

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Compressão de Sinais para *fronthaul* em arquitetura CRAN utilizando Algoritmo Evolutivo

Vitória Alencar de Souza

DM: 19/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil

2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Vitória Alencar de Souza

Compressão de Sinais para *fronthaul* em arquitetura CRAN utilizando Algoritmo Evolutivo

DM: 19/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Vitória Alencar de Souza

Compressão de Sinais para *fronthaul* em arquitetura CRAN utilizando Algoritmo Evolutivo

Dissertação de mestrado submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA, como requisito para a obtenção do Grau de Mestre em Engenharia Elétrica na área de Telecomunicações.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil

2019

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

S719c Souza, Vitória Alencar de
Compressão de Sinais para fronthaul em arquitetura CRAN
utilizando Algoritmo Evolutivo / Vitória Alencar de Souza. — 2019.
91 f. : il. color.

Orientador(a): Prof. Dr. João Crisóstomo Weyl Albuquerque
Costa

Dissertação (Mestrado) - Programa de Pós-Graduação em
Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal
do Pará, Belém, 2019.

1. Algoritmo genético. 2. Quantização vetorial. 3.
Algoritmo de compressão. 4. Compressão de sinais,. I. Título.

CDD 621.3822

Compressão de Sinais para *fronthaul* em arquitetura CRAN utilizando Algoritmo Evolutivo

Dissertação de mestrado submetida à avaliação da banca examinadora aprovada pelo colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará e julgada adequada para obtenção do Grau de Mestre em Engenharia Elétrica na área de Telecomunicações.

Aprovada em ____ / ____ / ____

Prof. Dr. João Crisóstomo Weyl Albuquerque Costa

(ORIENTADOR - UFPA)

Prof. Dra. Jasmine Priscyla Leite de Araujo

(AVALIADORA - UFPA)

Prof. Dr. Rosinei de Sousa Oliveira

(AVALIADOR - UFOPA)

Prof. Dr. Gilvan Soares Borges

(AVALIADOR - IFPA)

Dra. Maria Emília de Lima Tostes

(COORDENADORA DO PPGE)

Ofereço este trabalho à todos que sempre acreditaram em
mim.

Agradecimentos

Primeiramente dedico e agradeço as minhas melhores amigas Silvia e Wana pela paciência nas noites em que sofri com a insegurança e por sempre acreditarem em mim.

Dedico também a toda a minha família pelas mensagens de perseverança, em especial ao meu querido padrinho e a minha maninha do coração Micaele que me acompanhou nesta fase tão difícil.

Também dedico a minha psicóloga que me acompanhou nestes 2 anos sempre me incentivando a ser forte e lutar contra todas as barreiras impostas.

Aos meus dois pais, o pessoal e o profissional, por sempre acreditarem em mim e me incentivarem a ser a minha melhor versão todos os dias.

E por fim ao professor João pela oportunidade por ele concedida e toda a sua paciência e cuidado nesta reta final. E aos meus colegas do LEA, fundamentais nesta reta final.

Todos desempenharam um papel inigualável em minha educação.

*Engraçado, costumam dizer que tenho sorte.
Só eu sei que quanto mais me preparo,
mais sorte eu tenho.*

Anthony Robbins

Glossário

AG	Algoritmo Genético
BS	<i>Base Station</i> , estação base
BBU	<i>Base-Band Unit</i> , unidade de banda base
CAPEX	<i>Capital Expenditure</i> , custo de implantação
CP	<i>Cyclic Prefix</i> , prefixo cíclico
CPRI	<i>Common Public Radio Interface</i> , interface de rádio pública comum
CRAN	<i>Cloud Radio Access Network</i> , rede de acesso por rádio centralizado
EA	<i>Cloud Radio Access Network</i> , Algoritmo evolutivo
EP	<i>Evolutionary Programming</i> , Programação evolutiva
ES	<i>Evolution Strategies</i> , Estratégia evolutiva
EVM	<i>Error Vector Magnitude</i> , magnitude do vetor de erro
FLAC	<i>Free Lossless Audio Codec</i> , codec de áudio livre sem perdas
FFT	<i>Fast Fourier Transform</i> , transformada de Fourier
GLA	<i>Generalized Lloyd Algorithm</i> , algoritmo de Lloyd generalizado
ISI	<i>Inter-Symbol Interference</i> , interferência inter-simbólica
IFFT	<i>Inverse Fast Fourier Transform</i> , transformada rápida de Fourier inversa
LTE	<i>Long Term Evolution</i> , evolução de longo prazo
MSE	<i>Mean Squared Error</i> , erro quadrático médio
MSVQ	<i>Multistage Vector Quantization</i> , quantização vetorial em múltiplos estágios
OBSAI	<i>Open Base Station Architecture Initiative</i> , iniciativa aberta para arquiteturas de estações base
OFDM	<i>Orthogonal Frequency Division Multiplexing</i> , multiplexação das frequências por divisão de frequências ortogonais

OFDMA	<i>Orthogonal Frequency Division Multiple Access</i> , multiplexação do acesso por divisão de frequências ortogonais
QPSK	<i>Quadrature Phase Shift Keying</i> , modulação por chaveamento de fase
RB	<i>Resource Block</i> , bloco de recurso
RE	<i>Resource Element</i> , elemento de recurso
RRH	<i>Remote Radio Head</i> , central de rádio remoto
SQ	<i>Scalar Quantization</i> , quantização escalar
VQ	<i>Vector Quantization</i> , quantização vetorial

Lista de Símbolos

Símbolo	Descrição
B	Número de blocos normalizados
BW	Banda do sinal
$BW_{\text{útil}}$	Banda útil
C_{total}	Taxa de compressão total
C_{cp}	Taxa de compressão devido a remoção do CP
C_{down}	Taxa de compressão devido a sub-amostragem
C_{ent}	Taxa de compressão devido a codificação por entropia
C_{vq}	Taxa de compressão devido ao VQ
F_s	Frequência de amostragem
F_{nyquist}	Frequência definida no teorema de Nyquist
f_{filtro}	Frequência de amostragem do filtro
K	Fator de sobre-amostragem
L	Fator de sub-amostragem
L_{cp}	Comprimento do prefixo cíclico
L_{huff}	Comprimento médio do código de Huffman
L_i	Comprimento da palavra binária do dicionário i
L_{sym}	Comprimento do símbolo OFDMA
L_{vq}	Comprimento da palavra binária antes da codificação
N_{ant}	Número de antenas
N_b	Número de amostras por bloco
N_{ov}	Fator de redundâncias no CPRI
N_s	Número inicial de amostras
p_i	Probabilidade do dicionário i

Q_0	Número de bits transmitido no CPRI por amostra I/Q
Q_a	Número de bits utilizado na quantização dos valores máximos
Q_b	Número bits utilizados na quantização dos blocos
Q_{vq}	Número de bits por amostra complexa após o VQ
R_{CPRI}	Taxa de bits no CPRI
S_{eq}	Sinal equalizado
$S_{inicial}$	Sinal inicial
$S_{comprimido}$	Sinal comprimido
T_s	Período de amostragem

Lista de Figuras

1.1	Rede de acesso por radio centralizado, adaptado de [2].	2
2.1	Diagrama de blocos do OFDMA, retirada de [8].	9
2.2	Prefixo cíclico normal por símbolo OFDMA, retirada de [11].	10
2.3	Um <i>slot</i> de LTE, retirada de [6].	12
2.4	Banda útil transmitida em um sinal de LTE, retirada de [11].	14
2.5	Sinal no domínio da frequência antes do processo de sub-amostragem.	14
2.6	Filtro utilizado no processo de sub-amostragem quando $K = 2$	15
2.7	Filtro utilizado no processo de sub-amostragem quando $K = 5$	15
2.8	Sinal após o processo de sub-amostragem.	16
2.9	Esquema de quantização para um sinal de LTE normalizado.	18
2.10	Compressão obtida na taxa de bits.	19
2.11	CRAN utilizando <i>fronthaul</i> CPRI, retirada de [18].	21
2.12	<i>Frame</i> básico de CPRI, retirada de [20].	22
3.1	Cenário de atuação do algoritmo.	33
3.2	Histograma de 16384 <i>bins</i> de amostras reais e imaginárias de sinais de LTE	34
3.3	Algoritmo de treinamento resumido	35
3.4	Método de compressão desenvolvido.	36
3.5	Esquema de geração de dicionários ótimos.	38
3.6	Fluxograma do <i>software</i> que utilizando o LBG.	42
3.7	Fluxograma do <i>software</i> que faz o uso do K-means++.	43
3.8	Dicionário treinado e conjunto de treino.	46
4.1	Testes de decisão de parâmetros para o AG-LBG.	51
4.2	Testes de decisão de parâmetros para AG Kmeans.	52

4.3	Treinamento do dicionário de 1024 pontos com AG-LBG	54
4.4	Treinamento do dicionário de 1024 pontos com AG-Kmeans	54
4.5	Treinamento do dicionário de 4096 pontos com AG-LBG	55
4.6	Treinamento do dicionário de 4096 pontos com AG-Kmeans	55
4.7	EVMs obtidas utilizando o método proposto.	57
4.8	Dicionário de 1024 genes treinado com LBG e AG.	58
4.9	Dicionário de 1024 genes treinado com AG-Kmeans.	58
4.10	Dicionário de 4096 genes treinado com LBG e AG.	59
4.11	Dicionário de 4096 genes treinado com AG-Kmeans.	59
4.12	Dicionário de 16384 genes treinado com LBG e AG.	60
4.13	Dicionário de 16384 genes treinado com AG-Kmeans.	60

Lista de Tabelas

2.1	Comprimentos assumidos pelo prefixo cíclico <i>short</i> no primeiro e nos demais símbolos.	11
2.2	Características dos sinais de LTE , obtida em [15].	13
2.3	Metáfora de computação evolutiva, retirada de [29].	23
2.4	Categorias de EA, retirada de [29].	25
3.1	EVM máxima por transmissão de LTE.	37
3.2	Gene desenvolvido no trabalho.	40
3.3	Um indivíduo de 16 genes.	40
3.4	Metáfora de VQ utilizada no AG desenvolvido.	40
4.1	Valores de EVM reproduzidos utilizando somente sobre-amostragem.	49
4.2	Variáveis e valores utilizados nas simulações	50
4.3	Variáveis e valores utilizados nas simulações	53
4.4	MSE obtidos nos treinos de dicionários com LBG.	56
4.5	MSE obtidos nos treinos de dicionários com Kmeans.	56

Sumário

Agradecimentos	viii
Lista de Siglas	x
Lista de Símbolos	xii
Lista de Figuras	xiv
Lista de Tabelas	xvi
Sumário	xvii
1 Introdução	1
1.1 Motivação	2
1.2 Trabalhos Correlatos	3
1.2.1 Técnicas de Compressão de sinais de LTE em <i>fronthaul</i> CRAN	4
1.3 Objetivos gerais e específicos	6
1.4 Estrutura da dissertação	6
2 Referencial teórico	8
2.1 Compressão de sinais de LTE	9
2.1.1 Prefixo Cíclico	10
2.1.2 Sobre-amostragem	11
2.1.3 Normalização em blocos	16
2.1.4 Quantização Vetorial	17
2.1.5 Codificação por entropia	19
2.1.6 Compressão total	20
2.2 Visão geral do <i>fronthaul</i> CPRI das redes CRAN	21

2.3	Características dos sinais no <i>Fronthaul</i> CPRI	21
2.4	Algoritmos Genéticos	23
2.4.1	Codificação do indivíduo	24
2.4.2	Geração da população inicial	24
2.4.3	Função de Aptidão	25
2.4.4	Avaliação e Otimização Local	25
2.4.5	Seleção de parentes	27
2.4.6	Cruzamento	28
2.4.7	Mutação	29
2.4.8	Seleção de sobreviventes	30
2.4.9	Critério de convergência	31
2.5	Conclusão	32
3	Algoritmo de Compressão de Sinais de LTE	33
3.1	Metodologia do trabalho	34
3.2	Treino de dicionários	34
3.3	Algoritmo de teste e compressão de sinais de LTE	35
3.3.1	Geração dos sinais	36
3.3.2	Métrica de avaliação do Algoritmo de Compressão	36
3.4	Treinamento de Dicionário por Algoritmos Genéticos	38
3.4.1	Problema de Treinamento de Dicionários	38
3.5	Algoritmo Genético proposto	39
3.5.1	Codificação do indivíduo	39
3.5.2	Geração da população inicial	41
3.5.3	Função de Aptidão	41
3.5.4	Avaliação e Otimização Local	41
3.5.5	Seleção de parentes	44
3.5.6	Cruzamento	44
3.5.7	Mutação	44
3.5.8	Seleção de sobreviventes	45
3.5.9	Critério de convergência	45
3.5.10	Métrica de avaliação do desempenho dos dicionários	46
3.6	Conclusão	47

4	Resultados	48
4.1	Compressão utilizando somente sobre-amostragem	48
4.2	Resultados	49
4.2.1	Cenário final	50
4.2.2	Avaliação do Algoritmo de Treinamento	52
4.2.3	Avaliação do algoritmo de Compressão	56
4.2.4	Dicionários Treinados	57
4.3	Conclusão	61
5	Conclusão	62
5.1	Sugestões para trabalhos futuros	62
5.2	Publicações	63
5.3	Conclusão	63
	Bibliografia	65

Abstract

Centralized radio access networks are present as a potential alternative for next generation of wireless networks, because they are able to provide high data rates and allow the reduction of structural and operational costs in the network. The centralized architecture implements the concept of fronthaul, and enables the challenge to increase the capacity of data transmission in these links. In this way, the study of digital signal compression techniques presents itself as an alternative to reduce the cost of implementing centralized radio access networks. This work investigates the use of vector quantization methods in the compression of complex samples of baseband LTE signals. We propose the use of Genetic Algorithms in the training of sub-optimal dictionaries for the process of vector quantization in order to reduce the errors imposed in this process and consequent increase in fronthaul capacity. Results showed that the proposed compression algorithm allows reduction of fronthaul data rates associated with acceptable errors. It has been shown to be possible data rate compression factors of 5.4 times, with errors of approximately 4.4%, proving the effectiveness of codebook training process in LTE signals present in the downlink of centralized radio access networks.

