



## OPTIMIZING LOSSLESS COMPRESSION BY NORMALIZED DATA LENGTH IN HUFFMAN ALGORITHM



TONNY HIDAYAT

DOCTOR OF PHILOSOPHY

2022



**Faculty of Information and Communication Technology**

**OPTIMIZING LOSSLESS COMPRESSION BY NORMALIZED DATA  
LENGTH IN HUFFMAN ALGORITHM**

**Tonny Hidayat**  
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

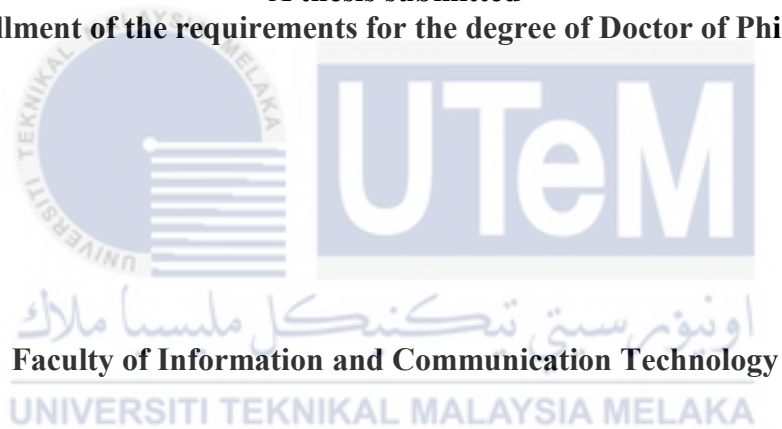
**Doctor of Philosophy**

**2022**

**OPTIMIZING LOSSLESS COMPRESSION BY NORMALIZED DATA LENGTH  
IN HUFFMAN ALGORITHM**

**TONNY HIDAYAT**

**A thesis submitted  
in fulfillment of the requirements for the degree of Doctor of Philosophy**



**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

**2022**

## DECLARATION

I declare that this thesis entitled “Optimizing Lossless Compression by Normalized Data Length in Huffman Algorithm” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Signature :   
Name : Tonny Hidayat  
Date : 14/04/2022



اونيورسيتي تيكنيكل مليسيا ملاك  
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

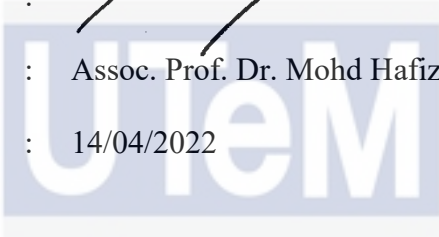
## APPROVAL

I hereby declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of the Doctor of Philosophy.

Signature : 

Supervisor Name : Assoc. Prof. Dr. Mohd Hafiz Zakaria

Date : 14/04/2022



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

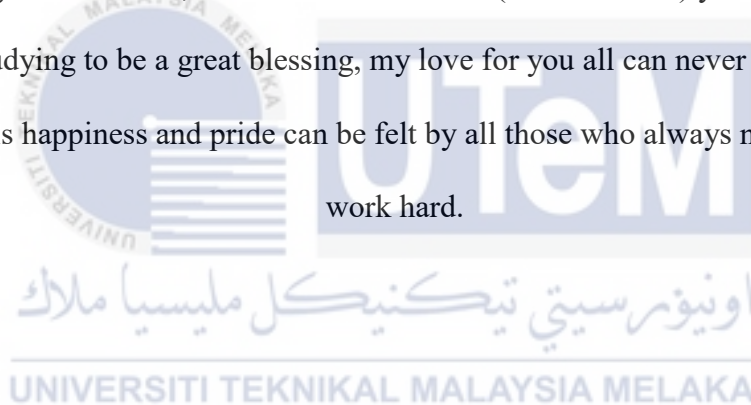
## DEDICATION

I dedicated my thesis work to my family.

Special thanks to my loving parents, Aflahah and Wardah, who have always been the motivation and encouragement of my life.

For my little family, Elfin Aquarista Putri (Wife) was the one who sacrificed the most for me during this study, Levano Arsakha Valin (First Child) forgive Ayah because time and attention for you is reduced, Revalinka Aluna Saira (Second Child) you were born when Ayah is studying to be a great blessing, my love for you all can never be quantified.

