete bisection algorithm our aim is to find n^* such that

$$f[n^*] > 0 \quad \text{and} \quad f[n^* + 1] \leq 0. \tag{3.3}$$

Algorithm 3.4. Discrete bisection algorithm

Step 1: Set $n_1 = 1$ and $n_2 = N$.
Step 2: Set $\bar{n} = \lfloor (n_1 + n_2)/2 \rfloor$.
Step 3: If (3.3) is satisfied at $n^* = \bar{n}$, then stop the algorithm; $n^* = \bar{n}$ is the solution. Otherwise:
If $f[\bar{n}] > 0$ set $n_1 = \bar{n} + 1$ and go to Step 2.
If $f[\bar{n}] < 0$ set $n_2 = \bar{n} - 1$ and go to Step 2.

Stop.

This algorithm has at most $\lfloor \log_2(N) \rfloor + 1$ iterations and is $O(N)$ (see the proof in Appendix A.2). For non-decreasing discrete functions directions of inequalities in the above algorithm should be reversed.

Sometimes we encounter the problem of searching the condition (3.3) in discrete functions of type $f[n] = g\,(\Psi(n, \mathbf{T}), \mathbf{\Phi}(n, \mathbf{T}))$, where $\mathbf{T}$ is an arbitrary set with N members ($|\mathbf{T}| = N$), $g(\alpha, \mathbf{U})$ is a function of the scalar α and the set $\mathbf{U}$ with arbitrary number of members and we have

$$g(\alpha, \mathbf{U}) = \bar{g}(\alpha, \mathbf{h}(\mathbf{U})),$$

where $\mathbf{h}(\mathbf{U}) = [h_1(\mathbf{U}), h_2(\mathbf{U}), \ldots, h_q(\mathbf{U})]$ is a vector of q functions of the set $\mathbf{U}$ (in this paper we assume that $q \leq 2$) and $\bar{g}(\alpha, \mathbf{h})$ is a function of the scalar α and the vector $\mathbf{h}$ and we have

$$\mathbf{h}(\mathbf{U_1} \cup \mathbf{U_2}) = \mathbf{h}(\mathbf{U_1}) + \mathbf{h}(\mathbf{U_2}) - \mathbf{h}(\mathbf{U_1} \cap \mathbf{U_2}), \tag{3.4a}$$

$$\mathbf{h}(\{\}) = 0, \tag{3.4b}$$

where $\{\}$ denotes the empty set.

Since the complexity of finding $\Psi(\bar{n},\, \mathbf{T})$ and $\mathbf{\Phi}(\bar{n},\, \mathbf{T})$ is $O\,(N)$, the total complexity of Algorithm 3.4 for the function $f[n] = g\,(\Psi(n,\, \mathbf{T}), \mathbf{\Phi}(n, \mathbf{T}))$ will be $O\,(N \log_2(N))$. But by some modifications, the complexity can be reduced to $O\,(N)$.

First, since at each step of Algorithm 3.4 we have $\bar{n} \in [n_1,\, n_2]$, to reduce the complexity of finding $\mathbf{\Phi}(n,\, \mathbf{T})$, at each iteration we can limit the set $\mathbf{T}$ as $\tilde{\mathbf{T}} = \bar{\mathbf{\Phi}}(n_1, n_2, \mathbf{T})$ and use the relation

$$\mathbf{\Phi}(\bar{n},\, \mathbf{T}) = \mathbf{\Phi}(n_1,\, \mathbf{T}) \cup \mathbf{\Phi}(\bar{n} - n_1 + 1,\, \tilde{\mathbf{T}}).$$

Similarly we have

$$\Psi(\bar{n}, \mathbf{T}) = \Psi(\bar{n} - n_1 + 1, \tilde{\mathbf{T}}).$$

Define $\mathbf{T}_0 \triangleq \mathbf{\Phi}(n_1,\, \mathbf{T})$. Since $\tilde{\mathbf{T}} \cap \mathbf{T_0} = \{\}$ according to (3.4) we would have

$$f[\bar{n}] = g\left(\Psi(\bar{n}, \mathbf{T}), \mathbf{h}(\mathbf{T_0}) + \mathbf{h}(\mathbf{\Phi}(\bar{n} - n_1 + 1, \tilde{\mathbf{T}}))\right).$$

Second, since during the bisection algorithm n_1 and n_2 are closing together, the set $\tilde{\mathbf{T}}$ will be a subset of $\tilde{\mathbf{T}}$ at the previous iteration, and hence $\tilde{\mathbf{T}}$ can be found from the recursive relation:

$$\tilde{\mathbf{T}}^{(i+1)} = \begin{cases} \mathbf{\Phi}'({n_2}^{(i)} - \bar{n}^{(i)},\, \tilde{\mathbf{T}}^{(i)}), & \text{if } f[\bar{n}^{(i)}] > 0 \\ \mathbf{\Phi}(\bar{n}^{(i)} - {n_1}^{(i)},\, \tilde{\mathbf{T}}^{(i)}), & \text{otherwise,} \end{cases}$$

where the superscript $^{(i)}$ denotes the iteration number.

Third, when n_1 is changed to $\bar{n} + 1$, the value of $\mathbf{T}_0 = \mathbf{\Phi}(n_1,\, \mathbf{T})$ should be updated and the members of $\mathbf{T}$ with ranks n_1 to $\bar{n}$ (*i.e.* $\bar{\mathbf{\Phi}}(n_1,\, \bar{n},\, \mathbf{T}) = \mathbf{\Phi}(\bar{n} - n_1 + 1,\, \tilde{\mathbf{T}})$) should be added to $\mathbf{T}_0$. Thus, $\mathbf{T}_0$ can be recursively obtained as:

$${\mathbf{T}_0}^{(i+1)} = \begin{cases} {\mathbf{T}_0}^{(i)} \cup \mathbf{\Phi}(\bar{n}^{(i)} - {n_1}^{(i)} + 1,\, \tilde{\mathbf{T}}^{(i)}), & \text{if } f[\bar{n}^{(i)}] > 0 \\ {\mathbf{T}_0}^{(i)} \quad \text{(no change)}, & \text{otherwise.} \end{cases}$$

Similarly, the value of $\mathbf{h}_0 \triangleq \mathbf{h}(\mathbf{T}_0)$ can be recursively updated as

$${\mathbf{h}_0}^{(i+1)} = \begin{cases} {\mathbf{h}_0}^{(i)} + \mathbf{h}(\mathbf{\Phi}(\bar{n}^{(i)} - {n_1}^{(i)} + 1, \tilde{\mathbf{T}}^{(i)})), & \text{if } f[\bar{n}^{(i)}] > 0 \\ {\mathbf{h}_0}^{(i)} \quad \text{(no change)}, & \text{otherwise.} \end{cases}$$

Based on the above modifications, we come to the following algorithm.

Algorithm 3.5. Modified Discrete Bisection Algorithm for discrete functions of type $f[n] = g(\Psi(n, \mathbf{T}), \mathbf{\Phi}(n, \mathbf{T}))$

Step 1: Set $n_1 = 1$, $n_2 = N$, $\mathbf{T}_0 = \{\}$ (empty set), $\mathbf{h}_0 = 0$, and $\tilde{\mathbf{T}} = \mathbf{T}$.

Step 2: Set $\bar{n} = \lfloor (n_1 + n_2)/2 \rfloor$ and find $f[\bar{n}] = \bar{g}\left(\Psi(\bar{n} - n_1 + 1, \tilde{\mathbf{T}}), \mathbf{h}_0 + \mathbf{h}(\mathbf{\Phi}(\bar{n} - n_1 + 1, \tilde{\mathbf{T}}))\right)$ and $f[\bar{n} + 1] = \bar{g}\left(\Psi(\bar{n} - n_1 + 2, \tilde{\mathbf{T}}), \mathbf{h}_0 + \mathbf{h}(\mathbf{\Phi}(\bar{n} - n_1 + 2,\, \tilde{\mathbf{T}}))\right)$.