Resumo

As redes de acesso por rádio centralizado apresentam-se como uma potencial alternativa para a próxima geração de telefonia celular, devido à sua capacidade de prover grandes taxas de dados e permitir a redução dos gastos estruturais e operacionais na rede. A arquitetura centralizada implementa o conceito de *fronthaul*, mas aponta o desafio para aumento da capacidade de transmissão de dados nesses enlaces. Desta forma, o estudo de técnicas de compressão de sinais digitais apresenta-se como uma alternativa para reduzir o custo de implantação das redes de acesso por rádio centralizado. Nesse contexto, este trabalho investiga o uso de métodos de quantização vetorial na compressão de amostras complexas de sinais de LTE em banda-base. Propõe-se o uso de Algoritmos Genéticos no treinamento de dicionários sub-ótimos para o processo de quantização vetorial com objetivo de reduzir os erros impostos neste processo e consequente aumento na capacidade do *fronthaul*. Os resultados mostram que o algoritmo de compressão proposto permite redução de taxas de dados em *fronthaul* associados a erros aceitáveis. Demonstra-se ser possível fatores de compressão de taxa de dados de 5,4 vezes, com erros de, aproximadamente, 4,4%, comprovando-se a efetividade do método de treinamento de dicionários de LTE no *downlink* das redes de acesso por rádio centralizado.

Capítulo 1

Introdução

Introdução

O desenvolvimento da tecnologia sem fio de quinta geração (5G) está em andamento. Ela tem sido moldada para atender às crescentes demandas por altas taxas de dados, baixa latência e acesso móvel onipresente instigado por aplicativos móveis, com processamento centralizado em nuvem [1].

É estimado que será necessário prover taxas de cerca de 100 à 1000 vezes as taxas de dados do 4G. Para tal fim, será necessária não somente a expansão radical das bandas de transmissão até altas frequências na faixa dos THz, mas também a promoção da densificação das redes celular e a utilização de um número massivo de antenas com o intuito de prover altas taxas de dados, com baixíssimo atraso e baixo consumo energético da rede [2].

Entretanto, há um forte apelo para que no desenvolvimento da próxima geração seja levado em conta o requerimento energético, impulsionando o estudo de técnicas que promovam o uso eficiente do espectro, reduzam o custo energético, diminuam a latência ponto a ponto e aumentem o número de conexões por ponto. Criando assim a perspectiva de pesquisa *green and soft*, que busca encontrar soluções que sejam responsáveis com o uso dos recursos ambientais [3].

Neste contexto, as Redes de Acesso por rádio centralizada (CRAN, *Cloud Radio Access Networks*) tem se mostrado como potencial alternativa para suprir aos requerimentos de redes de alta capacidade com baixíssimo consumo energético, pois a centralização do processamento proporciona uma arquitetura com menores necessidades de refrigeração e consequentemente com menor custo energético. A economia de energia proporciona uma redução de até

70% em comparação com as arquiteturas de 3G [3].

1.1 Motivação

Na arquitetura CRAN, as funcionalidades das estações base (BS, *Base Station*), desde a camada física até as camadas superiores, encontram-se virtualizadas em processadores centralizados de uso geral denominados unidades de banda-base (BBU, *Base-Band Unit*) [4].

A transmissão dos sinais ao usuário no entanto ocorre nas unidades de rádio remoto (RRH, *Remote Radio Heads*) que são separados fisicamente por enlace físico de alta capacidade denominado *fronthaul* como mostrado na Figura 1.1, adaptada de [1]. Esta ligação permite também a *cloudification* do processamento de sinais digitais que busca torná-lo flexível para que o mesmo também seja gerenciado pela BBU [4].

As interfaces típicas da *fronthaul* incluem a interface de rádio pública comum (CPRI, *Common Public Radio Interface*) e a iniciativa de arquitetura de estação de base aberta (OBSAI, *Open Base Station Architecture Initiative*). Como o CPRI é o protocolo de *fronthaul* mais utilizado na indústria, o trabalho descreve uma solução para um *fronthaul* CPRI [4].

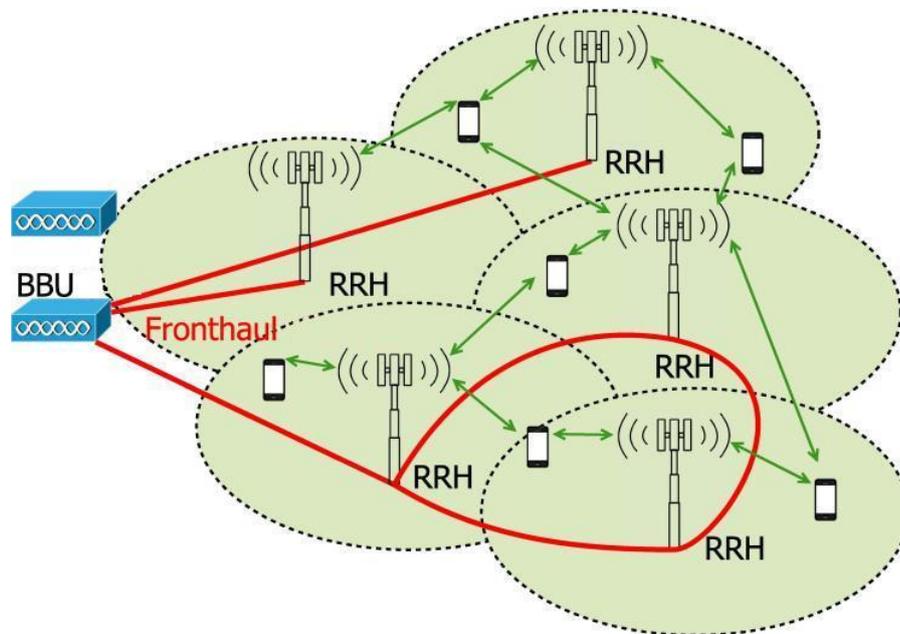


Figura 1.1: Rede de acesso por rádio centralizado, adaptado de [2].

Assim, ao virtualizar as funções de redes em um processador de propósito geral é possível construir redes de baixo custo de implantação (CAPEX, *Capital Expenditure*), operação (OPEX,

Operational Expenditure), grande eficiência energética e espectral pelo uso de redes definidas por *software* [3].

Entretanto, ao incrementar a centralização da rede, aumenta-se também o volume de dados necessário para coordenar as transmissões. Em outras palavras, a centralização pode consumir um grande número de recursos processamento e de fibra, o que aumenta o custo de implantação das redes e operação [4].

A crescente necessidade de expansão da capacidade das redes e conseqüentemente do *fronthaul*, motivou vários autores a conceber, para esse segmento, a compressão dos sinais, como alternativa à necessidade de ampliação dos enlaces (frequentemente de fibra), diminuindo os custos de implantação e operação da rede [5].

Desta forma, os principais desafios à implantação dos sistemas centralizados passaram a ser a virtualização das camadas de rede, a *cloudification* do processamento de sinais digitais e a limitada capacidade de transmissão dos enlaces de *fronthaul*, pois quanto maior for a centralização da rede, maior será a necessidade de implantação de *links* de fibra. O que torna onerosa a implantação das CRANs [4].

Este trabalho busca alternativas para aumentar as taxas de dados no *fronthaul* CPRI e reduzir o ruído imposto no processo de compressão, através do uso de técnicas de compressão de sinais de LTE em conjunto com os Algoritmos Genéticos.

O uso dos Algoritmos Genéticos foi proposto como uma alternativa aos algoritmos tradicionais, como forma de pesquisa de uma solução ótima, de mínimo ruído e maior compressão, dentro do espaço de busca.

1.2 Trabalhos Correlatos

A crescente ampliação das taxas de transmissão de amostras I/Q no *fronthaul* tem motivado diversos autores a proporem inúmeras técnicas de compressão de sinais complexos de LTE para *fronthaul* com o intuito de reduzir ampliar as taxas de transmissão e assim os custos de implantação e operação da arquitetura CRAN.

Os processos de compressão de sinais desenvolvidos para *fronthaul* da arquitetura CRAN podem ser divididos em duas categorias principais: Compressão multiterminal (*Multiterminal compression*) e a compressão Ponto-a-ponto [6].

As técnicas baseadas em compressão multi terminal, permitem no *downlink*, que as BBUs

para correlacionem os ruídos de quantização dos sinais em banda base que serão enviados a RRHs vizinhas. No *uplink*, esta técnica permite que a BBU aproveite a correlação entre os sinais oriundos de RRHs vizinhas e efetue processo de descompressão com menores erros.

As técnicas que empregam a compressão ponto-a-ponto fazem o uso de dicionários de quantização para o *downlink* e *uplink*. Estes dicionários são projetados independentemente dos dicionários de canal que são utilizados nas transmissões em canais sem fio, e são compatíveis com os mesmos, podendo assim serem utilizados em conjunto [6].

Esta revisão discutirá os principais métodos de compressão de sinais ponto-a-ponto, visto que o trabalho propõe uma técnica de treinamento de dicionários para compressão ponto-a-ponto. As técnicas de compressão ponto a ponto possuem diversas vantagens como um reduzido custo computacional de decodificação [7]. Deste forma são descritos a seguir os principais métodos compressão ponto a ponto são descritos a seguir.

1.2.1 Técnicas de Compressão de sinais de LTE em *fronthaul* CRAN

No *fronthaul* as amostras de LTE utilizam quantização escalar. Desta forma em [8], os autores desenvolveram um método baseado em *downsampling*, normalização em blocos e quantização escalar das amostras e do erro de quantização. As sucessivas quantizações proporcionaram ao método um baixo ruído e uma baixa latência, no entanto atinge pouca compressão por se tratar de um método de baixa complexidade.

Os artigos [9] e [10] propuseram técnicas que utilizam sobre-amostragem, normalização em blocos e quantização escalar das amostras e dos valores máximos dos blocos.

O conceito de treinamento de níveis de LTE foi introduzido no trabalho [9], pois, este trabalho faz o uso de um quantizador não uniforme foi treinado e testado com sinais de LTE em banda base.

Os métodos baseados em SQ atingem erros dados em EVM (*Error Vector Magnitude*) de cerca de 2%. No entanto proporcionam baixas taxas de compressão, da ordem de 3 vezes.

A VQ (VQ, *Vector Quantization*) é uma generalização bi-dimensional da SQ que introduz complexidade no treinamento e reduz o ruído de quantização. Os trabalhos [5] e [11] foram introduzidas a remoção do CP e são comparados diversas técnicas de quantização e fazem a codificação por entropia. As técnicas de quantização utilizadas são SQ, VQ e a MSVQ (*multistage vector quantization*). A MSVQ implementada mostrou ter complexidade de implementação reduzida, no entanto conduziu a maiores ruídos de quantização.

As técnicas baseadas em VQ reduziram os erros de compressão para cerca EVMs de 3% com taxa de compressão de 4,5 vezes. No entanto a solução encontrada possui alta complexidade, pois a decodificação dos sinais é feita utilizando-se busca em tabelas [7].

Em [12] foi implementada a re-amostragem dos frames CPRI e utilização de um *codec* de áudio denominado FLAC (FLAC *Free Lossless Audio Codec* [13]) obtendo taxas de compressão de até 2 vezes com EVMs menores que 2%. No entanto, o FLAC possui uma implementação em *fronthaul* com alto custo computacional que aumenta a latência.

No trabalho [14] é feita a remoção do CP e VQ utilizando *fast K-means* [15] nas subportadoras separadas, conseguindo alcançar taxas de compressão de cerca de 6 vezes com EVMs de 0,025%. No entanto este método possui uma grande latência pois os sinais I/Q são convertidos para o domínio da frequência para depois serem passarem pelo processo de clusterização sendo necessário receber um símbolo inteiro de OFDMA para que o processo se inicie.

O artigo [16] propõe o uso de uma técnica denominada *noise-shaping* [17], em conjunto com a quantização escalar não uniforme em conjunto com a remoção do prefixo cíclico e o *downsampling*, conseguindo taxas de compressão de 3 vezes com erros inferiores a 2%.

Este trabalho propõe o treinamento de dicionários de VQ utilizando-se o AG para aumentar as taxas de compressão e reduzir o ruído de compressão de dicionários treinados com auxílio do K-means++ [18] e do algoritmo LBG [19].

A compressão das amostras I/Q de Sinais de LTE é feita no *downlink* através da remoção do prefixo cíclico seguida de *dowsampling*, escalonamento por blocos e VQ.

A solução encontrada conduz a resultados similares aos propostos de [5] e [11] com o mesmo grau de complexidade e ainda possui latência igual a [14], pois os métodos baseados em VQ utilizam a decodificação por busca em tabelas [7].

A geração dicionários é feita antes da transmissão o que não produz novos atrasos. A solução conduz a resultados similares aos propostos em [5] e [11], e mostrou-se eficaz pois consegue atingir taxas de compressão de aproximadamente 5,4 vezes com EVMs de aproximadamente 4,4% indicando que o método é vantajoso.

1.3 Objetivos gerais e específicos

O estudo e implementação de técnicas de compressão de sinais tem se mostrado fundamental na implantação de redes 4G e posteriormente redes 5G.

Neste sentido o trabalho tem por objetivo implementar um algoritmo evolutivo para aplicado à compressão de sinais, de forma que este comprima os sinais com o mínimo de distorção possível, no *fronthaul* de redes de acesso por rádio centralizado.

O algoritmo busca remover as redundâncias presentes nos sinais de 4G, que podem ser reinseridas após o *fronthaul* aumentando a sua capacidade com ruídos de compressão mínimos. O trabalho avalia também, relação entre o ruído imposto no processo de compressão e o nível de compressão.

Para que tal objetivo objetivo se concretize, os seguintes objetivos específicos são mandatórios:

- Investigar as limitações do padrão de interface CPRI na implantação do *fronthaul* das redes CRAN.
- Investigar algoritmos de clusterização que podem ser aplicados ao treino dicionários para o problema de quantização vetorial.
- Desenvolver um algoritmo genético que realize a tarefa de clusterização para promover o treinamento de dicionários factíveis no domínio do problema.
- Investigar técnicas de compressão de sinais de LTE para efeito de comparação com o método proposto.
- Implementar um algoritmo que retire as redundâncias dos sinais de LTE e reporte o ruído produzido no processo de compressão.

1.4 Estrutura da dissertação

O trabalho proposto é organizado em 4 capítulos brevemente descritos a seguir:

No Capítulo 2 são expostas as principais etapas de compressão de sinais de LTE que são utilizadas no processo de compressão. Ademais é dada uma visão geral dos sinais no *fronthaul* da arquitetura CRAN e os Algoritmos Genéticos, que são uma classe de algoritmos evolutivos

muito utilizadas em problemas otimização de funções com múltiplas variáveis e ou múltiplos objetivos, são brevemente descritos.

O Capítulo 3 apresenta o *software* de compressão e o algoritmo evolutivo de treinamento de dicionários. O *software* de compressão desenvolvido que faz o uso dos Algoritmos genéticos no treinamento de dicionários, com o propósito de aumentar a taxa de compressão dos sinais e reduzir o ruído imposto no processo de quantização vetorial (VQ, *Vector Quantization*).

No Capítulo 4 os resultados são discutidos. O Capítulo 5 faz uma discussão final acerca do método proposto e propõe diversos trabalhos futuros relevantes.