Hopefully, this happiness and pride can be felt by all those who always never give up and work hard.



## ABSTRACT

Due to the grown need of storage space, the demand for efficient compression scheme becomes increasingly important. One of the lossless data compression goals is to archive raw audio data to ensure the file is restored to the original form when it is to be reused. Generally, raw data is stored as 16-bit (65,536 difference possible values). Huffman Algorithms is currently still very effective at compressing 8-bit data, which can be grouped into Static, Dynamic, and Adaptive extensions, however its performance cannot be determined if it is performed on data that has several variables and probabilities. Based on the literature review, the measurement of the compression performance for files archives is to use the Compression Ratio (CR) and Compression Time (CT) indicators. These two indicators are used to calculate and analyse the file size reduction and the ability of the file to be reconstructed back to its original form without compromising its quality. This research produces a new scheme called Quaternary Arity (4-ary) Modification Quadtree (MQ) or 4-ary/MQ based on entropy coding which has its roots in other variants of Huffman schemes such as Binary / Static, Quadtree, Octatree, and Hexatree. The 4-ary/MQ method employs the characteristics of the Quadtree structure and extends the Dynamic Huffman coding mechanism (FGK rule) in node arrangement while adopting the Adaptive Huffman method that uses additional variable data. The novelty of this scheme is the work of adding additional variables to maintain the branch root to ensure it is always consistent with four branches. A descriptive analysis of the 4-ary/MQ was performed on several audio datasets (Music, Mono Music, Stereo Music, Ripping CD, Speech, Noise, Sound Effects, and Instruments) to compare with the Huffman Schematic Variant. A comparative analysis with several lossless compression applications has significantly shown that CR is more optimal than PKZIP, WinZip, 7-Zip, and Monkeys Audio. It was found that the 4-ary/MQ compression benefits the compressed data that is stored in local storage media as well as for hosting and optimizing bandwidth. The new algorithm also has a good performance in producing optimal CR with fast CT in most of the 16-bit WAV audio datasets. The proposed new algorithm has more optimal CR than the various variants of the Huffman-based lossless application. It is also expected that this new algorithm scheme may potentially work well on data above 16-bit for future research.

## **MENGOPTIMUMKAN PEMAMPATAN TANPA HILANG MELALUI PANJANG DATA TERNORMAL DALAM ALGORITMA HUFFMAN**

### **ABSTRAK**

Keperluan ruang simpanan yang semakin meningkat telah menjadikan keperluan untuk skema pemampatan yang efisien semakin penting. Salah satu tujuan pemampatan data berjenis tanpa hilang adalah untuk penyimpanan data audio berjenis raw bagi memastikan fail dapat dikembalikan ke bentuk asal apabila ingin digunakan semula. Secara amnya, data raw disimpan sebagai 16-bit (65,536 perbezaan kemungkinan nilai). Algoritma Huffman pada masa ini masih sangat berkesan dalam memampatkan data 8-bit, yang dapat dikategorikan sebagai berjenis statik, dinamis, dan adaptif, namun prestasinya tidak dapat ditentukan pada data yang memiliki sejumlah pemboleh ubah dan kebarangkalian. Berdasarkan tinjauan literasi, pengukuran prestasi pemampatan untuk fail adalah dengan menggunakan indikator Nisbah Pemampatan (NP) dan masa pemampatan (MP). Kedua-dua indikator ini digunakan untuk mengira dan menganalisis pengurangan ukuran fail dan kemampuan fail untuk disusun kembali ke bentuk asalnya tanpa menjejaskan kualitinya. Penyelidikan ini menghasilkan skema baharu yang disebut *Quarternary Arity (4-ary) Modification Quadtree (MQ)* atau *4-ary/MQ* berdasarkan pengekodan entropi yang berasal dari pelbagai varian lain dari skema Huffman seperti *Binary / Static*, *Quadtree*, *Octatree*, dan *Hexatree*. Kaedah *4-ary/MQ* menggunakan ciri struktur *Quadtree* dan memperluaskan mekanisme pengekodan Huffman dinamik (peraturan FGK) dalam susunan nod, dengan mengadaptasi kaedah Huffman adaptif yang menggunakan data pemboleh ubah tambahan. Keunikan skema ini adalah pada penambahan pemboleh ubah tambahan untuk mengekalkan akar cabang bagi memastikannya sentiasa konsisten dengan empat cabang. Analisis deskriptif ke atas *4-ary/MQ* dilakukan pada beberapa set data audio (Muzik, Muzik Mono, Muzik Stereo, Ripping CD, Ucapan, Kebisingan, Kesan Bunyi, dan Instrumen) untuk dibandingkan dalam varian Skema Huffman. Analisis perbandingan dengan beberapa aplikasi pemampatan tanpa hilang lain telah menunjukkan bahawa NP lebih optimum berbanding PKZIP, WinZip, 7-Zip, dan Monkeys Audio. Kajian juga mendapati bahawa pemampatan *4-ary/MQ* memberi keuntungan kepada data termampat yang disimpan di media penyimpanan tempatan dan juga untuk hosting dan mengoptimumkan lebar jalur. Algoritma baru ini juga mempunyai prestasi yang baik dalam menghasilkan NP yang optimum dengan MP pantas di kebanyakan set data audio WAV 16-bit. Algoritma baru yang dicadangkan mempunyai NP yang lebih optimum daripada pelbagai varian aplikasi tanpa hilang berasaskan Huffman. Diharapkan skema algoritma baru ini berpotensi untuk berfungsi dengan baik pada data di atas 16-bit untuk penyelidikan lain di masa akan datang.