Step 3: If (3.3) is satisfied at $n^* = \bar{n}$, then stop the algorithm; $n^* = \bar{n}$ is the solution. Otherwise:
If $f[\bar{n}] > 0$ set $\mathbf{T}_0 = \mathbf{T}_0 \cup \mathbf{\Phi}(\bar{n} - n_1 + 1,\, \tilde{\mathbf{T}})$, $\mathbf{h}_0 = \mathbf{h}_0 + \mathbf{h}(\mathbf{\Phi}(\bar{n} - n_1 + 1,\, \tilde{\mathbf{T}}))$ and $n_1 = \bar{n} + 1$, then set $\tilde{\mathbf{T}} = \mathbf{\Phi}'(n_2 - \bar{n},\, \tilde{\mathbf{T}})$ and go to Step 2.
If $f[\bar{n}] < 0$ set $n_2 = \bar{n} - 1$ and then set $\tilde{\mathbf{T}} = \mathbf{\Phi}(\bar{n} - n_1,\, \tilde{\mathbf{T}})$ and go to Step 2.

Stop.

The total complexity of this algorithm is $O(N)$ (see the proof in Appendix A.3).

4. Solving the problem

In this section we solve the SBC problem (2.2) for two cases:

(1) The values of bits are constrained to be integer, *i.e.* $b_i \in \mathbb{Z}, \forall i$. In this case we call the problem as integer SBC (ISBC).
(2) The values of bits are constrained to be nonnegative and integer, *i.e.* $b_i \in \mathbb{Z}^+, \forall i$. In this case we call the problem as nonnegative integer SBC (NISBC).

In the unconstrained problem (3.2), the ith term of each summation only depends on b_i and $\{b_i^*(\lambda)\}$ in ISBC problem can be obtained from

$$\{b_i^*(\lambda)\} \ = \arg \min_{b_i \in \mathbb{Z}} \{J(b_i, \lambda)\},$$

where

$$J(b_i, \lambda) = \frac{\varepsilon_i \sigma_i^2}{\tau_i m_i} 4^{-b_i} + \lambda b_i \ , \quad i = 1, \ldots, p$$

Define the continuous function

$$J(\hat{b}_i \, , \lambda) \ = \frac{\varepsilon_i \sigma_i^2}{\tau_i m_i} 4^{-\hat{b}_i} + \lambda \hat{b}_i,$$

where $\hat{b}_i$ is a real variable. For convenience define

$$C_i \triangleq \frac{\varepsilon_i \sigma_i^2}{\tau_i m_i}.$$

To find $\{b_i^*(\lambda)\}$ based on Theorem 3.1, we should solve

$$J(\hat{b}_i \, , \lambda) - J(\hat{b}_i - 1 \, , \ \lambda) = 0$$

$$C_i 4^{-\hat{b}_i} + \lambda \hat{b}_i \ - (C_i 4^{-\hat{b}_i + 1} + \lambda(\hat{b}_i - 1)) = 0$$

$$3 C_i 4^{-\hat{b}_i} = -\lambda$$

$$\hat{b}_i(\lambda) = \log_4(3C_i) - \log_4(\lambda), \quad \lambda \geq 0 \tag{4.1}$$

According to Theorem 3.1, $\{b_i^*(\lambda)\}$ which is the minimum of the discrete function $J(b_i \, , \lambda)$ is obtained from

$$\{b_i^*(\lambda)\} = \begin{cases} \left\lfloor \hat{b}_i(\lambda) \right\rfloor & \text{if } \hat{b}_i(\lambda) \text{ is not integer} \\ \hat{b}_i(\lambda) \text{ or } \hat{b}_i(\lambda) - 1 & \text{if } \hat{b}_i(\lambda) \text{ is integer,} \end{cases} \tag{4.2}$$

where $\hat{b}_i(\lambda)$ is given in (4.1). It should be noted that since the $J(b_i \, , \lambda)$ has only one local minimum as (4.2), it is the global minimum of the $J(b_i, \lambda)$ in $\mathbb{Z}$ for $\lambda \geq 0$ (According to Thm. 3.3, we only consider the problem for $\lambda \geq 0$).

In the case of NISBC problem in which b_is should be nonnegative, since $J(b_i \, , \lambda)$ is a unimodal function of b_i, if the minimum of $J(b_i \, , \lambda)$ in $\mathbb{Z}$ is negative, since the unimodal function $J(b_i \, , \lambda)$ at the right hand side of its minimum is increasing, its minimum in $\mathbb{Z}^+$ will be zero. Thus, in NISBC we have

$$\{b_i^*(\lambda)\} = \begin{cases} \left\lfloor \hat{b}_i(\lambda) \right\rfloor \ u(\hat{b}_i(\lambda)) & \text{if } \hat{b}_i(\lambda) \text{ is not integer} \\ \hat{b}_i(\lambda) \, u(\hat{b}_i(\lambda)) \text{ or} & \\ (\hat{b}_i(\lambda) - 1) \, u(\hat{b}_i(\lambda) - 1) & \text{if } \hat{b}_i(\lambda) \text{ is integer,} \end{cases} \tag{4.3}$$

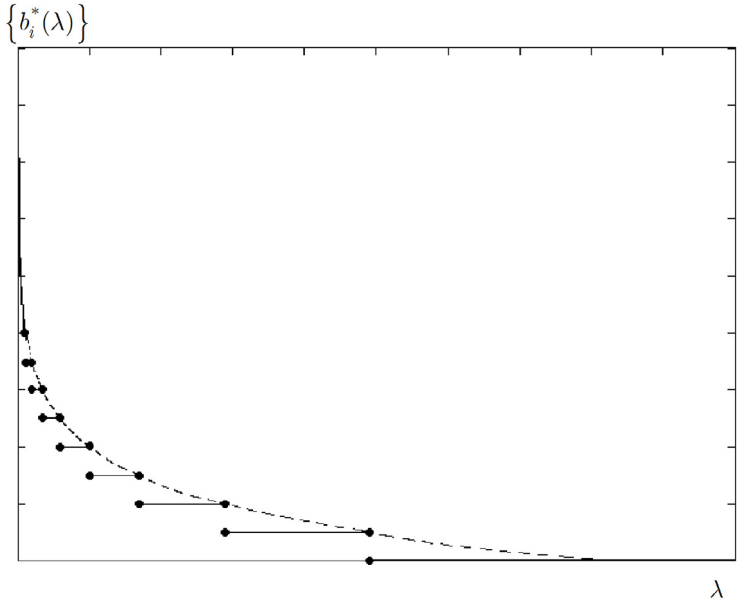

FIGURE 1. Typical diagram of $\{b_i^*(\lambda)\}$ (solid) and $\hat{b}_i(\lambda)$ (dashed) *versus* λ.

where, $u(.)$ denotes the unit step function:

$$u(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0. \end{cases}$$

Define

$$\{B^*(\lambda)\} \triangleq \sum_{i=1}^{p} k_j \{b_i^*(\lambda)\}. \tag{4.4}$$

Typical plots of $\{b_i^*(\lambda)\}$ and $\{B^*(\lambda)\}$ are shown in Figures 1 and 2, respectively. The optimum λ can be found by solving $\{B^*(\lambda)\} = B_1$ (B_1 is defined in Sect. 3.2).

Some Definitions:

Define λ^c as a critical λ of $\{b_i^*(\lambda)\}$ if $\{b_i^*(\lambda)\}$ has a jump and has two different values at λ^c. According to (4.2) and (4.3) at critical λs $\hat{b}_i(\lambda)$ is integer. For distinguishing the two values of $\{b_i^*(\lambda)\}$ at λ^c define

$$b_i^*(\lambda^c)_L = \text{lower value of } \{b_i^*(\lambda)\} \text{ at } \lambda^c, \tag{4.5}$$

$$b_i^*(\lambda^c)_H = \text{higher value of } \{b_i^*(\lambda)\} \text{ at } \lambda^c. \tag{4.6}$$

According to (4.4), at each critical λ of $\{b_i^*(\lambda)\}$s, $\{B^*(\lambda)\}$ also has a jump. Since some of $\{b_i^*(\lambda)\}$s may have common critical λs, $\{B^*(\lambda)\}$ may have more than two values at critical λs. Define

$$B^*(\lambda^c)_L = \sum_{i=1}^{k} b_i^*(\lambda^c)_L = \text{lowest value of } \{B^*(\lambda)\} \text{ at } \lambda^c,$$

$$B^*(\lambda^c)_H = \sum_{i=1}^{k} b_i^*(\lambda^c)_H = \text{highest value of } \{B^*(\lambda)\} \text{ at } \lambda^c.$$