Capítulo 2

Referencial teórico

Introdução

Os sinais de LTE utilizam na camada física em sua transmissão de *downlink* a multiplexação do acesso por divisão de frequência ortogonal (OFDMA, *Orthogonal Frequency-Division Multiple Access*), uma técnica multi-portadora que aloca recursos de rádio a vários usuários. O OFDMA utiliza a multiplexação por divisão de frequências ortogonais (OFDM, *Orthogonal Frequency Division Multiplexing*) [20].

O OFDM oferece diversas vantagens nas transmissões em canais sem-fio. Sendo assim, no OFDM há vários sub-canais de banda estreita não seletivos em frequência que são sobrepostos, mas não interferem entre si devido à sua ortogonalidade [21].

O uso do OFDMA permite que a estação base se comunique com vários celulares diferentes ao mesmo tempo. Além disso, essa tecnologia também implementa técnicas poderosas para aumentar a eficiência espectral e minimizar problemas de canais multi-percurso e interferência inter-simbólica (ISI, *Inter-Symbol Interference*) [22].

Na Figura 2.1, retirada de [22], apresenta-se um esquemático de como o sinal de OFDMA é criado. Os sinais que serão destinados a diversos usuários, primeiramente são mapeados para sub-portadoras complexas, utilizando modulações digitais. Tais sub-portadoras serão convertidas para o domínio real, na operação da transformada de Fourier inversa (IFFT, *Inverse Fast Fourier Transform*). Em seguida, o prefixo cíclico é adicionado e o sinal resultante é convertido para serial e então é transmitido.

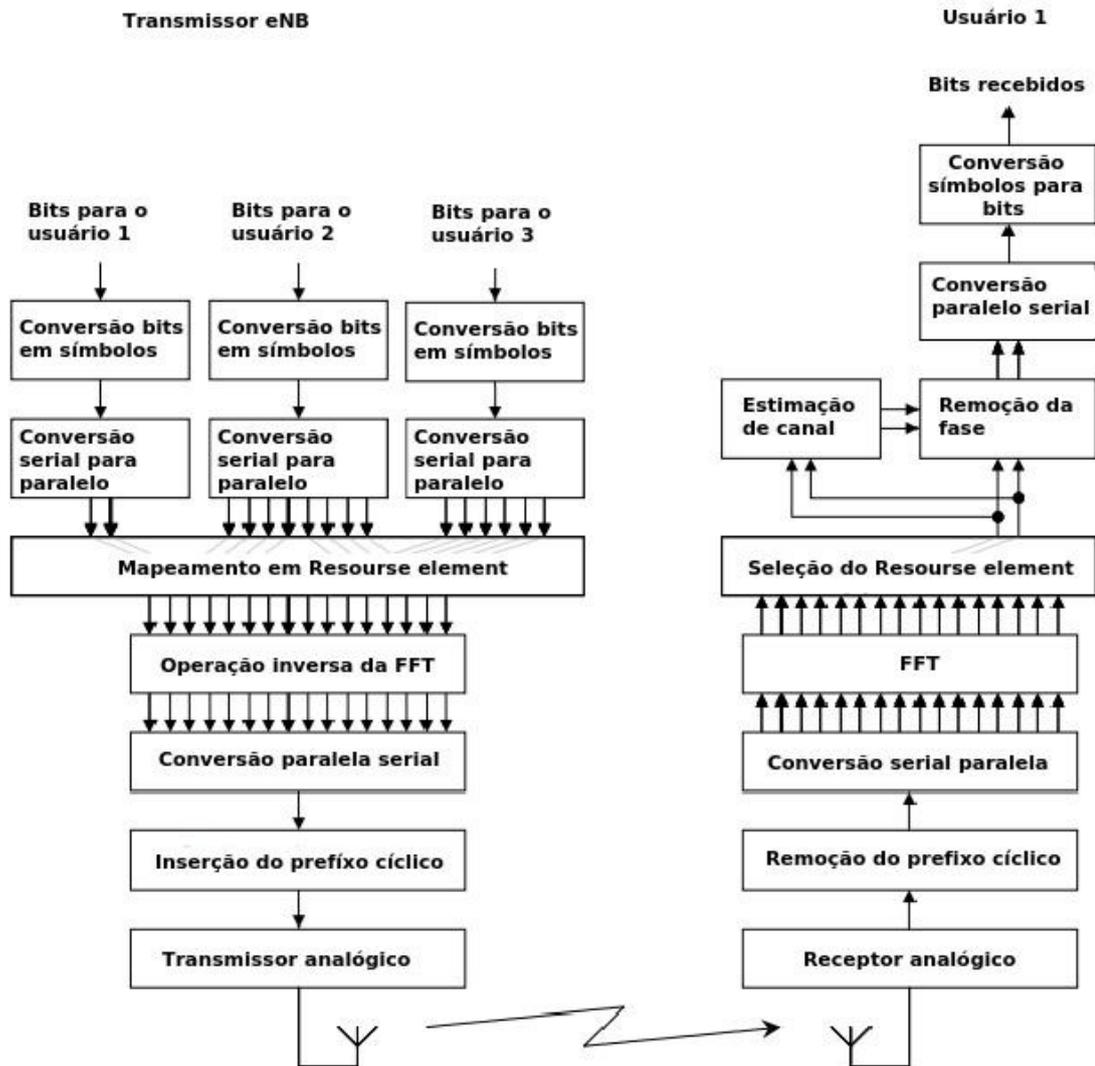


Figura 2.1: Diagrama de blocos do OFDMA, retirada de [8].

A seguir são descritas as redundâncias presentes nos sinais de LTE e suas utilidades. Em seguida são relatadas as principais técnicas de compressão de sinais de LTE.

2.1 Compressão de sinais de LTE

Os sinais de LTE em banda-base são compostos de diversas redundâncias que são inseridas para conferir robustez ao sinal em diversos requerimentos necessários à transmissão em canal sem-fio. No entanto, no contexto de transmissões feitas em *fronthaul*, estas redundâncias representam uma sobrecarga de dados que pode ser removida utilizando-se técnicas de compressão de sinais digitais.

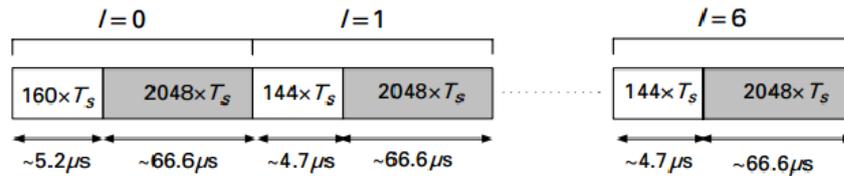


Figura 2.2: Prefixo cíclico normal por símbolo OFDMA, retirada de [11].

2.1.1 Prefixo Cíclico

O prefixo cíclico (CP, *Cyclic Prefix*) é uma redundância que é inserida no domínio de tempo. Tal CP consiste em um período de guarda que contém uma fração do final de um símbolo de OFDMA que é adicionada ao início do símbolo quando nada é transmitido [22].

Em canais seletivos em frequência ocorre a interferência entre as sub-portadoras. Logo, um canal dispersivo no tempo é equivalente a um canal com resposta em frequência seletiva. No entanto, a ortogonalidade entre as sub-portadoras de OFDMA não é simplesmente devida à separação no domínio da frequência, mas também à estrutura específica no domínio da frequência de cada subportadora [23].

Mesmo que o canal possua uma resposta em frequência constante, na largura de banda correspondente ao lóbulo principal de uma subportadora OFDMA, e apenas os lóbulos laterais da subportadora estiverem corrompidos, devido à seletividade em frequência de canal de rádio, a ortogonalidade entre sub-portadoras será perdida e ocorrerá interferência entre as sub-portadoras como consequência. Para lidar com este problema e fazer um sinal OFDMA verdadeiramente resistente à dispersão do tempo no canal de rádio, ocorre a chamada inserção de prefixo cíclico nas transmissões de OFDMA [23].

Na Figura , retirada de do livro [24], temos todos os comprimentos de prefixo cíclico, dos símbolos OFDMA em um slot de LTE. Onde T_s representa o período de amostragem e os valores $(160 \times T_s)$ e $(144 \times T_s)$ foram selecionados para que o resultado da soma em um *slot* seja um número inteiro de amostras de tamanhos da IFFT de 128, 256, 512, 1024 ou 2048, assim a operação de IFFT não seria prejudicada com a inserção do prefixo cíclico e nem com a sua remoção.

A Tabela 2.1, retirada do livro [25] lista os comprimentos do prefixo cíclico *short* nos domínios do tempo e da frequência, pois as amostras de símbolo de OFDMA são no domínio das frequências *subcarriers* com largura de 15 kHz .

A Desta forma, o prefixo cíclico pode ser removido sem que seja prejudicada a recepção

Tabela 2.1: Comprimentos assumidos pelo prefixo cíclico *short* no primeiro e nos demais símbolos.

BW (MHz)	CP em (μs)	CP (em amostras de símbolo OFDMA)
1,4	(5, 2/4, 7)	(10/18)
3	(5, 2/4, 7)	(20/18)
5	(5, 2/4, 7)	(40/36)
10	(5, 2/4, 7)	(80/72)
15	(5, 2/4, 7)	(120/108)
20	(5, 2/4, 7)	(160/144)

do sinal, pois a operação de FFT que é realizada na recepção não será avariada. A remoção do prefixo cíclico é feita com o auxílio do *toolbox* de LTE do *framework* Matlab. Primeiramente, as amostras I/Q são moduladas para OFDMA e então o prefixo é retirado das amostras de OFDMA e o sinal de-modulado e, convertido em amostras de LTE em banda-base. Desta forma, ao removê-lo o ganho de compressão é de dado pela Equação 2.1.

$$C_{cp} = \left(\frac{L_{sym} + L_{cp}}{L_{sym}} \right) \quad (2.1)$$

Onde L_{sym} e L_{cp} representam, respectivamente, o tamanho do símbolo de OFDMA e o tamanho do prefixo cíclico.

2.1.2 Sobre-amostragem

A sobre-amostragem é uma redundância no domínio da frequência que ocorre nos sinais de LTE em banda base que são amostrados nas BBUs em uma taxa superior a frequência de Nyquist [9], [26] e [14].

A unidade fundamental de um sinal de LTE é o RE (RE, *Resource Element*, elemento de recurso). Cada RE transmite uma modulação, que pode ser *Phase Shift Keying* (QPSK) ou *Quadrature Amplitude Modulation* (QAM), através de uma *subcarrier* que possui 15 kHz.

O conjunto de 12 *subcarriers* forma um símbolo multiplexado em OFDMA. Um símbolo OFDMA contém 12 REs. Um *Resource Block* (RB) contém 7 símbolos OFDMA com prefixo

cíclico normal. Na Figura 2.3, retirada de [20], observa-se, à esquerda, um *slot* de LTE e, à direita, a visualização de um RB deste mesmo *slot*.

Desta forma no domínio do tempo um RB de LTE possui a duração de $0,5\text{ ms}$ e, no domínio na frequência um RB ocupa 180 kHz . Os sinais de LTE na banda-base podem conter diversos números de RBs onde se encontram a informação útil do sinal.

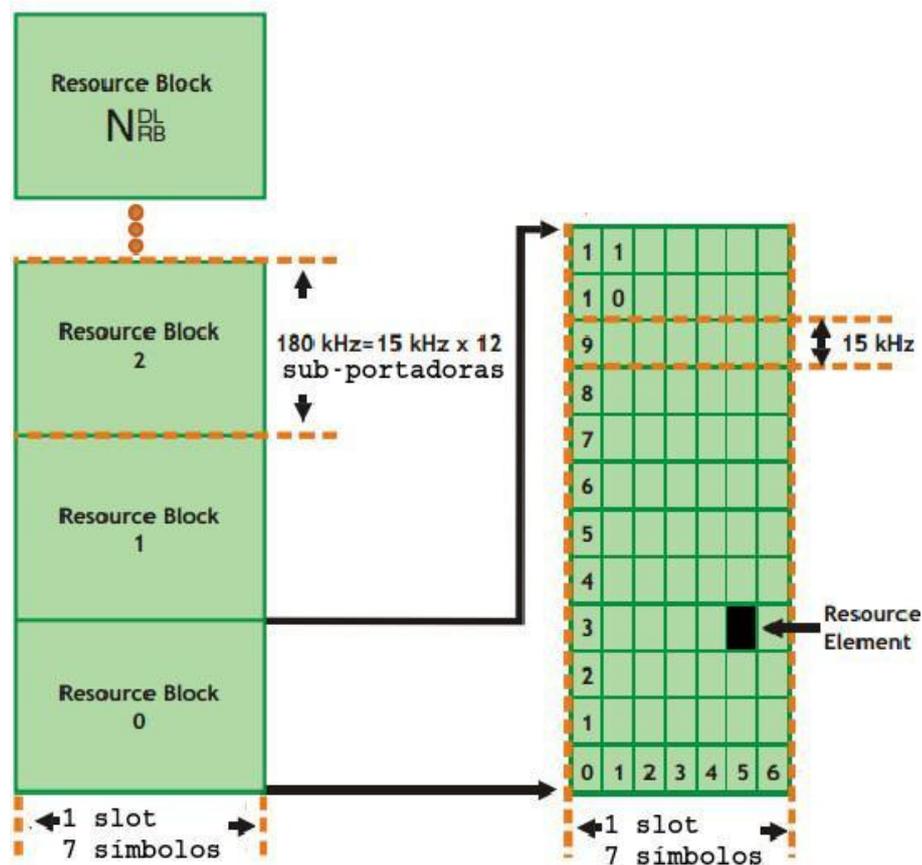


Figura 2.3: Um *slot* de LTE, retirada de [6].

No entanto, devido à operação de IFFT presente nos sinais de OFDMA, uma maior frequência de amostragem é necessária para transmitir estes sinais, a fim de que a IFFT possa ser desfeita no receptor e os símbolos demodulados.

Por exemplo o sinal de LTE de Banda $BW = 20\text{ MHz}$, possui 100 RBs e banda útil dada pela Equação 2.2:

$$BW_{\text{útil}} = 100(RB) \times 12 \left(\frac{\text{subcarrier}}{RB} \right) \times 15.10^3 \left(\frac{Hz}{\text{subcarrier}} \right) = 18MHz \quad (2.2)$$

Tabela 2.2: Características dos sinais de LTE , obtida em [15].

BW (MHz)	3	5	10	15	20
Duração do frame (ms)	10	10	10	10	10
Duração do sub-frame (ms)	1	1	1	1	1
Largura das sub-portadoras (kHz)	15	15	15	15	15
Tamanho da FFT	256	512	1024	1536	2048
Sub-portadoras ocupadas	151	301	601	901	1201
Sub-portadoras de guarda	105	211	423	635	847
Número de <i>resource blocks</i>	12	25	50	75	100
Banda ocupada	2,265	4,515	9,015	13,515	18,015
Eficiência espectral (%)	90%	90%	90%	90%	90%

Logo, de acordo com o Teorema de *Nyquist*, este sinal poderia ser amostrado com no mínimo $F_s = 18 \text{ MHz}$. No entanto ele é amostrado com $F_s = 30,72 \text{ MHz}$, pois possui 1200 sub-portadoras porém precisa transmitir com 2048 sub-portadoras, ou seja, com $(2048 \times 15 \text{ kHz} = 30,72 \text{ MHz})$. Logo, o sinal de LTE, de banda de 20 MHz possui uma redundância espectral de cerca de $\left(\frac{2048}{1200}\right) = 1,70$.