## ACKNOWLEDGEMENTS

In the Name of Allah, the Most Gracious, the Most Merciful

First and foremost, all praise and glory to Almighty Allah (Subhanahu Wa Taalaa), my Creator, my Sustained, for everything I received since the beginning of my life. Shalawat and Salam to our Prophet Muhammad (Peace Be upon Him) who has brought us out from the darkness into the brightness world that full of knowledge.

I would like to extend my appreciation to the Universitas Amikom Yogyakarta and Universiti Teknikal Malaysia Melaka (UTeM) for providing the research platform. My utmost appreciation goes to my main supervisor Assoc. Prof. Dr. Mohd Hafiz Zakaria (UTeM) for all his support, advice and inspiration. His constant patience for guiding and providing priceless insights will forever be remembered. Also, to my co-supervisor, Assoc. Prof. Dr. Ahmad Naim Bin Che Pee (UTeM) who constantly supported my journey.

Last but not least, from the bottom of my heart a gratitude to my beloved wife, Elfin Aquarista Putri, for her encouragements and who have been the pillar of strength in all my endeavours. My eternal love also to all my children, Levano Arsakha Valin, and Revalinka Aluna Saira, for their patience and understanding. I would also like to thank my beloved parents for their endless support, love, and prayers. Finally, thank you to all the individual(s) who had provided me the assistance, support, and inspiration to embark on my study.

## TABLE OF CONTENTS

	PAGE
<b>DECLARATION</b>	
<b>DEDICATION</b>	
<b>ABSTRACT</b>	<b>i</b>
<b>ABSTRAK</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>xv</b>
<b>LIST OF APPENDICES</b>	<b>xx</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xxi</b>
<b>LIST OF PUBLICATIONS</b>	<b>xxiii</b>
<b>CHAPTER</b>	
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Background of the study	1
1.2 Research problems	3
1.3 Research questions	6
1.4 Research objectives	7
1.5 Purpose of the study	9
1.6 Research scopes	10
1.7 Research contributions	12
1.8 Organization of the thesis	13
<b>2. LITERATURE REVIEW</b>	<b>15</b>
2.1 Introduction	15
2.2 The issue of the large data size problem	16
2.3 Fundamental of lossless compression	17
2.4 Information theory and entropy	25
2.5 Huffman algorithm	27
2.5.1 Huffman code	29
2.5.2 Adaptive (dynamic) compression	33
2.5.3 Dynamically decreasing a node weight	50
2.5.4 Canonical Huffman codes	56
2.5.5 Prefix codes	58
2.6 Lossless compression applications	59
2.6.1 PKZIP	60
2.6.2 WinZip / WinRar	61
2.6.3 7-Zip	62
2.6.4 Monkeys Audio	63
2.7 Measurement of performance indicators	64
2.8 Research gap analysis	70
2.9 Conclusion	74