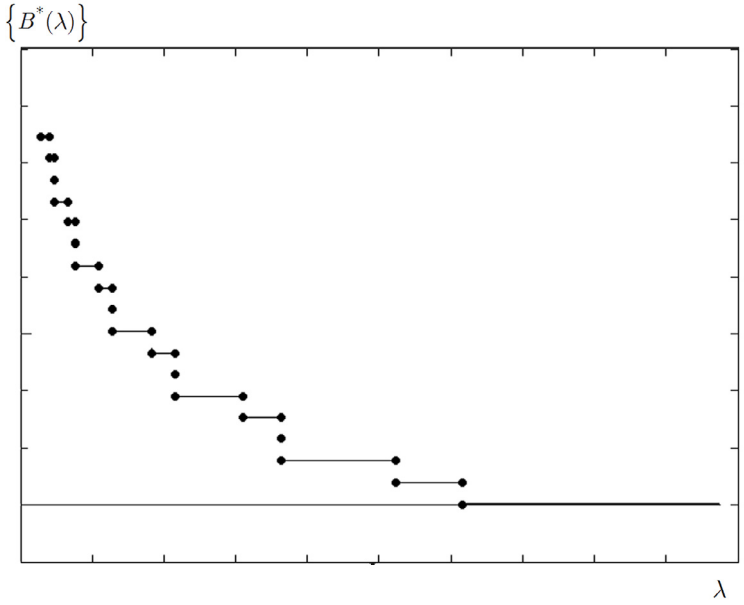

FIGURE 2. Typical diagram of $\{B^*(\lambda)\}$ *versus* λ.

Define $\lambda^c_{i,j}$ as a critical λ of $\{b^*_i(\lambda)\}$ at which $b^*_i(\lambda^c_{i,j})_L = j-1$ and $b^*_i(\lambda^c_{i,j})_H = j$ where i and j are integer numbers. According to (4.1)–(4.3) critical λs can be obtained from

$$\log_4(3C_i) - \log_4(\lambda) = j,$$

$$\lambda^c_{i,j} = 3C_i 4^{-j}. \tag{4.7}$$

Define the interval of λ at which the diagram $\{B^*(\lambda)\}$ is constant as a step. A step of $\{B^*(\lambda)\}$ may be a singular point or a nonsingular step as illustrated in Figures 3 and 4 (see also Lem. 4.1).

Define $\mathbf{S} = \{1, 2, \ldots, p\}$ as the set of all subbands.

Denote the number of members of any set like $\mathbf{T}$ by $|\mathbf{T}|$.

Denote the set of subbands with positive (nonzero) optimal number of bits by $\mathbf{S}^+$ and define $p^+ = |\mathbf{S}^+|$.

Lemma 4.1. *ISBC and NISBC problems have the following properties:*
If the problem has more than one optimal solution, all the solutions occur at a unique critical λ *as* λ_s *and the number of solutions is* $\binom{m}{l}$*, where* m *is the number of* $\{b^*_i(\lambda)\}$*s that have jump at* λ_s *and* $l = B - B^*(\lambda_s)_L$*. If the problem has a unique solution, the solution occurs at a nonsingular step of* $\{B^*(\lambda)\}$ *(see Fig. 3).*

The proof of Lemma 4.1 can be found in Appendix A.4.

Lemma 4.2.

(a) *The NISBC problem is an ISBC problem for the subbands in* $\mathbf{S}^+$*. In other words, the solution of NISBC problem can be found by solving the problem:*

$$\underset{b_i \in \mathbb{Z}}{\text{Minimize}} \left\{ \sum_{i \in \mathbf{S}^+} C_i 4^{-b_i} \right\} \quad \text{Subject to:} \sum_{i \in \mathbf{S}^+} \frac{b_i}{m_i} \leq R.$$

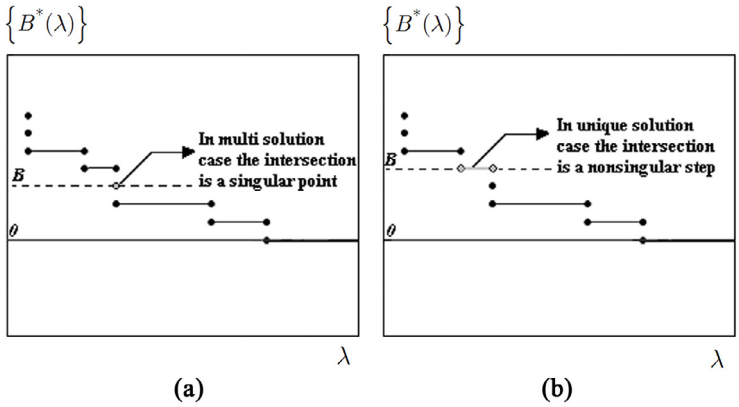

FIGURE 3. Intersection of $\{B^*(\lambda)\}$ with the horizontal line $f(\lambda) = B$. (a) In multi solution case the intersection is a singular point. (b) In unique solution case the intersection is a nonsingular step.

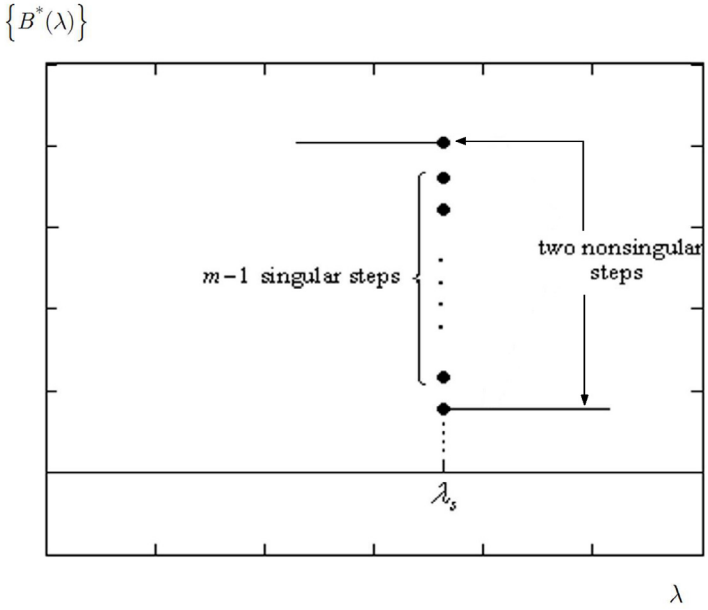

FIGURE 4. Diagram of $\{B^*(\lambda)\}$ about the critical point λ_s when m number of $\{b_i^*(\lambda)\}$s $(m \geq 2)$ have two values at λ_s.

(b) *Let* $\mathbf{S}^n$ *be an arbitrary subset of* $(\mathbf{S} - \mathbf{S}^+)$ *and* $\mathbf{S}^c = \mathbf{S}^n \cup \mathbf{S}^+$. *Then, in the ISBC problem for the subbands in* $\mathbf{S}^c$, i.e., *in the following problem:*

$$\underset{b_i \in \mathbb{Z}}{\textit{Minimize}} \left\{ \sum_{i \in \mathbf{S}^c} C_i 4^{-b_i} \right\} \quad \textit{Subject to:} \sum_{i \in \mathbf{S}^c} \frac{b_i}{m_i} \leq R$$

we have $b_i^* \leq 0, \forall i \in \mathbf{S}^n$, *where* b_i^* *is the optimal value of* b_i *in the above problem.*

The proof of Lemma 4.2 can be found in Appendix A.5.

Remark 4.3. According to (4.2)–(4.4), $\{b_i^*(\lambda)\}$s and $\{B^*(\lambda)\}$ are non-increasing as λ increases.