Levando em conta a largura de banda de 2 MHz , reservada para a banda de guarda do sinal, há cerca de 10 MHz de redundância de largura de banda no domínio espectral [9].

A Tabela 2.2 retirada de [27], exhibe as bandas dos sinais de LTE, as bandas úteis e as frequências em que esses sinais são amostrados.

A Figura 2.4, retirada de [24], apresenta um sinal LTE de 15 RBs, sendo a banda útil representada pela $BW_{config} = 2,7 \text{ MHz}$ e a banda total do sistema é representada pela $BW_{channel} = 3 \text{ MHz}$, os *active resource blocks* referem-se ao número de *resource blocks* ativos na atual transmissão, que pode ser no máximo 15 para o caso de sinais de LTE com $BW_{channel} = 3 \text{ MHz}$.

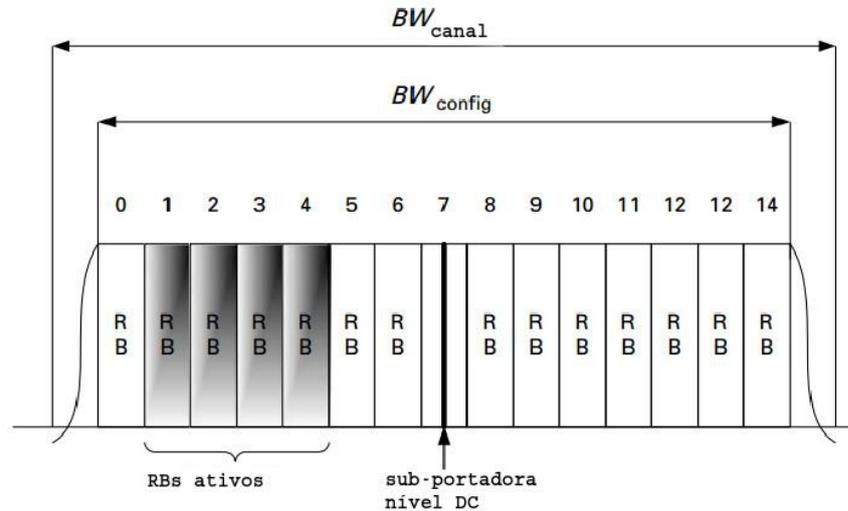


Figura 2.4: Banda útil transmitida em um sinal de LTE, retirada de [11].

Técnicas de processamento de sinais podem ser utilizadas para retirar as redundâncias do domínio espectral como proposto no Capítulo 3.

Este passo visa reduzir a frequência de amostragem do sinal, removendo assim as redundâncias no domínio da frequência. Tal etapa funciona como descrito a seguir.

Primeiramente, as amostras I/Q sem prefixo cíclico são sobre-amostradas por um fator de K proposto em [5] e [11]. Então, a sua frequência de amostragem inicial é multiplicada K vezes. Implicando no aparecimento de réplicas do sinal inicial no domínio da frequência dificultando a sua posterior reconstrução. O sinal inicial pode ser visto na Figura 2.5.

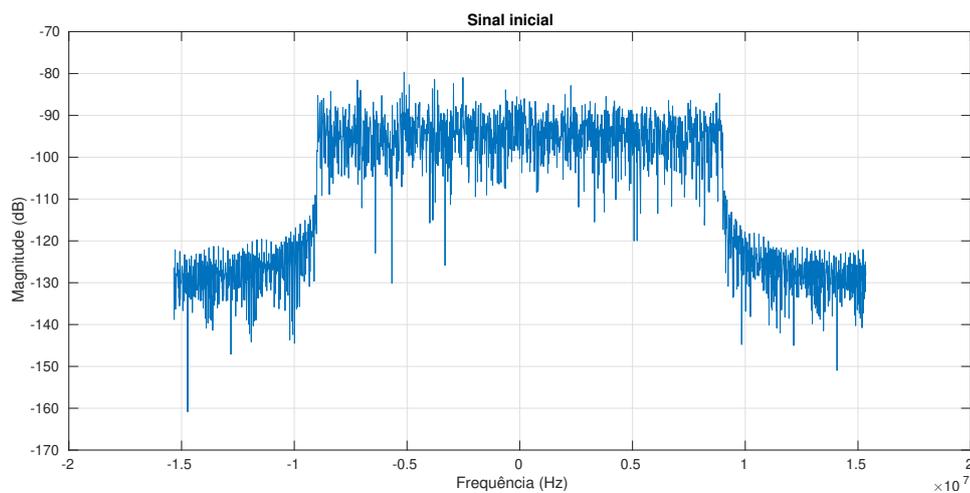


Figura 2.5: Sinal no domínio da frequência antes do processo de sub-amostragem.

Após isto, este sinal com réplicas é passado através de um filtro passa-baixa, implemen-

tado com o auxílio da ferramenta *fdatool* do *framework* Matlab e implementa uma janela de *Kaiser* proposta em [9].

A frequência de amostragem do filtro deve ser de $f_{filtro} = (K \times f_s)$, pois o sinal a ser filtrado possui frequência de amostragem de $(K \times f_s)$ e as bandas de passagem e corte são de respectivamente 9 MHz e $11,85\text{ MHz}$, como mostrado nas Figuras 2.6 e 2.7 que são modelos de filtro criados para os casos de sobre-amostragem com $K = 2$ e $K = 5$.

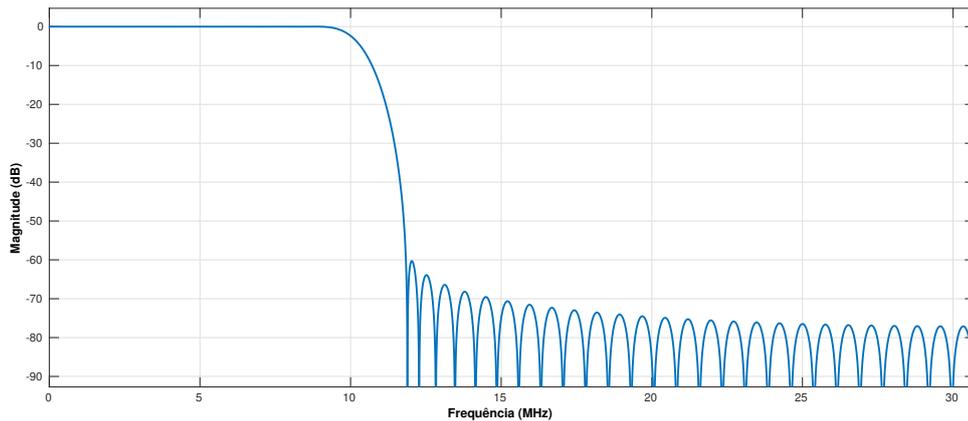


Figura 2.6: Filtro utilizado no processo de sub-amostragem quando $K = 2$.

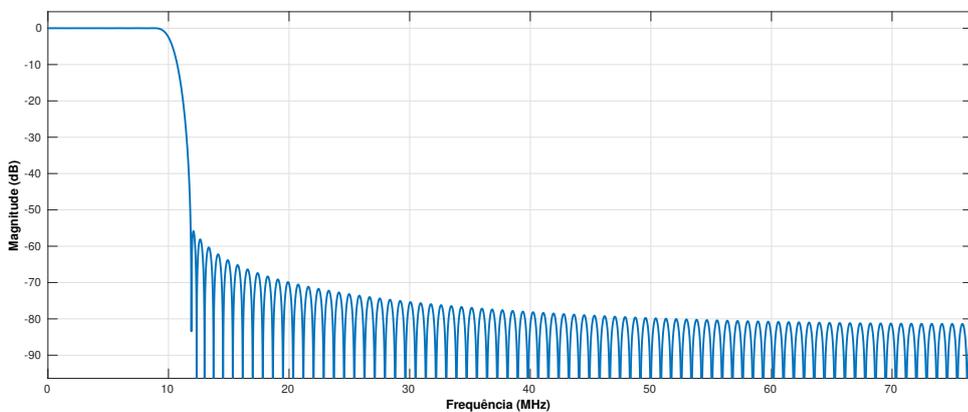


Figura 2.7: Filtro utilizado no processo de sub-amostragem quando $K = 5$.

Filtros similares aos das Figuras 2.6 e 2.7 foram desenvolvidos neste trabalho, no entanto o grau de complexidade (número de coeficientes do filtro FIR) dos filtros aumentou consideravelmente, conforme o aumento de K , para que o sinal filtrado não perdesse a sua ortogonalidade como discutido na subseção 2.1.1.

O sinal resultante é sub-amostrado de um fator de L segundo [9], [5] e [11]. O sinal resultante é mostrado na Figura 2.8.

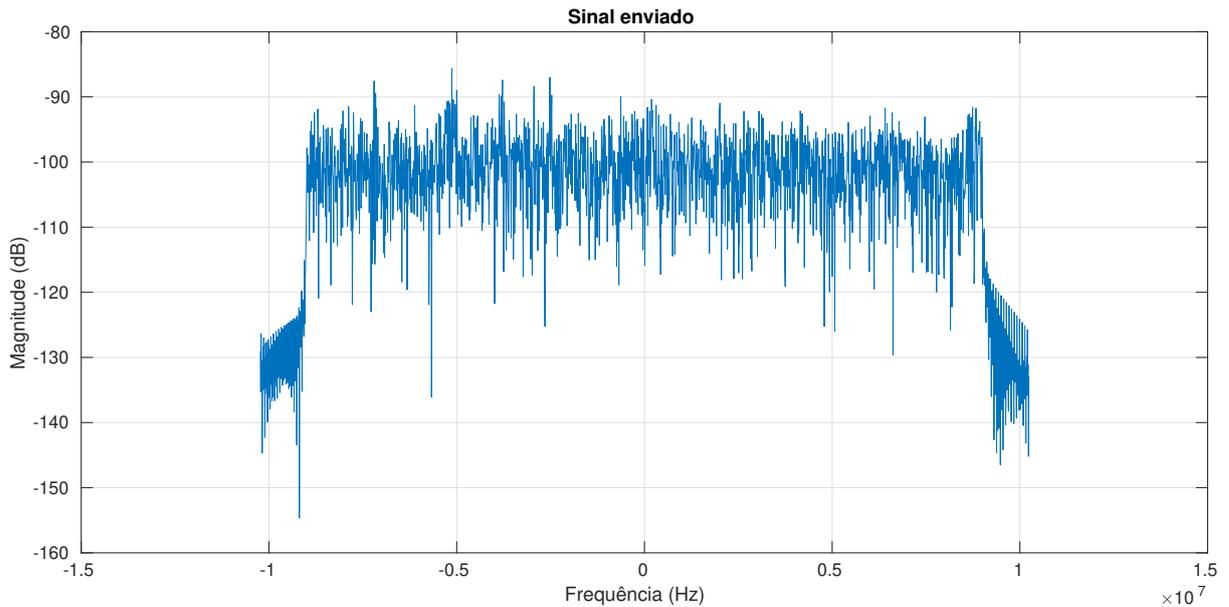


Figura 2.8: Sinal após o processo de sub-amostragem.

O ganho de compressão em relação ao sinal anterior é dado pela Equação 2.3.

$$C_{down} = \left(\frac{L}{K}\right) \quad (2.3)$$

Na Figura 2.7 é mostrado um filtro desenvolvido para $K = 5$, pode-se notar que as bandas de passagem e de corte são idênticas as da Figura 2.6. No entanto o filtro difere nas suas atenuações e na frequência de amostragem que é proporcional ao fator de sobre-amostragem, neste caso $K = 5$.

Como os sinais de LTE possuem uma redundâncias de aproximadamente $1,70 \times F_{nyquist}$, a compressão de uma taxa de 1,6 uma distorção muito pequena, como mostrado no Capítulo 4

2.1.3 Normalização em blocos

Um dos principais inconvenientes dos sinais de LTE é que o sinal OFDMA do qual eles são constituídos possui elevados picos de potência em relação potência a média (PAPR, Peak-to-Average Power Ratio) [21].

No caso geral, o transmissor OFDMA pode ser visto como uma transformada linear realizada através de um grande bloco de símbolos independentes modulados e identicamente distribuídos (no domínio da frequência). De acordo com o teorema do limite central [28] e [29],

o símbolo OFDMA no domínio do tempo pode ser aproximado como uma forma de onda gaussiana.

As variações de amplitude do sinal modulado OFDMA podem, portanto, ser muito altas. No entanto, os amplificadores de potência práticos dos transmissores de RF são lineares apenas dentro de uma faixa dinâmica limitada. Assim, sinais de OFDMA são susceptíveis à distorções não-lineares causadas pela saturação do transmissor [21].

Esta função tem por objetivo diminuir a resolução do sinal e manter a faixa dinâmica controlando assim o limite dinâmico dos sinais complexos. Desta forma, melhorando-se a distribuição dos níveis do sinal para que amplitudes pequenas não sejam interpretadas como ruído e amplitudes grandes não sejam saturadas e preparando o sinal para o processo de quantização [9].

A normalização em blocos permite ainda que o sinal produzido seja consistente com o processo de quantização vetorial que ocorrera na próxima etapa [9], [5] e [11]. Assim, esta segmenta o sinal em blocos tamanho $N_b = 32$ amostras, como proposto em [9] e em seguida, normaliza cada bloco pelo seu valor máximo assim os sinais são normalizados.

Esta etapa produz como resíduo, um bloco extra B , necessário para transmitir os valores máximos dos B blocos criados. Este bloco é quantizado com Q_a bits.

2.1.4 Quantização Vetorial

Os métodos de VQ compreendem uma classe de algoritmos que tem por objetivo reduzir o número de níveis disponíveis em um conjunto de pontos. Tais estratégias consideravelmente úteis em sistemas de comunicação digital, nos quais, geralmente, o sinal de entrada já possui alguma representação finita. A entrada de um sistema de VQ são as entradas finitas e a saída é a sua versão comprimida, ou seja com menos níveis disponíveis [7].

Desta forma, este bloco recebe como entrada os dicionários treinados com algoritmos computacionais. Para cada amostra I/Q de um sinal complexo de LTE, é feita uma pesquisa de qual ponto do dicionário encontra-se mais próximo e, após isto, este ponto é codificado para o ponto mais próximo dentro do dicionário treinado. Reduzindo assim o número de níveis disponíveis com uma distorção mínima possível.