<b>3.</b>	<b>RESEARCH METHODOLOGY</b>	<b>76</b>
3.1	Research Design	76
3.2	Samples data	78
3.2.1	16-Bit data	81
3.2.2	WAV file format	84
3.2.3	Sound noise	87
3.3	Process settings	89
3.4	Discussion of the proposed bit allocation Huffman coding algorithm	93
3.4.1	Huffman scheme with binary tree (Static)	103
3.4.2	Huffman scheme with Quadtree	106
3.4.3	Huffman scheme with Octatree	113
3.4.4	Huffman scheme with Hexatree	119
3.4.5	Proposed Huffman scheme with 4-ary/Modif Quadtree	124
3.4.6	Comparison of results of all Huffman tree schemes	137
3.5	Calculation of data lossless compression	138
3.6	Data process compression and decompression collection	140
3.7	Experimental setting	142
3.8	Conclusion	143
<b>4.</b>	<b>IMPLEMENTATION RESULTS</b>	<b>145</b>
4.1	Introduction	145
4.2	Overview of research data processing	145
4.3	Compression process results and execution time on the Huffman tree schematic variants	148
4.3.1	IRMAS instruments dataset	148
4.3.2	FSDnoisy18k dataset	160
4.3.3	RipCD dataset	160
4.3.4	Musan dataset	161
4.3.5	Music orchset dataset	166
4.3.6	General conclusion results from all variants of the Huffman scheme	167
4.4	Comparison of 4-ary/MQ ratio with lossless compression applications	170
4.4.1	IRMAS instruments dataset	170
4.4.2	FSDnoisy18k dataset	181
4.4.3	RipCD dataset	181
4.4.4	Musan dataset	182
4.4.5	Dataset music orchset	188
4.4.6	General conclusion results from all lossless compression applications	190
4.5	Conclusion discussion of the results	191
<b>5.</b>	<b>ANALYSIS AND DISCUSSION</b>	<b>194</b>
5.1	Introduction	194
5.2	Analysis of compression process results on Huffman tree schematic variants	195
5.2.1	Compression ratio Huffman tree schematic variants	195

5.2.2	Space-saving percentage Huffman tree schematic variants	200
5.2.3	Compression time Huffman tree schematic variants	205
5.2.4	ANOVA and DMRT test Huffman schematic variants	208
5.3	Comparative analysis of 4-ary/MQ compression process with several lossless compress applications	212
5.3.1	Compression ratio several lossless compress applications	212
5.3.2	Space saving percentage several lossless compress applications	216
5.3.3	ANOVA and DMRT test several lossless compress applications	221
5.4	Benefits of using 4-ary/MQ on cloud storage costs and bandwidth	223
5.4.1	Comparison and estimation of cost per GB storage	226
5.4.2	Comparison and estimation of cost per GB bandwidth	232
5.5	Conclusion of the analysis discussion	237
<b>6.</b>	<b>CONCLUSIONS</b>	<b>239</b>
6.1	Conclusions and research contributions	239
6.2	Conclusions on the comparison of variants of the Huffman tree scheme	241
6.3	Conclusions on the comparison of other lossless compress applications	243
6.4	Conclusions on the comparison of cloud storage costs and bandwidth	243
6.5	Implication	244
6.6	Recommendations	245
	<b>REFERENCES</b>	<b>247</b>
	<b>APPENDICES</b>	<b>275</b>



## LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Data Source Example	30
2.2	Huffman Codes for the Symbols and Corresponding Huffman Tree	31
2.3	The Result of the Huffman Code Deviation	31
2.4	Final Huffman Codes After Input String 'Engineering'	44
2.5	Canonical Huffman Code	57
2.6	Illustration of Three Possible Prefix Codes	59
2.7	Performance Measurement Based on The Objectives of Compression	64
2.8	General Comparison of Huffman Binary Variants	72
3.1	Explanation of Wave File Structure	85
3.2	Adding Another Chunk to the WAV File	87
3.3	Alphabet Initials 5 Letters	96
3.4	Data String Sample	102
3.5	List of Symbols and Code for Encoding with the Static	104
3.6	Huffman Compression Results with a Binary Tree Static	106
3.7	Data for Compression with Quadtree	111
3.8	Huffman Compression Result with Quadtree	113
3.9	Data for Compression with Octatree	117
3.10	Huffman Compression Result with Octatree	119
3.11	Data for Compression with the Hexatree	122
3.12	Huffman Compression Result with Hexatree	124