Here, we first solve the ISBC problem and then modify the solution for NISBC. Let λ^* be an optimum λ and b_i^* be an optimum value of b_i related to λ^* in the ISBC problem. Then, we have $\lambda^c_{i,b_i^*+1} \le \lambda^* \le \lambda^c_{i,b_i^*}$ because outside this interval we have $\{b_i^*(\lambda)\} \neq b_i^*$. First, we assume that the solution of ISBC problem occurs at a nonsingular step and all the b_is have a unique optimal value. In this case, since $\{B^*(\lambda^*)\} = B_1 \le B$, $\lambda^c_{i,b_i^*+1} \le \lambda^* \le \lambda^c_{i,b_i^*}$ and $\{B^*(\lambda)\}$ is non-increasing with λ, we have $B^*(\lambda^c_{i,b_i^*})_H \le B$. Also $\{B^*(\lambda)\}$ has a jump at λ^c_{i,b_i^*+1} and $B^*(\lambda^c_{i,b_i^*+1})_H > B$. Consequently, b_i^* is the largest integer number as j for which $B^*(\lambda^c_{i,j})_H \le B$. *i.e.*

$$b_i^* = \max\left\{ j \,:\, B(\lambda^c_{i,j})_H = \sum_{n=1}^{p} k_n b_n^*(\lambda^c_{i,j})_H \le B \text{ and } j \in \mathbb{Z} \right\}. \tag{4.8}$$

Substituting (4.1), (4.2) and (4.7) in (4.8) yields

$$b_i^* = \max\left\{ j \,:\, \sum_{n=1}^{p} k_n \left\lfloor \log_4(C_n) - \log_4(C_i) + j \right\rfloor \le B \text{ and } j \in \mathbb{Z} \right\}. \tag{4.9}$$

Defining $k \triangleq \sum_{n=1}^{p} k_n$ we obtain

$$\begin{aligned} b_i^* &= \max\left\{ j \,:\, kj + \sum_{n=1}^{p} k_n \left\lfloor \log_4(C_n) - \log_4(C_i) \right\rfloor \le B \text{ and } j \in \mathbb{Z} \right\} \\ &= \max\left\{ j \,:\, j \le \frac{B}{k} - \frac{1}{k} \sum_{n=1}^{p} k_n \left\lfloor \log_4(C_n) - \log_4(C_i) \right\rfloor \text{ and } j \in \mathbb{Z} \right\}. \end{aligned} \tag{4.10}$$

According to definition of floor we have

$$b_i^* = \left\lfloor \frac{B}{k} - \frac{1}{k} \sum_{n=1}^{p} k_n \left\lfloor \log_4(C_n) - \log_4(C_i) \right\rfloor \right\rfloor. \tag{4.11}$$

In the case that λ^* is critical, the problem has more than one optimal solution. Moreover, (4.11) can be expressed in another form with lower computational complexity, as declared in Theorem 4.4.

Theorem 4.4. *The optimal solution of ISBC problem can be found from*

$$b_i^* = \begin{cases} \tilde{b} + \lfloor \log_4(C_i) \rfloor + 1 & \text{if } \ \rho_i \ge \tilde{\rho} \\ \tilde{b} + \lfloor \log_4(C_i) \rfloor & \text{otherwise} \end{cases} \tag{4.12}$$

where

$$\tilde{b} = \left\lfloor \frac{B - B_0}{k} \right\rfloor, \quad \rho_i = \log_4(C_i) - \lfloor \log_4(C_i) \rfloor,$$

$$B_0 = \sum_{i=1}^{p} k_i \lfloor \log_4(C_i) \rfloor, \quad \tilde{B} = B - B_0 - k\tilde{b},$$

$$\tilde{k}_\rho = -\sum_{i=1}^{p} k_i \lfloor \rho_i - \rho \rfloor = \sum_{i \in \{j:\rho_j \le \rho\}} k_i \ , \ \forall \rho \in [0, 1),$$

$$\tilde{\rho} = \min\left\{ \rho_n : \tilde{k}_{\rho_n} \ge k - \tilde{B} \right\}.$$

The value of $\tilde{\rho}$ *can be obtained using Algorithm 3.5 with* $f[n] = g(\mathbf{\Phi}(n, \mathbf{\Gamma})) = \left(\sum_{i\in\{j:\rho_j\in\mathbf{\Phi}(n,\mathbf{\Gamma})\}} k_i\right) - \left(k - \tilde{B}\right)$ *and setting* $\tilde{\rho} = \Psi(n^*, \mathbf{\Gamma})$*, where* $\mathbf{\Gamma} \triangleq \{\rho_i\}_{i=1}^{p}$*. In the case that some of* b_i*s have two optimal values, the corresponding* ρ_i*s are equal. In this case, the problem has* $\binom{m}{l}$ *solutions where* m *is the number of* b_i*s that have two optimal values (have equal* ρ_i*s) and* $l = B - \left(\sum_{i=1}^{p} k_i \left(\tilde{b} + \lfloor \log_4(C_i) \rfloor\right)\right)$.

The proof of Theorem 4.4 can be found in Appendix A.6.

Now, we modify the solution for NISBC problem. Define

$$\mathbf{C} = \{C_i\}_{i=1}^{p}\ , \quad \bar{\mathbf{S}}(n) = \{i \in \mathbf{S} : C_i \in \mathbf{\Phi}(n, \mathbf{C})\},$$

$$\bar{B}_0(n) = \sum_{i\in\bar{\mathbf{S}}(n)} k_i \lfloor \log_4(C_i) \rfloor, \quad \bar{k}(n) = \sum_{i\in\bar{\mathbf{S}}(n)} k_i$$

and

$$\bar{\tilde{b}}(n) = \left\lfloor \frac{B - \bar{B}_0(n)}{\bar{k}(n)} \right\rfloor.$$

Define $\bar{b}_i^*(n)$ as the upper value of b_i^* in (4.12) with $\tilde{b} = \bar{\tilde{b}}(n)$, *i.e.*:

$$\bar{b}_i^*(n) = \bar{\tilde{b}}(n) + \lfloor \log_4(C_i) \rfloor + 1.$$

According to part (a) of Lemma 4.2, to solve the NISBC problem, one can find the subbands with positive number of bits ($\mathbf{S}^+$) and then find the ISBC for those subbands. According to (4.12), b_i^* is non-decreasing with C_i and the subbands in $\mathbf{S}^+$ have larger values of C_i compared to those in $(\mathbf{S} - \mathbf{S}^+)$. Thus, $\mathbf{S}^+$ can be found as

$$\mathbf{S}^+ = \left\{i : C_i \in \mathbf{\Phi}\left(p^+,\ \mathbf{C}\right)\right\}. \tag{4.13}$$

But p^+ should be found. Define i_n as the index of the subband for which

$$C_{i_n} = \Psi(n, \mathbf{C}).$$

At first, we assume that the upper value of b_i^* in (4.12) is the optimal value of b_i (we call this assumption as **assumption (*)**) and then we check the lower value.

According to (4.13), we have $\bar{\mathbf{S}}(p^+) = \mathbf{S}^+$ and $\bar{\mathbf{S}}(p^+ + 1) = \mathbf{S}^+ \cup \left\{i_{p^+ +1}\right\}$ where it is clear from (4.13) that $i_{p^+ +1} \in (\mathbf{S} - \mathbf{S}^+)$. Thus, in the case of assumption (*), according to part (b) of Lemma 4.2 we have $\bar{b}^*_{i_{p^+}}(p^+) > 0$ and $\bar{b}^*_{i_{p^+ +1}}(p^+ + 1) \le 0$. Therefore, p^+ can be found using Algorithm 3.5 with $f[n] = \bar{b}^*_{i_n}(n)$.

It should be noted that $\bar{b}^*_{i_n}(n)$ is a function of $C_{i_n} = \Psi(n, \mathbf{C})$ and $\bar{\mathbf{S}}(n) = \{i \in \mathbf{S} : C_i \in \mathbf{\Phi}(n, \mathbf{C})\}$ and hence $\bar{b}^*_{i_n}(n)$ is in the form of $g(\Psi(n, \mathbf{C}), \mathbf{\Phi}\ (n,\ \mathbf{C}))$ where

$$g(\alpha, \mathbf{U}) = \bar{g}(\alpha, \mathbf{h}(\mathbf{U})) = \left\lfloor \frac{B - h_1(\mathbf{U})}{h_2(\mathbf{U})} \right\rfloor + \lfloor \log_4(\alpha) \rfloor + 1,$$

$$\mathbf{h}(\mathbf{U}) = [h_1(\mathbf{U}) = \bar{B}_0(\mathbf{U}), h_2(\mathbf{U}) = \bar{k}(\mathbf{U})],$$

$$\bar{B}_0(\mathbf{U}) = \sum\nolimits_{i\in\bar{\mathbf{S}}(\mathbf{U})} k_i \lfloor \log_4(C_i) \rfloor, \quad \bar{k}(\mathbf{U}) = \sum\nolimits_{i\in\bar{\mathbf{S}}(\mathbf{U})} k_i$$

and

$$\bar{\mathbf{S}}(\mathbf{U}) = \{i \in \mathbf{S} : C_i \in \mathbf{U}\}.$$

According to the above relations, we have $\mathbf{h}(\mathbf{U_1} \cup \mathbf{U_2}) = \mathbf{h}(\mathbf{U_1}) + \mathbf{h}(\mathbf{U_2}) - \mathbf{h}(\mathbf{U_1} \cap \mathbf{U_2})$ and the condition (3.4) is satisfied.