Os sinais de LTE são amostrados e quantizados para o *fronthaul* CPRI utilizando-se geralmente 30 bits (15 são para cada amostra I e Q). Isto significa que os sinais complexos são representados por até 2^{30} níveis possíveis [30].

O uso de técnicas computacionais como, por exemplo VQ ou ainda clusterização, aplicada à VQ, pode reduzir o número de níveis bi-dimensionais disponíveis e, conseqüentemente, o número de bits necessários para representar um sinal de LTE.

No entanto, ao reduzir o número de níveis disponíveis, o erro de quantização é criado, como mostrado na Figura 2.9, na qual se pode ver um sinal de LTE normalizado, em vermelho, e quantizado com 16 níveis ou 4 bits, representados em pontos azuis. Esta quantização exibe erros de quantização consideráveis ao quantizar os sinais de uma região como por exemplo a região azul claro para o seu centroide correspondente, devido á pequena quantidade de centroides.

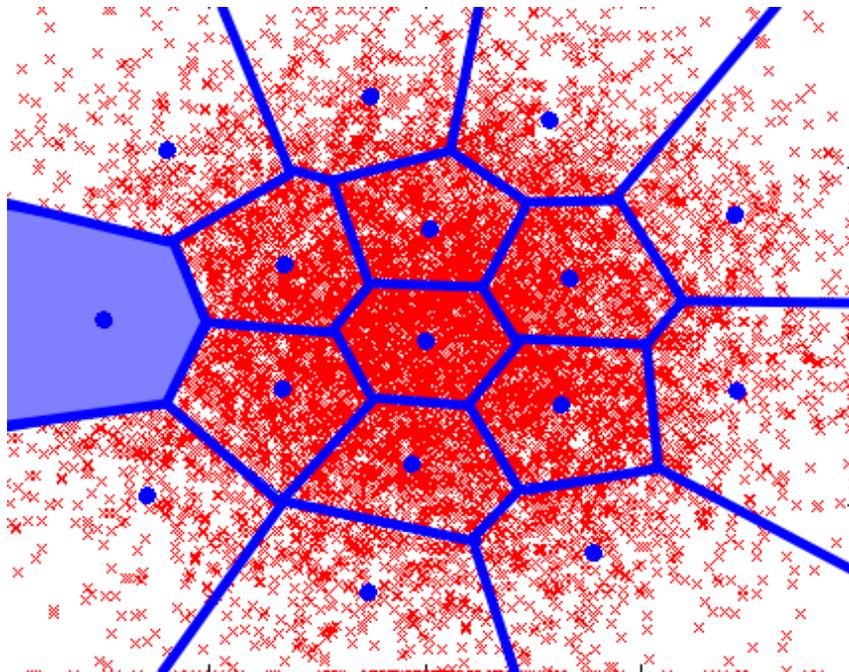


Figura 2.9: Esquema de quantização para um sinal de LTE normalizado.

Assim o estudo de técnicas computacionais é fundamental para reduzir os erro de quantização, tornando viável a compressão dos sinais à medida que busca reduzir os erros de quantização como mostrado nos capítulos seguintes.

As amostras I/Q encontram-se normalizadas, no entanto, infinitos níveis entre $(-1, -1)$ e $(1, 1)$ são possíveis. Assim, esta etapa quantiza um sinal normalizado e avalia o erro neste processo. Ademais é assumido que o sinal de referência foi quantizado com 15 *bits* como se fosse transmitido no *fronthaul* CPRI.

Desta forma o efeito de quantizar vetorialmente um sinal em um número de bits menor

do que o usual, tem o fator de compressão dado pela Equação 2.4:

$$C_{vq} = \left(\frac{Q_0}{Q_{vq}} \right) \quad (2.4)$$

Onde Q_0 é a quantidade de bits por amostra I/Q do sinal inicial do (assumida como 15), e Q_{vq} é a quantidade de bits após a quantização vetorial.

Este processo proporciona uma compressão, não somente no número de níveis, mas também no número de bits disponíveis para a transmissão, como mostrado na Figura 2.10

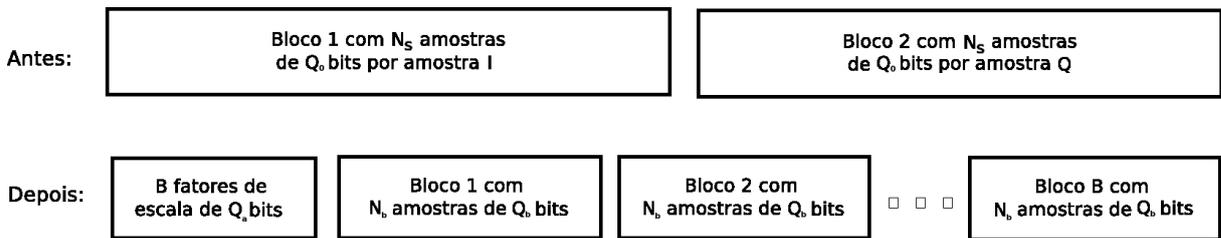


Figura 2.10: Compressão obtida na taxa de bits.

No qual, N_s representa o número inicial de amostras e as variáveis B , N_b e Q_b representam, respectivamente, o número de amostras por bloco, número de amostras por bloco e o número de bits utilizados para quantizar vetorialmente as amostras. Como mostrado na Equação 2.5.

$$\begin{aligned} C_{vq} &= \left(\frac{Q_0}{Q_{vq}} \right) \\ C_{vq} &= \left(\frac{2 \times N_s \times Q_0}{(N_s \times Q_b) + (B \times Q_a)} \right), \text{ como } N_s = (B \times N_b) \\ C_{vq} &= \left(\frac{2 \times Q_0}{Q_b + \left(\frac{Q_a}{N_b} \right)} \right) \end{aligned} \quad (2.5)$$

2.1.5 Codificação por entropia

Codificação por Entropia é um método de compressão sem perdas. A qual tem por objetivo reduzir o número total de bits necessários para transmitir a informação. Esta técnica cria palavras de maior tamanho binário para *codewords* que possuem uma menor frequência (maior entropia) e palavras binárias menores para *codewords* que possuem maior ocorrência (menor entropia). Diminuindo o número de bits necessários para transmitir a informação total do sinal [5].

A técnica de codificação mais utilizada é o Código de Huffman (*Huffman Coding*). No

código de *Huffman* a construção de tabelas de associação binária é feita em um tempo linear através da criação de árvores de *Huffman* ou *splitting trees* [31].

O Teorema da Codificação de *Shannon* estabelece que o comprimento médio ótimo (mínimo) de um código para uma fonte é igual ao valor da sua entropia. Os códigos *Huffman* provaram ser ótimos e de complexidade linear para codificação de cada dicionário [32].

Cada dicionário requer L_{vq} bits, logo a informação total requer exatamente $L_{vq} \times Q_{vq}$ bits para a sua representação. No entanto, após a codificação por entropia, alguns dicionários serão representados por palavras binárias maiores e outros palavras binárias menores, dando origem a um comprimento médio da palavra binária, denotado por L_{huff} descrito pela Equação 2.6 [5].

$$L_{huff} = \sum_{i=1}^n (p_i \times L_i) \quad (2.6)$$

Onde p_i e L_i denotam respectivamente as probabilidades dos dicionários de índice i e os tamanhos das palavras binárias utilizadas para representar cada um dos i dicionários.

Observa-se nas Figuras 4.12 e 4.13 que a ocorrência dos dicionários não é uniforme, o que torna vantajosa a aplicação do Código de *Huffman* sobre estes que foram treinados no processo de VQ [5].

O ganho de compressão que a codificação de entropia possibilita é expresso pela Equação 2.7.

$$C_{ent} = N_s \times \frac{L_{vq}}{L_{huff}} \quad (2.7)$$

No qual as variáveis C_{ent} , N_s , L_{vq} e L_{huff} denotam respectivamente o ganho obtido no processo de codificação por entropia, o número de amostras, o comprimento da palavra binária após o VQ, o comprimento da palavra binária e o comprimento médio da palavra obtido na codificação de *Huffman*.

2.1.6 Compressão total

Decorrido o processo de compressão, temos que a compressão imposta no processo é dada pela Equação 2.8.

$$C_{total} = \left[\frac{\left(\left(\frac{L_{sym} + C_{cp}}{L_{sym}} \right) \times \left(\frac{L}{R} \right) \times (2 \times L_{ini}) \right)}{\left(L_{huff} + \left(\frac{Q_a}{N_b} \right) \right)} \right] \quad (2.8)$$

2.2 Visão geral do *fronthaul* CPRI das redes CRAN

A arquitetura CRAN propõe que o processamento seja centralizado nas BBUs, separado das RRHs. Neste sentido, é criado então o *fronthaul* com o intuito de fazer a ligação entre a BBU e as RRHs. Esta interface é feita utilizando-se o protocolo CPRI (CPRI, *Common Public Radio Interface*).

Na Figura 2.11, retirada de [30], temos uma rede CRAN implementada com *fronthaul* CPRI. As funções de redes, como por exemplo camada MAC e codificação de canal encontram-se virtualizadas nas BBUs. Nos *links* de fibra, são transportados dados dos usuários e dados de controle. Na RRH, o sinal é recuperado, modulado e transmitido ao usuário.

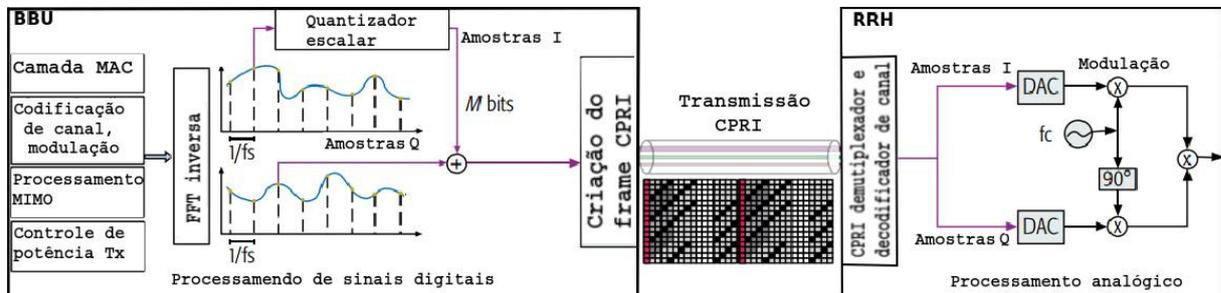


Figura 2.11: CRAN utilizando *fronthaul* CPRI, retirada de [18].

O CPRI foi definido em um acordo na indústria entre várias empresas de telecomunicações com o intuito de definir uma especificação padrão para a comunicação entre BBUs e RRHs de diversas empresas de modo a unificar os padrões impostos implementação *fronthaul* [33].

2.3 Características dos sinais no *Fronthaul* CPRI

A especificação CPRI define o uso de amostragem e quantização escalar (SQ, *Scalar Quantization*) no processo de digitalização dos sinais de banda base. Os sinais de banda base são oriundos da BBU no *downlink* ou oriundos do processo de *downconversion* dos sinais na RRH [4].

O padrão CPRI define ainda a estrutura de *frames* projetada para transportar dados do plano do usuário, as amostras I/Q quantizadas, juntamente com o plano de controle, como mostrado na Figura 2.12 retirada de [12], na qual se observa a composição de um *frame* básico de CPRI, que possui 16 *bytes*, onde um é destinado ao funções de controle e 15 são destinados aos dados do usuário e 3,84 MHz é a taxa de transmissão de um *chip* CPRI [12].

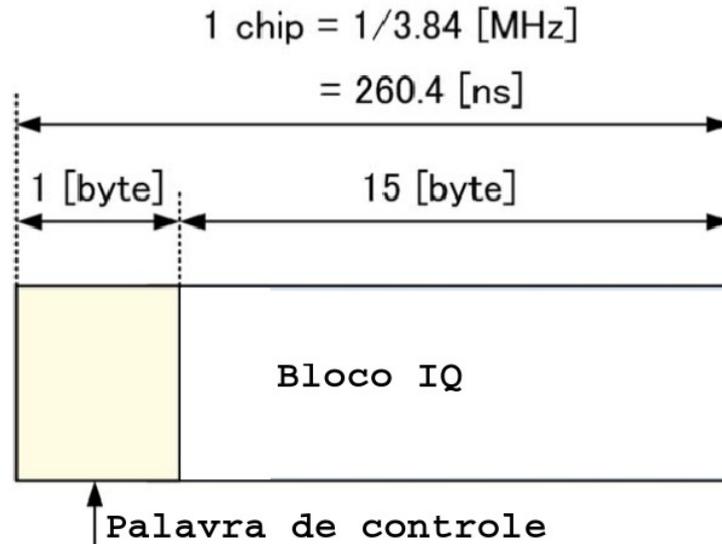


Figura 2.12: *Frame* básico de CPRI, retirada de [20].

O padrão CPRI suporta as tecnologias 3GPP GSM/EDGE, 3GPP, UTRA e LTE e permite as topologias estrela, cadeia, árvore, anel e multi-hop. As *line rates* são proporcionais à largura de banda do sinal a ser digitalizado, ao número de antenas no RRH e ao número de bits por amostra. Especificamente, a taxa de bits pode ser calculada pela Equação 2.9 retirada de [4].

$$R_{CPRI} = 2N_{ant} \times F_s \times Q_0 \times N_{ov} \quad (2.9)$$

Onde N_{ant} representa o número de antenas na RRH, F_s é a taxa de amostragem no CPRI que depende na largura de banda do sinal de acordo com uma Tabela 2.2, Q_0 é o número de bits por amostra complexa I ou Q, que varia no intervalo de 8 a 20 bits para LTE no *uplink* e no *downlink* e N_{ov} é um parâmetro que leva em conta os bytes extras necessários para transmitir o plano de controle e o plano de sincronização [4].

Considerando-se por exemplo um sinal LTE de 20 MHz, ele possui a frequência de amostragem de 30,72 MHz, assumindo 30 bits por amostra CPRI, a taxa de dados seria 0,9216 Gbps dados por antena da RRH [30]. Esta taxa pode aumentar consideravelmente com a utilização de técnicas como *carrier aggregation* ou *massive MIMO*, criando assim uma necessidade de compressão.

Tabela 2.3: Metáfora de computação evolutiva, retirada de [29].

Evolução	Resolução de problemas
Ambiente	Problema
Indivíduos	Soluções candidatas
Métrica	Qualidade das soluções

2.4 Algoritmos Genéticos

A computação evolutiva é uma área de pesquisa da Ciência da computação que estuda e desenvolve os Algoritmos Evolutivos (EAs), estes são procedimentos de busca de propósito geral, que são baseados em um poderoso princípio de evolução que promove a sobrevivência do mais indivíduo mais apto. Estes procedimentos modelam alguns fenômenos naturais como herança genética e luta darwiniana pela sobrevivência, constituem uma categoria interessante algoritmos de pesquisa heurística moderna e são atraentes devido à sua simplicidade, facilidade de interface e extensibilidade com aplicações de diversos propósitos [34].