3.13	Data Compressed with 4-ary/MQ Tree and with Normalization	135
3.14	Compression Results with 4-ary/MQ Tree with Tree Normalization	137
3.15	Comparison Results All Huffman Schematic Designs	138
3.16	Platform System Specifications Used for Implementation and Experiments	143
4.1	Terms of Process Calculation Units	146
4.2	Accumulated Total Source Size	147
4.3	Compression Result of IRMAS Music	149
4.4	Compression Result of IRMAS Speech	150
4.5	Compression Results of IRMAS Instrument Cello (cel)	151
4.6	Compression Result of IRMAS Instrument Clarinet (cla)	152
4.7	Compression Results of IRMAS Instrument Flute (flu)	153
4.8	Compression Results of IRMAS Instrument Acoustic Guitar (gac)	153
4.9	Compression Results of IRMAS Instrument Electric Guitar (gel)	154
4.10	Compression Results of IRMAS Instrument Organ (org)	155
4.11	Compression Results of IRMAS Instrument Piano (pia)	156
4.12	Compression Results of IRMAS Instrument Saxophone (sax)	156
4.13	Compression Results of IRMAS Instrument Trumpet (tru)	157
4.14	Compression Results of IRMAS Instrument Violin (vio)	158
4.15	Compression Result of IRMAS Instrument Human Singing Voice (voi)	159
4.16	Compression Results of IRMAS Instrument All	159
4.17	Compression Results of FSDnoisy18k.audio_test	160
4.18	Compression Result of RipCD	161

4.19	Compression Results of Musan Music Fma	161
4.20	Compression Result of Musan Music Fma-western	162
4.21	Compression Results of Musan Music Hd-classic	163
4.22	Compression Result of Musan Music Jamendo	163
4.23	Compression Results of Musan Music Rfm	164
4.24	Compression Results of Musan Noise	165
4.25	Compression Result of Speech Librivox	165
4.26	Compression Results of the Musan Speech US	166
4.27	Compression Results of Orchest Mono	167
4.28	Compression Result of Orchest Stereo	167
4.29	Comparison of Compression Results Variants the Huffman Algorithm	168
4.30	4-ary/MQ Comparison with Lossless Compression Applications on the IRMAS Music	171
4.31	4-ary/MQ Comparison with Lossless Compression Applications on the IRMAS Speech	171
4.32	4-ary/MQ Comparison with Lossless Compression Applications on the IRMAS Instrument Cello (cel)	172
4.33	4-ary/MQ Comparison with Lossless Compression Applications on the IRMAS Instrument Clarinet (cla)	173
4.34	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Flute (flu)	174
4.35	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Acoustic Guitar (gac)	174

4.36	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Electric Guitar (gel)	175
4.37	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Organ (org)	176
4.38	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Piano (pia)	177
4.39	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Saxophone (sax)	177
4.40	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Trumpet (tru)	178
4.41	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Violin (vio)	179
4.42	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instrument Human Singing Voice (Voi)	180
4.43	Comparison of 4-ary/MQ with Lossless Compression Applications on the IRMAS Instruments All dataset	180
4.44	Comparison of 4-ary/MQ with Lossless Compression Applications on the Dataset FSDnoisy18k.audio_test	181
4.45	Comparison of 4-ary/MQ with Lossless Compression Applications on the RipCD	182
4.46	Comparison of 4-ary/MQ with Lossless Compression Applications on the Musan Music Fma	183



4.47	4-ary/MQ comparison with Lossless Compression Applications on the Musan Music Fma-Western	183
4.48	Comparison of 4-ary/MQ with Lossless Compression Applications on the Musan Music HD-Classic	184
4.49	Comparison of 4-ary/MQ with Lossless Compression Applications on the Musan Music Jamendo	185
4.50	Comparison of 4-ary/MQ with Lossless Compression Applications on the Musan Music Rfm	186
4.51	4-ary/MQ comparison with Lossless Compression Applications on the Musan Noise	187
4.52	Comparison of 4-ary/MQ with Lossless Compression Applications on the Musan Speech Librivox	187
4.53	4-ary/MQ Comparison with Lossless Compression Applications on the US Musan Speech	188
4.54	4-ary/MQ Comparison with Lossless Compression Applications on the Orchest Mono	189
4.55	4-ary/MQ comparison with Lossless Compression Applications on the Orchest Stereo	190
4.56	Comparison of 4-ary/MQ Compression with Lossless Compression Applications	190
5.1	Results of the CR Calculation for the Variant of the Huffman Tree Scheme Based on the Audio Types	197
5.2	Deviation on CR 4-ary/MQ and Average CR Based on Audio Type	198