At the next step, to obtain the optimal solution in general case without assumption (*), we should check the lower value in (4.12) (See Step 2 of Algorithm 4.5 below).

Algorithm 4.5. Finding p^+ and $\mathbf{S}^+$

Step 1: Find p^+ in the case of assumption (*) using Algorithm 3.5 with $f[n] = \bar{b}^*_{i_n}(n)$. Then, set $p^+ = n^*$ and $\mathbf{S}^+ = \{i : C_i \in \mathbf{\Phi}\,(p^+,\ \mathbf{C}\,)\}$.

Step 2: For each $n \in \mathbf{S}^+$ check if $\rho_n < \tilde{\rho}$ and $b^*_n(p^+) - 1 \leq 0$ remove the subband n from $\mathbf{S}^+$. Then, set $p^+ = |\mathbf{S}^+|$.

Stop.

Obviously, similar to Algorithm 3.5, the complexity of Algorithm 4.5 is $O(p)$.

According to Lemma 4.2 and (4.12), the solution of the NISBC problem is obtained as

$$b^*_i = \begin{cases} 0 & \text{if } \ i \notin \mathbf{S}^+ \\ \tilde{b}^+ + \lfloor \log_4(C_i) \rfloor + 1 & \text{if } \ i \in \mathbf{S}^+ \text{ and } \rho_i \geq \tilde{\rho}^+ \\ \tilde{b}^+ + \lfloor \log_4(C_i) \rfloor & \text{otherwise.} \end{cases} \tag{4.14}$$

where

$$B_0^+ = \sum_{n \in \mathbf{S}^+} k_n \lfloor \log_4(C_n) \rfloor, \quad k^+ = \sum_{n \in \mathbf{S}^+} k_n,$$

$$\tilde{b}^+ = \left\lfloor \frac{B - B_0^+}{k^+} \right\rfloor, \quad \tilde{B}^+ = B - B_0^+ - k^+ \tilde{b}^+,$$

$$\tilde{k}^+_\rho = -\sum_{i \in \mathbf{S}^+}^{p} k_i \lfloor \rho_i - \rho \rfloor = \sum_{i \in \{j \in \mathbf{S}^+ : \rho_j \leq \rho\}} k_i \ , \ \forall \rho \in [0,\, 1),$$

$$\tilde{\rho}^+ = \min \left\{ \rho_n \in \mathbf{S}^+ : \tilde{k}^+_{\rho_n} \geq k - \tilde{B} \right\}.$$

Since the complexity of Algorithm 4.5 for finding p^+ and $\mathbf{S}^+$ is $O(p)$, the total complexity of computing the analytical solution of NISBC from (4.14) is also $O(p)$. The complexity is considerably lower than Fox algorithm which is $O\left((p + B)\log_2 p\right)$ (it should be noted that the complexity of Fox algorithm is $O(Bp)$ but, as explained in footnote of [9], by using binary insertion, the complexity can be reduced to $O\left((p + B)\log_2 p\right)$). Furthermore, unlike the Fox algorithm, the complexity of the proposed method does not depend on the total number of bits, B.

Compared to the method of rounding down the real valued solution (2.3), the proposed method leads to lower quantization MSE and is superior (see the next section). However, the order of computational complexity of proposed method is not more than the method of rounding down the real valued solution and the both methods are $O(p)$.

5. Simulation results

A computer simulation is performed for evaluating the performance of SBC algorithm. A sinusoidal signal with amplitude 1 in a zero mean white Gaussian noise with variance 1 is generated 10 000 times. The generated signal is quantized with the average rate of 3 bits/sample using the following methods:

(1) Proposed SBC method with Haar discrete wavelet transform (DWT).
(2) Method of rounding down the real valued solution (2.3) (common SBC method) with Haar DWT.
(3) Quantizing all samples with equal number of bits (3 bits) in time domain (Pulse Code Modulation (PCM) method).

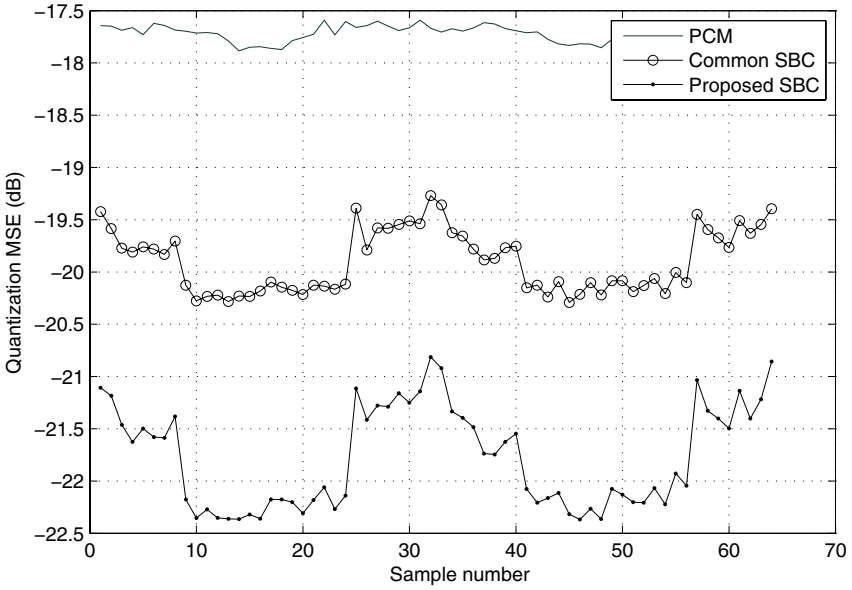

FIGURE 5. Quantization MSE for PCM, proposed SBC and common SBC methods for all time samples with the average bitrate of 3 bits per sample.

TABLE 1. Average quantization MSE for PCM, proposed SBC and common SBC methods using Haar DWT.

Method	Average Quantization MSE (dB)
PCM	−17.71
Common SBC	−19.89
Proposed SBC	−21.74

The other SBC parameters for the methods 1 and 2 are: $p = 6$, signal length= 64 samples, $k_1 = k_2 = 2$, $k_3 = 4$, $k_4 = 8$, $k_5 = 16$, $k_6 = 32$ and $B = 192$ (3 bits/sample).

The quantization MSE is estimated by averaging over squared errors in 10 000 runs. The results for all samples in time domain is shown in Figure 5. The superiority of the proposed SBC method is obvious in Figure 5. The average MSEs over all the samples for the above three methods are shown in Table 1. According to Table 1, the average MSE for the proposed SBC method is about 1.85 dB lower than the common SBC method and 4.03 dB lower than PCM method.

Another computer simulation is carried out to compare the running time needed for computing the analytical solution (4.14) with the running time of Fox algorithm (with binary insertion). Each program is runned 100 times for each p and the average running time over all runs is obtained (the specifications of the computer used for running all programs are: CPU: Intel B970, dual core, 2.3 GHz, RAM: 4 GB, OS: Windows 7, 64-bit). The results for $p = 2$ to 64 with random values of C_is and $B = 3p$ and $B = 6p$ are shown in Figures 6 and 7, respectively. The results confirm that the proposed method is faster than Fox algorithm.

Furthermore, comparing Figure 6 with Figure 7, since in Figure 7 the value of B is two times larger, it can be concluded that the complexity of Fox algorithm grow with total number of bits, B, but the complexity of proposed method does not grow as the total number of bits increases.