A tabela 2.3 retirada de [35] exemplifica como ocorrem a codificação dos problemas de otimização para a teoria da evolução.

Os AGs são algoritmos evolutivos estocásticos de busca e otimização de problemas baseados na população e inspirados no processo de seleção natural e genética. Uma das principais características dos AGs é que eles trabalham com população ou conjunto de soluções candidatas, ao contrário de outras abordagens clássicas que operam em uma única solução de cada vez. Desta forma eles exploram diferentes regiões do espaço de pesquisa simultaneamente, exibindo assim um desempenho aprimorado [36].

Qualquer algoritmo Evolutivo pode ser descrito entorno das suas variáveis principais que são [37]:

1. Representação do indivíduo.
2. Função de avaliação.
3. População.
4. Seleção de parentes.

5. Operadores reprodutivos.

6. Mecanismo de seleção.

2.4.1 Codificação do indivíduo

No contexto dos Algoritmos Genéticos o indivíduo é utilizado para denotar soluções possíveis. O espaço de todas as possíveis soluções candidatas é comumente chamado de espaço fenotípico [37]. A codificação do indivíduo é o processo de criar um genótipo, ou seja, criar uma representação biológica (genes) para um problema real. Os genes são as unidades fundamentais de herança, eles contêm todas as informações necessárias para construir o fenótipo (característica) específica de um indivíduo. Por exemplo, propriedades visíveis como a cor da pele ou o comprimento da cauda podem ser determinados pelos genes [37]. A codificação do indivíduo pode ser feita de dois modos segundo [38]:

1. Indivíduo Binário.

2. Indivíduo Real.

Problemas em que os valores das variáveis são contínuos e que possuem infinitas representações são melhor representados por indivíduos com genes reais, desta forma a precisão total da máquina poderá ser utilizada [38].

Quando o número de variáveis é grande, o número de genes binários aumenta consideravelmente. Quando as variáveis são naturalmente quantizadas a um valor fixo de níveis possíveis, o AG binário se encaixa bem. No entanto, quando as variáveis são valores reais, é mais lógico para representá-los por números de ponto flutuante facilitando assim representação da precisão da máquina [38].

2.4.2 Geração da população inicial

Um AG é um EA e estocástico, que opera sobre um conjunto de indivíduos (população) de forma determinística. O tamanho da população, o número de descendentes, o tipo de seleção e de sobrevivência definem o tipo de EA, como mostrado na Tabela 2.4 retirada do livro [35], onde m representa o número de indivíduos na população e n representa o número de descendentes criados a cada geração [35].

Tabela 2.4: Categorias de EA, retirada de [29].

EA	m	n	Seleção de pais	Seleção de sobreviventes
EA	< 20	$n = m$	Determinística	Determinística
EP	< 10	$n \geq m$	Estocástica	Determinística
GA	> 20	$n = m$	Estocástica	Determinística

Cada indivíduo representa um potencial solução para o problema a ser resolvido. Estas soluções são obtidas por meio de um mecanismo de codificação/decodificação que codifica os indivíduos em uma solução para o problema em questão. A população pode ser gerada aleatoriamente ou com a ajuda de uma heurística compatível com o problema em questão [39].

2.4.3 Função de Aptidão

A função de aptidão é uma função que atribui a cada um dos indivíduos uma medida de sua qualidade ou *fitness* com respeito para o problema em consideração [39].

Este valor é uma informação quantitativa dos indivíduos que é determinada pelo ambiente (problema em questão). O *fitness* relaciona o quão útil o indivíduo é para o problema em questão. Em outras palavras ele expressa as chances de sobrevivência e multiplicação que cada um dos indivíduos possuirá [37].

Desta forma o *fitness* é utilizado para mensurar a qualidade das soluções obtidas. Sendo utilizado para guiar o algoritmo de otimização na busca. Onde os melhores indivíduos serão mantidos e utilizados como sementes na tentativa de constituem melhores indivíduos em gerações posteriores, otimizando assim a função em questão [37].

2.4.4 Avaliação e Otimização Local

Na maioria das vezes, o AG procura a melhor solução dentre um conjunto específico de soluções (indivíduos). O espaço de todas as soluções viáveis (o conjunto de soluções entre as quais a solução desejada reside) é chamado de espaço de busca (também espaço de estados). Cada um e cada ponto no espaço de pesquisa representa uma solução possível [39].

A etapa de otimização possui como objetivo entregar ao AG um conjunto de indivíduos ou soluções viáveis utilizando algoritmos de pesquisa convenientes com o problema abordado.

Estes algoritmos de pesquisa encontram-se descritos no Capítulo 3. Na etapa de avaliação os *fitness* dos indivíduos são calculados e retornados ao AG.

2.4.4.1 Algoritmo LBG

O algoritmo LBG (Y. Linde, A. Buzo e R. Gray) ou GLA (*Generalized Lloyd Algorithm*) é um algoritmo que visa encontrar conjuntos de pontos uniformemente espaçados em subconjuntos dos espaços Euclidianos [19].

Este algoritmo pode ser inicializado de duas formas [40]:

- Aleatória na qual um dicionário de tamanho M *codewords* é escolhido de forma aleatória entre pontos pertencentes ao conjunto de treino.
- Por particionamento, inicialmente, é escolhido aleatoriamente uma *codeword* y do conjunto de treino e esta é particionada 2 *codewords* ao seu redor, utilizando para isto uma variável de perturbação ε . Este procedimento se repete até que seja alcançado o número M , onde M é uma potência de 2.

Decorrida a inicialização, o algoritmo designa no conjunto de treinamento os pontos mais próximos de cada *codeword*, definindo assim as regiões de *Voronoi* ou células. E então atualiza cada *codeword* pelo centro do *cluster* [7].

O processo se repete até que um critério de parada seja alcançado, que pode ser minimizar o erro de quantização, utilizando, um parâmetro de convergência ou do número de iterações [41].

O algoritmo desenvolvido, neste trabalho utiliza a ferramenta *vqdtol* do Matlab. Ele emprega a inicialização aleatória e tem como critério de parada o número de 30 iterações. Visto que, este foi um padrão de convergência estipulado baseado em testes efetuados com todos os tamanhos de dicionários utilizados.

2.4.4.2 Algoritmo K-means

O K-means é um algoritmo de clusterização que particiona um conjunto de dados em k *clusters*, onde k é definido pelo usuário.

O *software* proposto no trabalho faz o uso de um modelo de Kmeans implementado na plataforma de desenvolvimento Matlab, denominado Kmeans++ [42] que efetua uma inicialização pseudo-aleatória para o algoritmo Kmeans na tentativa de evadir de mínimos locais.

Primeiramente é definido $D(x)$ que denota a distância mais curta de um ponto pertencente ao conjunto de treino $x \in X$ e o centro mais próximo deste, o primeiro centro c_1 é escolhido uniformemente ao acaso do conjunto de treino X . O próximo centro c_i , será escolhido do conjunto de treino X com uma probabilidade de sorteio de $p(x_i)$ dado pela Equação 2.10:

$$p(x_i) = \left(\frac{D(x_i, C)^2}{\sum_{x \in X} D(x, C)^2} \right) \quad (2.10)$$

onde $d(x_i, C)^2$ representa a distância de um ponto x_i ao centro mais próximo e $\sum D(x, C)^2$ representa a soma das distâncias de todos os pontos a cada um dos seus centros mais próximos [18].

Após a inicialização o algoritmo seleciona a qual *clusters* pertence a cada um dos pontos do conjunto de treino e recalcula a nova posição dos *clusters* visando minimizar o quadrado da distância euclidiana total entre cada ponto x' e seu *cluster* mais próximo c' . O processo é repetido até que algum critério de convergência seja atingido [43].

O *software* desenvolvido criado utiliza como critério de parada o número de 30 iterações.

Como os algoritmos LBG e Kmeans++ possuem inicialização aleatória o apresenta-se como AG como uma alternativa de método de pesquisa a fim de escapar de um ótimo local proveniente destes algoritmos como mostrado a seguir.

2.4.5 Seleção de parentes

Seleção é o processo de escolher indivíduos da população para o cruzamento. Depois de decidir sobre uma codificação e avaliar os indivíduos, o próximo passo é decidir quais indivíduos na população que criarão descendentes para o próxima geração e quantos descendentes cada um criará.

O propósito da seleção é enfatizar indivíduos mais aptos na população na esperança de que seus descendentes possuam maior aptidão. Como o problema de treinamento de dicionários é um problema de mínimos locais, os melhores indivíduos possuirão o menor *fitness*.

O problema de seleção de indivíduos de acordo com a teoria da evolução de Darwin indica que somente os melhores sobrevivem para criar novas populações. Os principais métodos de seleção definidos no livro [39] são:

1. Seleção Roleta:

Neste operador os indivíduos são selecionados aleatoriamente com uma probabilidade proporcional à sua aptidão [39].

2. Seleção randômica:

Seleciona aleatoriamente indivíduos da população seleção de roleta [39].

3. Seleção por *rank*:

A seleção por *rank* classifica a população e cada indivíduo recebe um *ranking*. O indivíduo com pior *rank* é classificado com 1 e o melhor com N, o sorteio dos indivíduos é feito através de uma variável aleatória que selecionará o indivíduo com *rank* inferior ao valor sorteado [39].

4. Seleção por Torneio:

É uma estratégia de seleção capaz de aumentar desempenho de busca do GA, ao propor a competição dois ou mais indivíduos escolhidos aleatoriamente [39].

5. Seleção de Boltzmann:

Na seleção de Boltzmann, uma variável que obedece a uma distribuição de probabilidade de Boltzmann controla a taxa de seleção, que ocorre de acordo com uma programação predefinida [39].

6. Amostragem estocástica:

Os indivíduos são mapeados para segmentos contíguos de uma linha, de tal forma que cada indivíduo possui um segmento de tamanho igual ao seu *fitness*. São criados ponteiros igualmente espaçados que vão definir regiões que podem possuir um ou mais indivíduos. Então uma variável uniformemente distribuída sorteia uma região e consequentemente os indivíduos pertencentes a mesma [39].

2.4.6 Cruzamento

Crossover ou recombinação é o nome dado ao processo selecionar dois indivíduos e combiná-los para gerar descendentes. Após o processo de seleção (reprodução), a população é enriquecida com melhores indivíduos. O operador de *crossover* é aplicado a todos os indivíduos com o intuito de explorar as soluções atuais e desta forma melhorar a população pela busca

de combinações melhores nos indivíduos disponíveis. Os principais métodos de cruzamento definidos no livro [39] são:

1. **Crossover de um ponto:**

No qual indivíduos possuem os genes cortados uma vez em pontos correspondentes e as seções são trocadas [39].

2. **Crossover em dois pontos:**

O *crossover* em dois pontos implementa dois cortes em pontos correspondentes em indivíduos diferentes que terão suas seções trocadas. Deve-se notar que adicionar mais pontos de cruzamento reduz o desempenho do AG [39].

3. **Crossover uniforme:**

No *crossover* uniforme cada gene filho é criado copiando o gene correspondente de um ou outro pai. Os genes são escolhidos de acordo com uma máscara binária gerada aleatoriamente com o mesmo número de genes dos pais. Onde há um 1 na máscara de cruzamento, o gene é copiado do primeiro pai, e onde há um 0 na máscara, o gene é copiado do segundo pai. Uma nova máscara de cruzamento é gerada aleatoriamente para cada par de pais. Os descendentes, portanto, contêm uma mistura de genes de cada pai [39].

4. **Crossover de três pais:**

Nesta técnica de *crossover*, três pais binários são escolhidos aleatoriamente. Cada bit do primeiro pai é comparado com o bit do segundo pai. Se ambos são os mesmos, o bit da mesma posição no terceiro indivíduo é modificado [39].

2.4.7 Mutação

Mutação é o processo em que alterar o valor de cada gene é variado aleatoriamente dentro de seu domínio. A mutação impede que o algoritmo fique preso em um mínimo local. A mutação é um operador de pesquisa, que busca manter a diversidade genética na população, ao introduzir novas estruturas genéticas igualmente promissoras dentro do espaço de busca [39].

Mutações aleatórias podem introduzir traços não pertencentes à população original e, desta forma, evitando que ocorra uma convergência prematura no AG. O incremento no número

de mutações aumenta a liberdade do algoritmo de pesquisar novas soluções dentro do atual espaço de busca variável. Este procedimento tende a distrair o AG de convergir em uma solução popular [38].

O livro [38] descreve a mutação gaussiana que é uma forma de mutação aplicada a indivíduos com genes reais.

1. **Mutação Gaussiana:**

Na mutação gaussiana os genes dos indivíduos são somados a uma variável aleatória de distribuição normal, com desvio padrão δ , média 0 e variância 1.

No entanto, técnica deve ser utilizada com cuidado, pois a adição de um número aleatório δ pode estender variável além dos limites do espaço de busca, aumentando assim o tempo de processamento do computador [38].

2.4.8 Seleção de sobreviventes

O mecanismo de seleção de sobreviventes é responsável pelo processo de reduzir a memória de trabalho do AG. Nesta etapa um conjunto pais e descendentes são avaliados e alguns indivíduos são selecionados para formar a próxima geração segundo os critérios de seleção de sobreviventes. Os principais tipos de seleção de sobreviventes são [37]:

1. **Seleção por idade do indivíduo:**

Este critério considera que os indivíduos são projetados para que existam na população por um determinado número de iterações do AG. Isto não exclui a possibilidade de que descendentes altamente adaptados persistam na população, mas para que isso aconteça, eles devem ser escolhidas pelo menos uma vez na fase de seleção e depois sobreviver aos estágios de recombinação e mutação sem serem modificadas [37].

2. **Substitui o pior:**

Nesta técnica de seleção, um número definido de piores membros da população são selecionados para substituição. Embora isso possa levar a uma melhorias na aptidão média da população, também pode levar a convergência prematura como a população tende a se concentrar rapidamente no membro mais apto atualmente presente. Por esta razão, esta técnica é comumente utilizada em conjunto com grandes populações ou uma política de não repetição [37].

3. **Elitismo:**

Neste tipo de seleção os indivíduos mais aptos, são sempre mantidos na população. Desta forma um indivíduo com melhor aptidão não será descartado da população até que produza descendentes de melhor aptidão [37].

4. **Torneio de indivíduos (*Round-robin tournament*):**

O método funciona realizando competições de torneios em pares em formato *Round-robin*. Cada indivíduo é comparado em relação a outros q escolhidos aleatoriamente a partir das populações de pais e descendentes. Para cada competição, uma *score* é atribuído se o indivíduo for melhor que seu oponente. Depois de terminar todos os torneios, os indivíduos com o maior *score* são selecionados [37].