5.3	The result of Average CR Calculation Based on the Variant of the Huffman Tree Scheme	199
5.4	Correlation Between CR and SSc Variants of the Huffman Scheme	200
5.5	Outcomes from the SSc Variant of the Huffman Tree Scheme Calculations Based on the Audio Type	202
5.6	Deviation SSc Between 4-ary/MQ and Average Based on Audio Type	204
5.7	Results of the Average SSc Calculation Based on the Variant of the Huffman Tree Scheme	205
5.8	The Result of CT Calculation Variant of the Huffman Tree Scheme Based on the Audio Type	207
5.9	The Results of the Average CT Calculation Based on the Variant of the Huffman Tree Scheme	208
5.10	ANOVA Test Results CR Variant of the Huffman Scheme	209
5.11	DMRT Test Results CR Variant of the Huffman scheme	210
5.12	ANOVA Test Results CT Variant of the Huffman Scheme	211
5.13	DMRT Test Results CT Variant of the Huffman Scheme	211
5.14	The Result of CR Calculation of Huffman Based Lossless Compression Applications Based on Audio Type	214
5.15	Result of Average CR Calculation Based on Lossless Compression Applications	216
5.16	Correlation Between CR and SSc Variants of Lossless Compression Applications and 4-ary/MQ	217

5.17	Comparison of the SSc Value (%) for Lossless Compression Applications and 4-ary/MQ	219
5.18	Average SSc Calculation Result Based on Lossless Compression Applications	221
5.19	ANOVA Test Results for Lossless Compression Applications	222
5.20	DMRT Test Results Are Compression Results of Lossless Compression Applications	223
5.21	Comparison of Original and Compressed File Size on Huffman Variant (MB)	225
5.22	Comparison of Original and 4-ary/MQ Compressed File Size with Lossless Compression Applications	225
5.23	Charges for Cloud Storage and Bandwidth	226
5.24	Storage Costs Using the 4-ary/MQ	227
5.25	Comparison of Storage Costs for Each Service Provider by File Type	228
5.26	Storage Cost Savings After 4-ary/MQ Compressed	229
5.27	Total Cost Savings in Storing Compressed Audio Files Across Huffman Component Variants	230
5.28	Total Cost Savings in Storing Compressed Audio Files Between Lossless Compression Applications	232
5.29	Bandwidth Charges Using 4-ary/MQ Applications	233
5.30	Bandwidth costs before and after being compressed	233
5.31	Total Bandwidth Cost Savings of Compressed Audio Files Between Huffman Component Variants	235

5.32 Total Cost Savings in Compressed Audio File Bandwidth Between  
Lossless Compression Applications

236



## LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	K-Chart Scope of Study	11
2.1	Illustration of Lossless Compression	18
2.2	Illustration of Lossy Compression	18
2.3	System Schematic. <A> Compression. <B> Decompression	20
2.4	Code Tree According to Huffman	30
2.5	FGK Pseudocode Updates the Algorithm Tree	37
2.6	Knuth Detail Update Procedure	38
2.7	(a) Move q to the Right of its Block, (b) Exchange Procedure	39
2.8	Transfer q to the Next Block Subroutine	39
2.9	Adaptive Encoding Procedure from Dynamic	40
2.10	Adaptive Algorithm Example, 'e' Input to Tree	41
2.11	Adaptive Algorithm Example, 'n' and 'g' Input to Tree	42
2.12	Adaptive Algorithm Example, 'i' and second 'n' Input to Tree	43
2.13	Adaptive Algorithm Example, Input of the Two e's in the String 'enginee'	43
2.14	Adaptive Algorithm Example, Adding the 'r' (a). and the Final 'ing' (b)	43
2.15	Sibling Property Illustration	46
2.16	Core Pseudocode for Vitter's Algorithm $\Lambda$	49
2.17	Decreasing a Node 'e' Weight by One, to 2	51
2.18	Decreasing a Node 'e' Weight by One, to 1	51
2.19	Decreasing a Node 'e' Weight by One, to 0	51