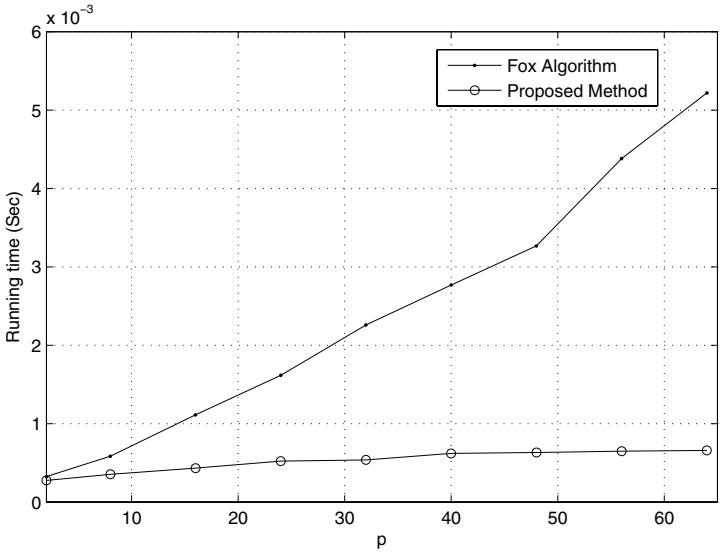

FIGURE 6. Average running time of proposed method and Fox algorithm for random values of C_is, $p = 2$ to 64 and $B = 3p$.

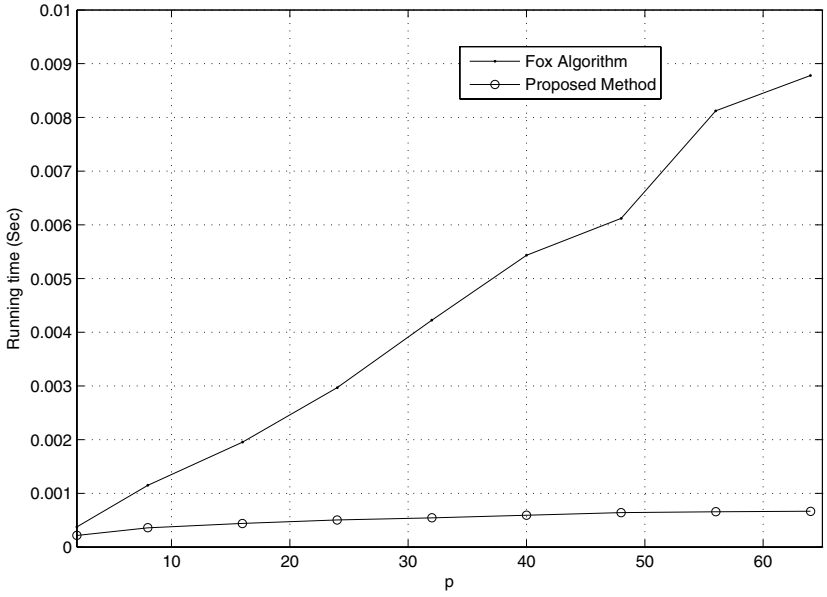

FIGURE 7. Average running time of proposed method and Fox algorithm for random values of C_is, $p = 2$ to 64 and $B = 6p$.

6. Conclusion

In this paper using an analytical discrete optimization approach, an analytical solution obtained for SBC with nonnegative integer bit allocation. The solution is valid for both uniform and multi-resolution filter banks (including wavelet). The complexity of computing the proposed analytical solution of NISBC is $O(p)$. The complexity is considerably lower than Fox algorithm which is $O((p + B)\log_2 p)$. Compared to the method of

rounding down the real valued solution, the proposed method leads to lower quantization MSE and is superior (see the next section). However, the order of computational complexity of proposed method is not more than the method of rounding down the real valued solution (2.3) and the both methods are $O(p)$.

Using computer simulation the performance of proposed SBC method was compared to the common SBC method (rounding the real valued solution) and PCM method. The results showed that the proposed method is superior. Moreover, the order of computational complexity of proposed SBC method is not increased compared to the common SBC method.

Appendix A. Proofs of theorems, lemmas and the computational complexity of algorithms

A.1. Proof of Theorem 3.1

Assume that n^* is a local minimum of $f[n]$. We have

$$f[n^*] \leq f[n^* - 1] \Rightarrow f_r(n^*) \leq f_r(n^* - 1)$$

$$\Rightarrow f_r(n^*) - f_r(n^* - 1) \leq 0. \tag{A.1}$$

Also we have

$$f[n^*] \leq f[n^* + 1] \Rightarrow f_r(n^*) \leq f_r(n^* + 1)$$

$$\Rightarrow f_r(n^* + 1) - f_r(n^*) \geq 0. \tag{A.2}$$

Since $f_r(x)$ is a continuous function on the interval $I_r = [a, b]$, $f_r(x - 1)$ is also a continuous function on the interval $[a + 1, b + 1]$ and therefore $g(x) = f_r(x) - f_r(x - 1)$ is a continuous function on $[a + 1, b]$. Assume that $I = \{n_1, n_1 + 1, \ldots, n_2\} \subset I_r$. According to definition of local minima (Sect. 3.1) we have $n^* \in \{n_1 + 1, n_1 + 1, \ldots, n_2 - 1\} \subset [a + 1,\ b - 1]$ and consequently $[n^*, n^* + 1] \subset [a + 1,\ b]$. According to (A.1) and (A.2) we have $g(n^*) \leq 0$ and $g(n^* + 1) \geq 0$ and since $g(x)$ is a continuous function, according to Bolzano's theorem it has at least one root in the interval $[n^*, n^* + 1]$. Thus, the equation $f_r(x) - f_r(x - 1) = 0$ has at least one solution in the interval $[n^*, n^* + 1]$. If the solution(s) are in the interval $[n^*, n^* + 1)$, the floor of the solution(s) would be n^*. If the solution is equal to $n^* + 1$, since $n^* + 1$ is integer, $n^* + 1$ and $(n^* + 1) - 1 = n^*$ would be in the set **A**. Thus, in either case the set **A** contains the local minimum n^*. Similarly, it can be shown that if n^* is a local maximum of $f[n]$, the set **A** contains n^* [14].

A.2. Computational complexity of Algorithm 3.4

In Step 2 of Algorithm 3.4 if $f[\bar{n}] > 0$ the value n_1 is set to $\bar{n} + 1$ and if $f[\bar{n}] < 0$ the value n_2 is set to $\bar{n} - 1$. Thus, in ith iteration we would have

$$N_i \leq \frac{N_{i-1}}{2} \leq \frac{N}{2^i},$$

where N_i denotes the value of $n_2 - n_1$ at ith iteration. Let I be the total number of iterations then

$$\frac{N}{2^I} \leq 1$$

$$I = \lfloor \log_2 N \rfloor + 1,$$

Thus, the maximum number of iterations is $\lfloor \log_2 N \rfloor + 1$ and the algorithm is $O(\log_2 N)$.

A.3. Computational complexity of Algorithm 3.5

Similar to Algorithm 3.4, this algorithm has at most $\lfloor \log_2(N) \rfloor + 1$ iterations. At each iteration the sets $\tilde{\mathbf{T}}$ is bisected. Thus, the complexity of computing $f[\bar{n}]$, $f[\bar{n}+1]$, $\tilde{\mathbf{T}}$, $\mathbf{T}_0$, $\mathbf{h}$ and $\mathbf{h}_0$ at the lth iteration using selection algorithm is $O(N/2^l)$ (it should be noted that q is a fixed number and in this paper we have $q \leq 2$). The order of total complexity of algorithm will be

$$\sum_{l=0}^{\lfloor \log_2(N) \rfloor + 1} \frac{N}{2^l} = N \frac{1 - (1/2)^{\lfloor \log_2 N \rfloor + 2}}{1 - (1/2)} \leq 2N.$$

A.4. Proof of Lemma 4.1

To find the solution(s) of the NISBC problem we should find the intersection of the diagram $\{B^*(\lambda)\}$ with the horizontal line $f(\lambda) = B$. Since $\{B^*(\lambda)\}$ is non-increasing with λ, we may encounter the following two cases. In the first case, the intersection of the diagram $\{B^*(\lambda)\}$ with the line $f(\lambda) = B$ is a singular point. In this case the solutions occur at a critical λ as λ_s. In the case that $\{B^*(\lambda)\}$ has a singular point at λ_s, two or more $\{b_i^*(\lambda)\}$ are discontinuous and have two values at λ_s. Assume that m number of $\{b_i^*(\lambda)\}$s have two values at $\lambda_s (m \geq 2)$ and let $l = B - B^*(\lambda_s)_L$. Then, in order to have $\{B^*(\lambda)\} = B$, for l number of $\{b_i^*(\lambda)\}$s which have two values at λ_s we must select the higher values and for the remaining $\{b_i^*(\lambda)\}$s select the lower values. Thus, in this case we have $\binom{m}{l}$ solutions for NISBC problem. Also $\{B^*(\lambda)\}$ would have $m-1$ singular points (corresponding to $1 \leq l \leq m-1$) and two nonsingular steps (corresponding to $l = 0$ and $l = m$) at λ_s (see Fig. 4).