2.4.9 Critério de convergência

O critério de convergência é o fator que indica ao AG quando a terminar um experimento. Em suma, as várias condições de parada são possíveis. O livro [38] apresenta e descreve as condições de parada a seguir:

1. **Número máximo de gerações:**

O AG para após decorrido um número específico de gerações [38].

2. **Tempo decorrido:**

O AG terminará quando um tempo especificado tiver decorrido [38].

3. **Nenhuma mudança na aptidão:**

O AG terminará se não houver mudança na melhor aptidão da população para um número específico de gerações [38].

4. **Limite de tempo de parada:**

O algoritmo terminará se não houver melhoria no *fitness* durante um intervalo de tempo pré-estabelecido [38].

2.5 Conclusão

Este capítulo apresentou as principais redundâncias presentes nos sinais de LTE em banda base e discutiu os principais métodos de compressão de sinais para *fronthaul* CPRI. Foram também discutidas as principais características dos AGs, como a representação do indivíduo, função de avaliação, características da população, formas de seleção de parentes, principais operadores reprodutivos e mecanismos de seleção de sobreviventes.

Capítulo 3

Algoritmo de Compressão de Sinais de LTE

Introdução

Neste capítulo, será abordada uma explicação do algoritmo de compressão de sinais de LTE desenvolvido. O presente trabalho busca uma solução menos dispendiosa para a crescente necessidade de ampliação do *fronthaul*.

O algoritmo desenvolvido possui extrema relevância para as redes 5G, pois, ao inserir um *hardware* ou até mesmo embarcar um *software* na BBU, a necessidade de implantação de novos *links* de fibra pode ser reduzida consideravelmente, diminuindo os custos de implantação da arquitetura CRAN. Na Figura 3.1, temos um exemplo de como o algoritmo desenvolvido pode ser utilizado posteriormente para comprimir amostras I/Q e ampliar a capacidade do *fronthaul*.

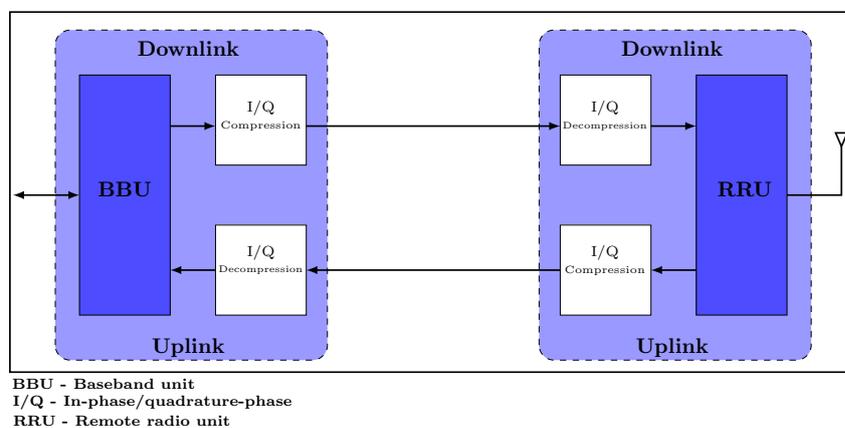


Figura 3.1: Cenário de atuação do algoritmo.

3.1 Metodologia do trabalho

O algoritmo desenvolvido foi implementada no ambiente de desenvolvimento Matlab, em duas etapas, como mostrado a seguir:

1. Algoritmo de treinamento dos *dicionários*.
2. Algoritmo de teste e Compressão de Sinais de LTE.

3.2 Treino de dicionários

O algoritmo de treinamento recebe como entrada amostras 40.000 amostras reais de sinais de LTE de 100 RBs por com a modulação QPSK dispostas em duas dimensões. Estas amostras possuem por objetivo representar um sinal de LTE de 20.000 amostras, visto que apresentam a mesma distribuição de probabilidade (Distribuição Gaussiana) que um sinal de LTE.

A Figura 3.2 demonstra que as distribuições de probabilidade das amostras reais e imaginárias de um sinal de LTE possuem a mesma distribuição de probabilidade. Logo as amostras reais podem ser utilizadas para criar um novo sinal complexo.

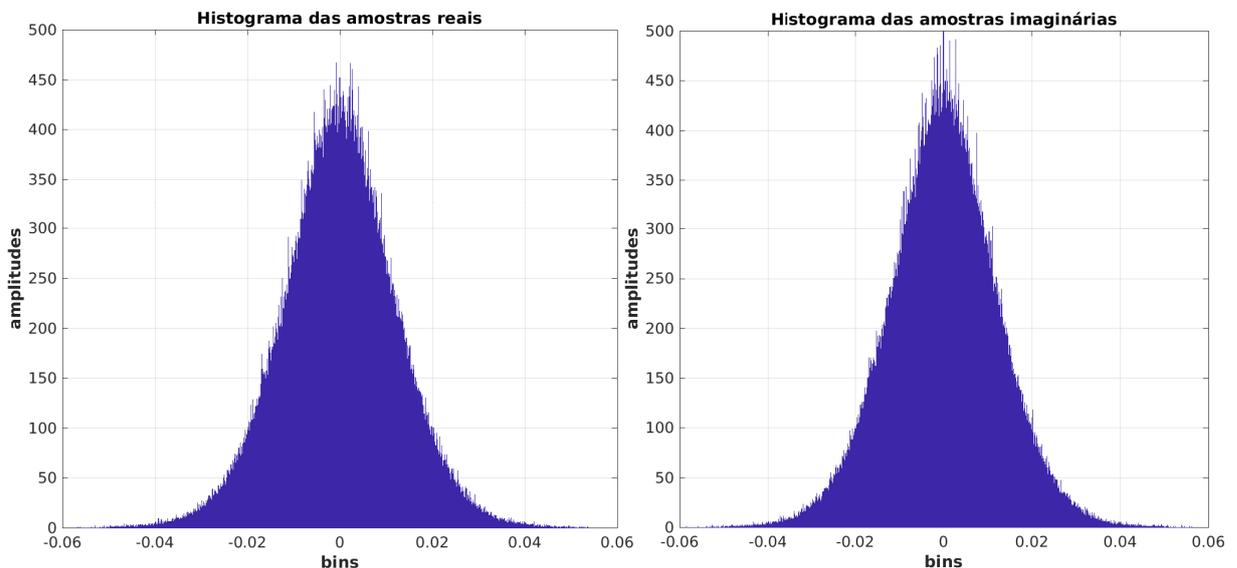


Figura 3.2: Histograma de 16384 *bins* de amostras reais e imaginárias de sinais de LTE

O treinamento de dicionários com sinais de LTE, não insere latência ao processo de compressão. Uma vez que poderá ser utilizado por quaisquer sinal de LTE com as mesmas carac-

terísticas (modulação, e frequência) visto que este possuirá a mesma distribuição de probabilidade, como apresentado na Figura 3.2.

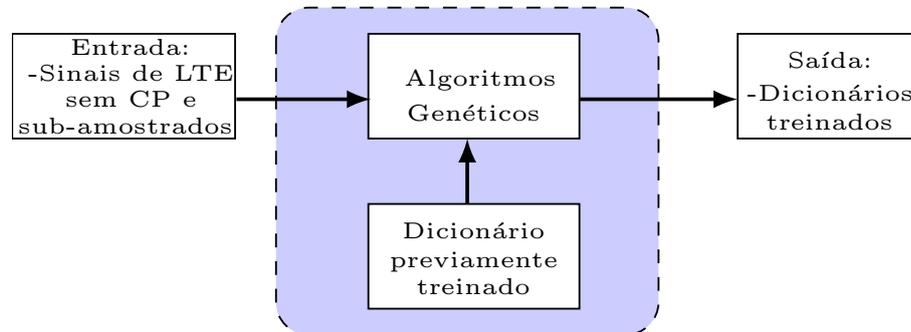


Figura 3.3: Algoritmo de treinamento resumido

O algoritmo então recebe como entradas as amostras reais de sinais de LTE, sem prefixo cíclico e sobre-amostradas e a partir destas realiza o treinamento dos dicionários utilizando os AGs, como exemplificado na Figura 3.3 e descrito na seção 3.4, até que o critério de convergência definido seja alcançado. Nas próximas seções o algoritmo de treinamento é descrito com maiores detalhes.

3.3 Algoritmo de teste e compressão de sinais de LTE

Os dicionários treinados no algoritmo anterior, são testados utilizando-se sinais I/Q de *downlink* LTE. Os sinais são produzidos com o auxílio do *toolbox* de LTE do *framework* Matlab no qual são produzidos 5 *frames* de sinais de LTE de 100 RBs por com a modulação QPSK e prefixo cíclico normal.

Os sinais são comprimidos e a EVM, descrita na subseção 3.3.2, é então calculada como métrica de avaliação da distorção imposta no processo de compressão. Desta forma a EVM será calculada entre os sinais de teste antes e depois da compressão.

O algoritmo de compressão, primeiramente, efetua a remoção do perfil cíclico, após isto, ocorre a sobre-amostragem, a normalização em blocos e, então, o VQ das amostras utilizando o dicionário treinado. Na Figura 3.4, é apresentado um esquemático que mostra as etapas do método de compressão.

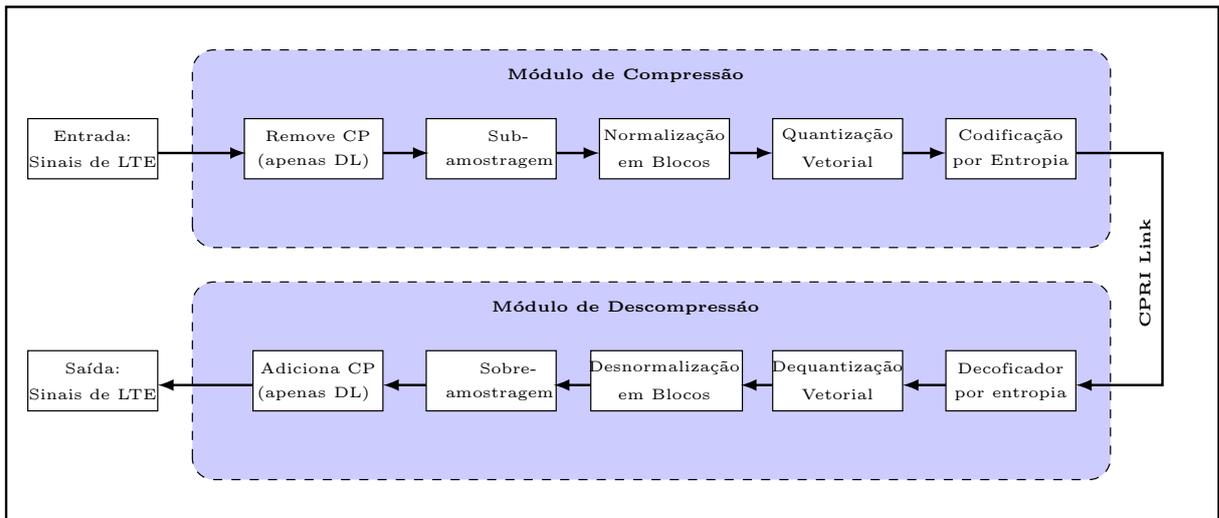


Figura 3.4: Método de compressão desenvolvido.

Após comprimir o sinal, o algoritmo retorna a taxa de compressão alcançada e a distorção imposta pelo processo total, utilizando como métrica de avaliação a EVM.

A avaliação do percentual de compressão de cada uma das etapas utilizando-se a variável C_{total} que é definida pela Equação 3.1:

$$C_{total} = \left(\frac{S_{inicial}}{S_{comprimido}} \right) \quad (3.1)$$

Onde $S_{inicial}$ representa as amostras complexas de um sinal de LTE de *downlink* e $S_{comprimido}$ representa o mesmo sinal comprimido.

3.3.1 Geração dos sinais

Nesta etapa, são criados os sinais I/Q de *downlink* LTE. São gerados 5 *frames* de sinais de LTE de 100 RBs por com a modulação QPSK e prefixo cíclico normal com o auxílio do *toolbox* de LTE da plataforma MATLAB.

3.3.2 Métrica de avaliação do Algoritmo de Compressão

O processo de compressão proposto tem o ganho de compressão total expresso pela Equação 2.8.

Tabela 3.1: EVM máxima por transmissão de LTE.

Tipo de modulação	EVM máxima
<i>QPSK</i>	17.5%
16 <i>QAM</i>	12.5%
64 <i>QAM</i>	8%
256 <i>QAM</i>	3.5%

O ruído imposto no processo é avaliado utilizando-se a métrica denominada magnitude do vetor de erro (EVM, *Error Vector Magnitude*). A EVM é uma medida da qualidade do símbolo recebido. O vetor de erro calcula a diferença entre os símbolos ideais e os símbolos recebidos equalizados. Os parâmetros de equalização são definidos em [44].

A EVM é uma porcentagem calculada a partir dos símbolos equalizados e é definida pela raiz quadrada da potência do erro médio pela potência média do sinal de referência. Como mostrado na Equação 3.2 [45].

O algoritmo avalia o erro de compressão e, conseqüentemente, a EVM é calculada entre os sinais comprimidos e não comprimidos. Portanto, não foi implementada a equalização dos sinais de LTE, somente a modulação das amostras I/Q para OFDMA.

A EVM deve ser calculada utilizando a Equação 3.2 sobre todas as larguras de banda e sobre os blocos de recursos e *subframes* dos sinais de *downlink* dentro de períodos de medição de 10 *ms*.

$$EVM(T, F) = \sqrt{\frac{\sum_{t \in T} \sum_{f \in F(T)} |S_{eq} - S_{inicial}|^2}{\sum_{t \in T} \sum_{f \in F(T)} |S_{inicial}|^2}} \quad (3.2)$$

A EVM máxima de cada portadora recebida é definido pelo 3GPP [45], como mostrado na Tabela 3.1.

Como o algoritmo de compressão implementa sinais de LTE com a modulação QPSK a EVM média é calculada sobre os sinais comprimidos em comparação aos sinais não comprimidos.

3.4 Treinamento de Dicionário por Algoritmos Genéticos

A VQ é uma generalização multi-dimensional de quantização Escalar (SQ, *Scalar, Quantization*), em que um valor real era quantizado para um valor discreto discreto. A ampliação para mais de uma dimensão proporcionou um aumento no número de aplicações, enquanto a SQ era primordialmente usada para a conversão analógico-digital, a VQ pode ser utilizada em aplicações sofisticadas de processamento de sinais, onde o sinal de entrada já encontra-se digitalizado. Como, por exemplo, para aplicações de compressão de imagens e de áudios digitais [7].

A compressão de sinais é obtida quando conseguimos criar vetores representativos a partir de uma sequência de treino. Assim, é possível atribuir diversos pontos a um *cluster* ou ponto que possa representá-lo com o mínimo de distorção. Na terminologia de VQ, os *clusters* são referidos como *codewords*, e a lista de todos os *codewords* é denominada *codebook* ou dicionário [46].