2.20	Node 'e' is Absorbed by the NYT Node	51
2.21	Frequent-k Algorithm Example, $k = 5$	55
2.22	Space Saving Algorithm Example, $k = 5$	55
3.1	Research Design	77
3.2	Two Unsigned Bytes Together Form a 16-bit Unsigned Word	82
3.3	16-Bit Continuous Data	83
3.4	The Low Byte of the 16-bit Continuous Data	83
3.5	The High Byte of the 16-bit Data	83
3.6	Wave File Format Diagram	84
3.7	Interpret Each Byte of a WAV File	86
3.8	Wave Forms of White, Pink, and Brown Noise	89
3.9	Compression and Decompression Process Scheme	90
3.10	Overview of Compression Flow	91
3.11	Process Compression (Encoding)	91
3.12	Process Decompression (Decoding)	92
3.13	The Parsing Process	92
3.14	Binary Tree in 4 Levels	96
3.15	Pruning Binary Tree to Get Custom Codeword Length	97
3.16	Generates Huffman Code	97
3.17	Pseudocode Encoding Huffman Binary Tree Static	98
3.18	Pseudocode Decoding Huffman Binary Tree Static	99
3.19	Building the Huffman Binary Tree Static	103
3.20	Pseudocode Encoding Huffman Quadtree	108

3.21	Pseudocode Decoding Huffman Quadtree	109
3.22	Building the Huffman Quadtree	110
3.23	Pseudocode Encoding Huffman Octatree	114
3.24	Pseudocode Decoding Huffman Octatree	115
3.25	Building the Huffman Octatree	116
3.26	Pseudocode Encoding Huffman Hexatree	120
3.27	Pseudocode Decoding Huffman Hexatree	121
3.28	Building the Huffman Hexatree	121
3.29	Pseudocode First Form Encoding Huffman 4-ary/MQ Tree	126
3.30	Building the Huffman 4-ary/MQ Tree	127
3.31	Pseudocode Encoding Huffman 4-ary/MQ with Normalization	130
3.32	Flowchart Encoding Huffman 4-ary/MQ Tree	131
3.33	Pseudocode Decoding Huffman 4-ary/MQ Tree	132
3.34	Flowchart Decoding Huffman 4-ary/MQ Tree	133
3.35	Building the Huffman 4-ary/MQ with Normalization Tree	134
3.36	Compression Result Data Collection	142
4.1	Compression Comparison Results of Different Types of Huffman Algorithms	169
4.2	Comparison of 4-ary/MQ with Lossless Applications	191
5.1	Graph CR of Variant Huffman Tree and 4-ary/MQ	196
5.2	Graph CR Average Huffman Tree and 4-ary/MQ Schematic Variants	199
5.3	Graph SSc Variant Huffman Tree and 4-ary/MQ Schema Variants	202
5.4	Graph SSc Mean Schematic Variants of Huffman Tree and 4-ary/MQ	204

5.5	Graph of CT (MBps) Variants of the Huffman Tree and 4-ary/MQ Schemes	206
5.6	Graph Comparison of Average CT (MBps) of Huffman and 4-ary/MQ Variants	207
5.7	Comparison Graph of the CR (%) Value of Lossless Compression Applications and 4-ary/MQ	213
5.8	Comparison of Average CR (%) Value in Lossless Compression Applications	215
5.9	Comparison Graph of the SSc Value (%) Lossless Compression Applications and 4-ary/MQ	218
5.10	Comparison of SSc Mean Value (%) in Lossless Compression Applications	221
5.11	Comparison of Original and Compressed File Size on Huffman Variant (MB)	224
5.12	Comparison of Original and 4-ary/MQ Compressed File Size with Lossless Compression Applications	225
5.13	Storage Costs Using the 4-ary/MQ	227
5.14	Graph of Storage Cost Savings After 4-ary/MQ Compressed	228
5.15	Comparison Graph of Total Cost Savings in Storing Compressed Audio Files Between Variants of Huffman Components	229
5.16	Comparison Graph of Total Cost Savings in Storing Compressed Audio Files Between Lossless Compression Applications	231
5.17	Bandwidth Cost Graph Using 4-ary/MQ Applications	232