In the second case, the intersection of the diagram $\{B^*(\lambda)\}$ with the line $f(\lambda) = B$ is a nonsingular step that contains infinite values of λ. Since we have no discontinuity at a nonsingular step, the values of $\{b_i^*(\lambda)\}$s are constant at a nonsingular step. Thus, in this case the NISBC problem has a unique solution.

A.5. Proof of Lemma 4.2

(a) Here for distinguishing ISBC and NISBC problems denote the optimum number of bits of ith subband in NISBC by b_i^{*+}. In the NISBC problem a λ should be found that

$$\sum_{i=1}^{p} k_i \left\{ b_i^{*+}(\lambda) \right\} = B_1,$$

$$\sum_{i:\, \left\{ b_i^{*+}(\lambda) \right\} > 0} k_i \left\{ b_i^{*+}(\lambda) \right\} + \underbrace{\sum_{i:\, \left\{ b_i^{*+}(\lambda) \right\} = 0} k_i \left\{ b_i^{*+}(\lambda) \right\}}_{=0} = B_1.$$

According to (4.2) and (4.3) we have $\left\{ b_i^{*+}(\lambda) \right\} = \{ b_i^*(\lambda) \}\, u(\{ b_i^*(\lambda) \})$. Then, we obtain

$$\sum_{i:\, b_i^{*+}(\lambda) > 0} k_i \{ b_i^*(\lambda) \}\, u(\{ b_i^*(\lambda) \}) = B_1. \tag{A.3}$$

In order to have

$$\left\{ b_i^{*+}(\lambda) \right\} = \{ b_i^*(\lambda) \}\, u(\{ b_i^*(\lambda) \}) > 0$$

we should have $u(\{ b_i^*(\lambda) \}) = 1$. Then (A.3) yields

$$\sum_{i:\, \left\{ b_i^{*+}(\lambda) \right\} > 0} k_i \{ b_i^*(\lambda) \} = B_1.$$

A λ should be found that satisfies the above relation. Thus the NISBC problem is an ISBC problem for the subbands that have positive optimal number of bits.

(b) Let λ_1^* be an optimum λ of the ISBC problem for the subbands $\mathbf{S}$ and λ_2^* be an optimum λ of the ISBC problem for the subbands $\mathbf{S}_c = \mathbf{S}_n \cup \mathbf{S}^+$. For the first problem we have

$$\sum_{i\in\mathbf{S}} k_i \{b_i^*(\lambda_1^*)\} = B_1,$$

$$\sum_{i:\,\{b_i^*(\lambda_1^*\})>0} k_i \{b_i^*(\lambda_1^*)\} + \sum_{i:\,\{b_i^*(\lambda_1^*)\}\le 0} k_i \{b_i^*(\lambda_1^*)\} = B_1.$$

According to part (a) we have

$$\mathbf{S}_n \subset \{i:\, b_i^*(\lambda_1^*) \le 0\}$$

and therefore

$$\sum_{i:\,\{b_i^*(\lambda)\}>0} k_i \{b_i^*(\lambda_1^*)\} + \sum_{i\in\mathbf{S}_n} k_i \{b_i^*(\lambda_1^*)\} \ge B_1.$$

For the second ISBC problem we have

$$\sum_{i:\,\{b_i^*(\lambda)\}>0} k_i \{b_i^*(\lambda_2^*)\} + \sum_{i\in\mathbf{S}_n} k_i \{b_i^*(\lambda_2^*)\} = B_1.$$

Since $\{b_i^*(\lambda)\}$s are non-increasing with λ, comparing the two last relations we deduce

$$\lambda_2^* \ge \lambda_1^*,$$

$$\forall i \in \mathbf{S}_n : \{b_i^*(\lambda_2^*)\} \le \{b_i^*(\lambda_1^*)\},$$
$$\forall i \in \mathbf{S}_n : \{b_i^*(\lambda_2^*)\} \le 0.$$

A.6. Proof of Theorem 4.4

Rewrite b_i^* in (4.11) as

$$b_i^* = \left\lfloor \frac{B}{k} - \frac{1}{k}\sum_{n=1}^{p} k_n \lfloor \log_4(C_n) - \log_4(C_i) \rfloor \right\rfloor$$

$$b_i^* = \left\lfloor \frac{B}{k} - \frac{1}{k}\sum_{n=1}^{p} k_n \lfloor \log_4(C_n) \rfloor + \frac{1}{k}\lfloor \log_4(C_i) \rfloor \sum_{n=1}^{p} k_n - \frac{1}{k}\sum_{n=1}^{p} k_n \lfloor \rho_n - \rho_i \rfloor \right\rfloor$$

$$b_i^* = \left\lfloor \frac{B}{k} - \frac{B_0}{k} + \lfloor \log_4(C_i) \rfloor - \frac{1}{k}\sum_{n=1}^{p} k_n \lfloor \rho_n - \rho_i \rfloor \right\rfloor, \tag{A.4}$$

where

$$B_0 = \sum_{n=1}^{p} k_n \lfloor \log_4(C_n) \rfloor,$$

$$\rho_n = \log_4(C_n) - \lfloor \log_4(C_n) \rfloor \ , \quad n = 1, 2, \ldots, k.$$

Since $\rho_n \in [0\,,\,1)$ then we have $(\rho_n - \rho_i) \in (-1\,,\,1)$ and hence $\lfloor \rho_n - \rho_i \rfloor \in \{-1\,,\,0\}$. Consequently we have

$$0 \le -\sum_{n=1}^{p} k_n \lfloor \rho_n - \rho_i \rfloor < \sum_{n=1}^{p} k_n = k,$$

$$0 \le -\frac{1}{k}\sum_{n=1}^{p} k_n \lfloor \rho_n - \rho_i \rfloor < 1.$$

Define

$$\tilde{k}_\rho = -\sum_{i=1}^{p} k_i \lfloor \rho_i - \rho \rfloor = \sum_{i \in \{j:\rho_j \le \rho\}} k_i \ , \ \forall \rho \in [0,\, 1), \tag{A.5}$$

$$\tilde{b} = \left\lfloor \frac{B - B_0}{k} \right\rfloor, \tag{A.6}$$

$$\tilde{B} = B - B_0 - k\tilde{b}. \tag{A.7}$$

Substituting (A.5) and (A.6) in (A.4) we obtain

$$\begin{aligned} b_i^* &= \left\lfloor \tilde{b} + \left(\tilde{B} + \tilde{k}_{\rho_i} \right) / k + \lfloor \log_4(C_i) \rfloor \right\rfloor \\ &= \tilde{b} + \left\lfloor \left(\tilde{B} + \tilde{k}_{\rho_i} \right) / k \right\rfloor + \lfloor \log_4(C_i) \rfloor . \end{aligned} \tag{A.8}$$

According to (A.7) $\tilde{B}$ is the reminder of dividing $B - B_0$ to k and we have $0 \le \tilde{B} < k$. Moreover, we have $0 \le \tilde{k}_{\rho_i} < k$ and consequently

$$0 \le \frac{\tilde{B} + \tilde{k}_{\rho_i}}{k} < 2. \tag{A.9}$$

According to (A.8) and (A.9) we conclude

$$b_i^* = \begin{cases} \tilde{b} + \lfloor \log_4(C_i) \rfloor + 1 & \text{if } \tilde{k}_{\rho_i} \ge k - \tilde{B}, \\ \tilde{b} + \lfloor \log_4(C_i) \rfloor & \text{otherwise.} \end{cases} \tag{A.10}$$

Define

$$\tilde{\rho} = \min \left\{ \rho_n : \tilde{k}_{\rho_n} \ge k - \tilde{B} \right\}. \tag{A.11}$$

Then we have

$$\tilde{k}_{\rho_i} \ge k - \tilde{B} \Leftrightarrow \tilde{k}_{\rho_i} \ge \tilde{k}_{\tilde{\rho}}.$$

According to (A.5) $\tilde{k}_{\rho_i}$ is non-decreasing with ρ_i, therefore

$$\tilde{k}_{\rho_i} \ge \tilde{k}_{\tilde{\rho}} \Leftrightarrow \rho_i \ge \tilde{\rho}.$$

Consequently (A.10) can be written as

$$b_i^* = \begin{cases} \tilde{b} + \lfloor \log_4(C_i) \rfloor + 1 & \text{if } \rho_i \ge \tilde{\rho}, \\ \tilde{b} + \lfloor \log_4(C_i) \rfloor & \text{otherwise.} \end{cases} \tag{A.12}$$

According to (A.11) and (A.5), $\tilde{\rho}$ can be obtained by applying Algorithm 3.5 to the discrete function $f[n] = \tilde{k}_{\Psi(n,\,\mathbf{\Gamma})} - \left(k - \tilde{B}\right) = \left(\sum_{i \in \{j:\rho_j \in \mathbf{\Phi}(n,\,\mathbf{\Gamma})\}} k_i\right) - \left(k - \tilde{B}\right)$ and setting $\tilde{\rho} = \Psi(n^*, \mathbf{\Gamma})$, where $\mathbf{\Gamma} \triangleq \{\rho_i\}_{i=1}^{p}$. It should be noted that $f[n]$ is in the form of $g(\mathbf{\Phi}(n, \mathbf{\Gamma}))$, where

$$\begin{aligned} g(\mathbf{U}) &= \bar{g}(h(\mathbf{U})) = h(\mathbf{U}) - (k - \tilde{B}), \\ h(\mathbf{U}) &= \sum\nolimits_{i \in \{j:\rho_j \in \mathbf{U}\}} k_i. \end{aligned}$$

According to the above relations, we have $h(\mathbf{U_1} \cup \mathbf{U_2}) = h(\mathbf{U_1}) + h(\mathbf{U_2}) - h(\mathbf{U_1} \cap \mathbf{U_2})$ and the condition (3.4) is satisfied.

When some of b_is have two optimal values, according to Lemma 4.1 some of $b_i^*(\lambda)$s have common critical λs which are equal to the optimal λ. Assume that b_i and b_k have two optimal values then

$$\lambda^c_{i,b_i^*} = \lambda^c_{k,b_k^*},$$

$$C_i 4^{-b_i^*} = C_k 4^{-b_k^*}.$$

By taking $\log_4$ we obtain

$$\log_4(C_i) - b_i^* = \log_4(C_k) - b_k^*.$$

Substituting b_i^* and b_k^* from (A.12) yields

$$\begin{aligned}
&\log_4(C_i) - \lfloor \log_4(C_i) \rfloor = \log_4(C_k) - \lfloor \log_4(C_k) \rfloor \text{ or} \\
&\log_4(C_i) - \lfloor \log_4(C_i) \rfloor = \log_4(C_k) - \lfloor \log_4(C_k) \rfloor + 1 \text{ or} \\
&\log_4(C_i) - \lfloor \log_4(C_i) \rfloor = \log_4(C_k) - \lfloor \log_4(C_k) \rfloor - 1.
\end{aligned}$$

$$\Rightarrow \rho_i = \rho_k \text{ or } \rho_i = \rho_k + 1 \text{ or } \rho_i = \rho_k - 1.$$

Since $\rho_n \in [0\,,\,1)$, $\forall n$ only the first equation is true and we have $\rho_i = \rho_k$. According to Lemma 4.1 we have $\binom{m}{l}$ solutions for the problem where m is the number of b_is that have two optimal values (have equal ρ_is) and

$$l = B - B_L^*(\lambda^*)$$

$$= B - \sum_{i=1}^{p} k_i \left(\tilde{b} + \lfloor \log_4(C_i) \rfloor \right).$$

Since $\tilde{\rho}$ and $\tilde{b}$ are independent of i, the total computational complexity of finding all b_i^* s from (A.12) is $O(p)$.

References

[1] K.C. Aas and C.T. Mullis, Minimum mean-squared error transform coding and subband coding. *IEEE Trans. Inform. Theory* **42** (1996) 1179–1192.

[2] R. Bernardini, M. Naccari, R. Rinaldo and M. Tagliasacchi, S. Tubaro and P. Zontone, Rate allocation for robust video streaming based on distributed video coding. *Signal Processing: Image Communication* **23** (2008) 391–403.

[3] M. Blum, R.W. Floyd, V. Pratt, R.L. Rivest and R.E. Tarjan, Time bounds for selection. *J. Comput. System Sci.* **7** (1973) 448–461.

[4] C. Caini and A. Vanelli-Coralli, Optimum bit allocation in subband coding with nonideal reconstruction filters. *IEEE, Signal Process. Lett.* **8** (2001) 157–159.

[5] H. Cherroun, A. Darte and P. Feautrier, Reservation table scheduling: branch-and-bound based optimization vs. integer linear programming techniques. *RAIRO: RO* **41** (2007) 427–454.

[6] R. Crochiere, S. Webber and J. Flanagan, Digital coding of speech in sub-bands. In *IEEE International Conference on ICASSP'76. Acoustics, Speech, and Signal Processing*, IEEE **1** (1976) 233–236.

[7] H. Everett III, Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Oper. Res.* **11** (1963) 399–417.

[8] P. Feautrier, Parametric integer programming. *RAIRO: RO* **22** (1988) 243–268.

[9] B. Fox, Discrete optimization *via* marginal analysis. *Manage. Sci.* **13** (1966) 210–216.

[10] E. Gershikov and M. Porat, On color transforms and bit allocation for optimal subband image compression. *Signal Processing: Image Communication* **22** (2007) 1–18.

[11] A. Gersho and R.M. Gray, Vector quantization and signal compression. Springer (1992).

[12] Ch. Grauel, Sub-band coding with adaptive bit allocation. *Signal Processing* **2** (1980) 23–30.

[13] M. Guignard and S. Kim, Lagrangean decomposition for integer programming: theory and applications. *RAIRO: RO* **21** (1987) 307–323.

[14] M. Hatam and M. Ali, Masnadi-Shirazi, Quantization of multiple sources using modified optimum nonnegative integer adaptive bit allocation. *ICEE2008 Conference, Iran* (2008).

[15] J.K. Karlof, Integer programming: theory and practice. CRC Press (2005).

[16] E. Donald Knuth. The art of computer programming. *Sorting and Searching* **3** (1999) 426–458.

[17] D. Li and X. Sun, Nonlinear Integer Programming. Vol. 84. Springer (2006).

[18] K.-Kuang Ma and S.A. Rajala, Generalized optimum dynamic bit allocation scheme for source compression. In vol. 2 of *Proc. of ICIP-94. IEEE International Conference on, Image Processing.* (1994) 864–868.

[19] M.K. Mihcak, P. Moulin, M. Anitescu and K. Ramchandran, Rate-distortion-optimal subband coding without perfect-reconstruction constraints. *IEEE Trans. Signal Process.* **49** (2001) 542–557.

[20] A. Schrijver, Theory of linear and integer programming. John Wiley & Sons (1998).

[21] P.P. Vaidyanathan, Theory of optimal orthonormal subband coders. *IEEE Trans. Signal Process.* **46** (1998) 1528–1543.

[22] M. Vetterli and J. Kovačević, *Wavelets and subband coding*, Prentice Hall PTR Englewood Cliffs. Vol. 87. New Jersey (1995).

[23] J.P. Walser, Integer optimization by local search: a domain-independent approach. Springer-Verlag (1999).

[24] G. Wang, Analysis of quantization errors in subband speech coding with modified dft filter banks. *Signal Process.* **86** (2006) 341–352.

[25] X. Wei, M.J. Shaw and M.R. Varley, Optimum bit allocation and decomposition for high quality audio coding. In vol. 1 of *IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP-97.* (1997) 315–318.