Por exemplo, a Figura 3.5, adaptada de [47], exhibe um esquema no qual os vetores de treino são utilizados para treinar um dicionário, a fim de que a função de mapeamento os codifique em seus respectivos *codevectors*.

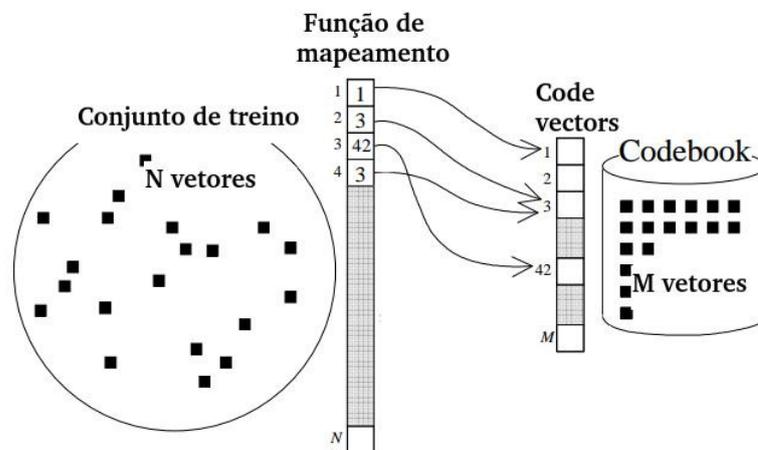


Figura 3.5: Esquema de geração de dicionários ótimos.

3.4.1 Problema de Treinamento de Dicionários

O problema de treinamento de dicionários tem por objetivo encontrar um conjunto de pontos representativos que podem ser utilizados para representar um conjunto de pontos similares aos utilizados no treinamento com uma distorção mínima.

O treinamento de dicionários ótimos, em sua forma combinatória, foi mostrado ser um

problema NP-completo [48]. Em outras palavras, não há um algoritmo polinomial capaz de encontrar um ótimo global em tempo hábil. No entanto soluções sub-ótimas podem ser obtidas são através de diversos algoritmos, como por exemplo *Fuzzy C-means* e *Kmeans* [47].

Desta forma, algoritmos de inicialização aleatória, geralmente, conduzem a mínimos locais ou dicionários sub-ótimos, neste contexto, este trabalho propõe o uso de um Algoritmo Evolutivo, o AG como forma de fazer a busca por soluções sub-ótimas, no espaço de busca de potenciais soluções já delimitadas por outros algoritmos que conduzem a ótimos locais.

Esta seção aborda o problema de treinamento de dicionários para VQ, e o AG proposto para o treinamento de dicionários. No qual, o objetivo, é reduzir a distorção média total imposta pela quantização.

Diversas heurísticas podem ser utilizadas na busca de soluções ótimas. Os Algoritmos Evolutivos (EAs) são procedimentos de busca de propósito geral, que são baseados em um poderoso princípio de evolução que promove a sobrevivência do mais indivíduo mais apto [34].

Estes procedimentos modelam alguns fenômenos naturais como herança genética e luta darwiniana pela sobrevivência, constituem uma categoria interessante de algoritmos de pesquisa heurística moderna e são atraentes devido à sua simplicidade, facilidade de interface e extensibilidade com aplicações de diversos propósitos [34].

A seguir são descritos algoritmos de geração de dicionários sub-ótimos e como foi desenvolvida o problema e a metodologia utilizada para que o AG retornasse dicionários sub-ótimos.

3.5 Algoritmo Genético proposto

3.5.1 Codificação do indivíduo

O problema de treinamento de dicionários é um problema com um número de genes finito, no entanto com infinitos valores possíveis para cada um deles. Assim ao quantizar uma variável para uma representação binária acrescentamos um erro de quantização da variável que será propagado para o erro de quantização do dicionário. Desta forma o trabalho é codifica um AG real, com genes reais, pois busca reduzir o erro no processo de quantização como mostrado na seção a seguir.

No problema de treinamento de dicionários abordado pelo trabalho cada gene representa as coordenadas (x_i, y_i) de um possível ponto do dicionário ou uma *codeword* que será utilizada para codificar todos os pontos próximos a ele. Como mostrado na Tabela 3.2 que representa

Tabela 3.2: Gene desenvolvido no trabalho.

x_i
y_i

Tabela 3.3: Um indivíduo de 16 genes.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}

um gene de indivíduo abordado no algoritmo.

Assim cada indivíduo possui um conjunto finito de genes que podem assumir os valores reais compreendidos entre $(-1, -1)$ e $(1, 1)$.

Logo os AGs que fazem o treinamento de dicionários de 16 pontos possuirão indivíduos 16 genes e assim sucessivamente. Os valores de genes utilizados são 16, 64, 256, 1024, 4096 e 16384. Como mostrado na Tabela 3.3 onde temos um indivíduo de 16 genes ou um dicionário com 16 pontos.

Os AGs desenvolvidos inicializam 20 indivíduos ou dicionários por experimento. Cada um desses indivíduos evoluirá segundo a teoria da luta darwiniana pela sobrevivência onde ao final de cada experimento restarão somente os 20 melhores indivíduos de cada experimento, como descrito nas seções a seguir. A Tabela 3.4 mostra a nomenclatura utilizada no desenvolvimento do trabalho.

Tabela 3.4: Metáfora de VQ utilizada no AG desenvolvido.

Nomenclatura AG	Interpretação em VQ
Problema	Reduzir a distância média dos pontos aos centros
Indivíduos	Dicionários
Genes	<i>Codeword</i>
Métrica	Distância média quadrática dos pontos aos centros

3.5.2 Geração da população inicial

Ao início de cada experimento são criados 20 indivíduos do genótipo (número de genes ou *codewords* M) definido pela simulação, como definido no livro [35]. Os 19 primeiros indivíduos possuem genes gerados aleatoriamente dentro do espaço de busca de um sinal de LTE normalizado em blocos. Assim estão compreendidos entre $(-1, -1)$ à $(1, 1)$.

O último indivíduo, no entanto, é um dicionário já salvo, treinado em uma simulação externa ao AG. Este indivíduo pode ser por exemplo um dicionário de LBG no AG híbrido com LBG ou um dicionário treinado com Kmeans++ no AG híbrido com Kmeans++ ou seja o *software* já recebe 1 mínimo local na sua inicialização. Este indivíduo é inserido como forma de acelerar a busca de ótimos locais, fornecendo um indivíduo para que a população possa evoluir a partir deste.

3.5.3 Função de Aptidão

Cada um dos AGs desenvolvidos neste trabalho possuem por função de aptidão ou *fitness* a média das distâncias euclidianas quadradas aos seus respectivos centroides (MSE, *Mean Squared Error*) como mostrado na Equação 3.3.

$$f = \frac{1}{N} \times \left(\sum_{i=1}^N D(x_i, C)^2 \right) \quad (3.3)$$

Onde N , x e D representam respectivamente o número de pontos no conjunto de treinamento, um ponto no conjunto de treino X e a distância entre um ponto do conjunto de x_i e seu centro mais próximo C .

3.5.4 Avaliação e Otimização Local

Nesta etapa cada um dos 20 dicionários são passados como entrada para uma função de otimização que pode ser o Kmeans++ ou o LBG. Esta abordagem foi proposta em [49].

Esta etapa utiliza a busca paralela do AG para otimizar os dicionários com o auxílio do LBG ou do Kmeans++. Pois esta reduz a possibilidade de encontrar um mínimo local ao final dos experimentos [50].

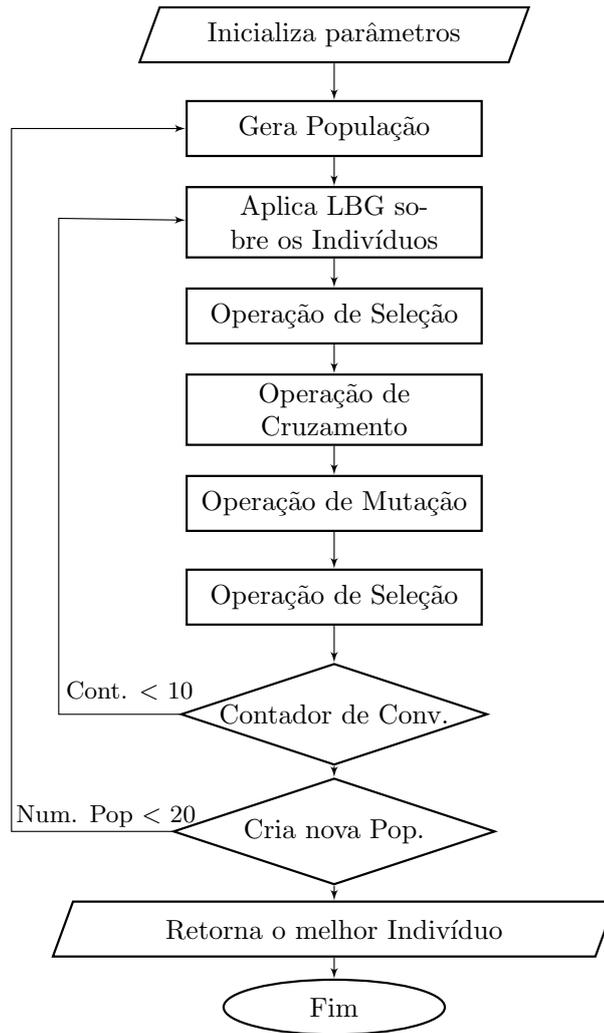


Figura 3.6: Fluxograma do *software* que utilizando o LBG.

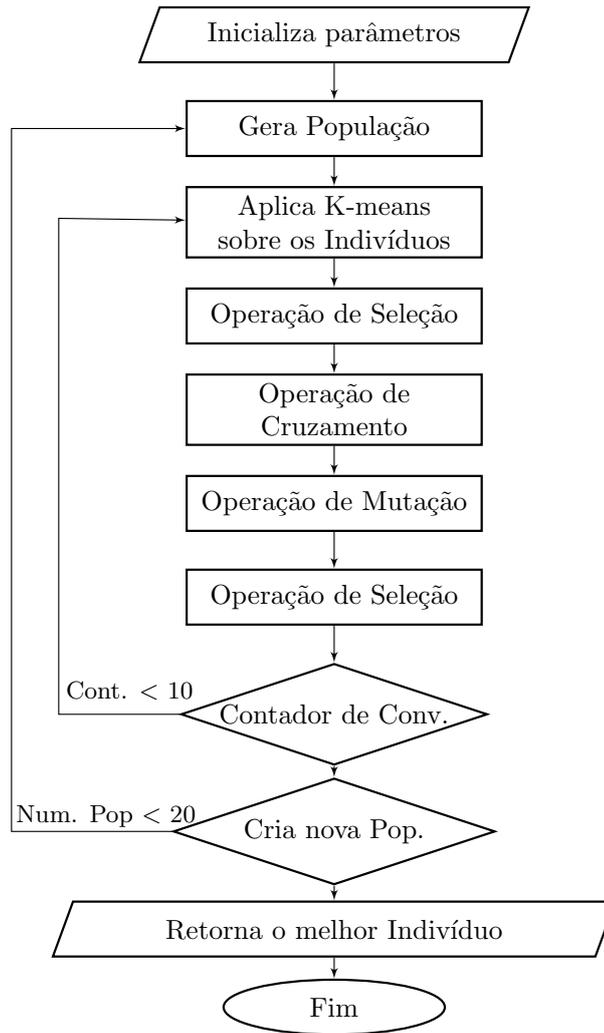


Figura 3.7: Fluxograma do *software* que faz o uso do K-means++.

O procedimento de Otimização de indivíduos é nos fluxogramas das Figuras 3.6 e 3.7 e ocorre da seguinte forma:

- No AG híbrido com LBG os indivíduos são passados como entrada para uma iteração da rotina de atualização de dicionários retirada do *toolbox vqdttool*. Esta rotina recebe como entrada o indivíduo e o conjunto de treino e otimiza os dicionários durante 30 iterações.
- O AG híbrido com Kmeans implementa a otimização dos indivíduos utilizando 30 iterações de correção de dicionários utilizando a rotina Kmeans. A rotina recebe a princípio os dicionários e o conjunto de treino.

Após isto são calculados os *fitness* de cada um dos indivíduos em relação ao conjunto de treino, utilizando-se a Equação 3.3.

3.5.5 Seleção de parentes

O algoritmo de treinamento utiliza a Seleção por Torneio, onde $N_{ring} = 2$ indivíduos são sorteados aleatoriamente e somente o indivíduo de menor *fitness* é selecionado para a reprodução. A seleção se repete por 20 vezes, tendo como resultado 20 indivíduos escolhidos nos torneios.

3.5.6 Cruzamento

O Crossover implementado no trabalho, é do tipo *Crossover* em um ponto e segue os seguintes passos:

1. O operador de reprodução seleciona um indivíduo i e um parceiro que não pode ser o próprio i .
2. Uma variável aleatória decide onde se ocorrerá o cruzamento com a probabilidade de cruzamento T_{cruz} .
3. Após isto eles trocam os genes a partir da posição selecionada aleatoriamente.
4. Então o indivíduo $i + 1$ é selecionado e retornamos ao passo inicial até que todos os indivíduos tenham passado pelo teste de cruzamento.

3.5.7 Mutação

O software utiliza a mutação uniforme de acordo com as seguintes etapas:

1. Um indivíduo i é selecionado.
2. Uma variável aleatória decide se esse indivíduo sofrerá mutação com probabilidade igual a T_{mut} .
3. Caso ocorra a mutação, todo esse indivíduo é multiplicado por um fator de escala com média 0 e desvio padrão $\delta = 0, 2$.
4. Após isto o indivíduo $i + 1$ é selecionado e retornamos ao passo inicial até que todos os indivíduos tenham passado pelo teste de mutação.

O livro [38], destaca a importância da mutação ser feita com cautela. Desta forma a implementação fez a opção pela multiplicação pela variável aleatória de desvio padrão δ para prevenir que o problema encontre soluções fora do domínio de busca.

A Figura 3.2 demonstrou que as amostras de LTE encontram-se concentradas entorno da média 0, desta forma a multiplicação por uma variável aleatória próxima ao valor nulo estimula a busca a soluções potenciais, como explicitado no Capítulo 4.

Além disto a seleção de sobreviventes garante que indivíduos não aptos passem para as próximas gerações.

A ideia da mutação é explorar o entorno da atual solução na busca de um indivíduo melhor.

3.5.8 Seleção de sobreviventes

Nesta etapa o *software* ordena a população pais em conjunto com a população de filhos, ordena e seleciona os indivíduos com menores *fitness*. Nesta seleção de sobrevivência indivíduos com *fitness* altos e indivíduos que não atendem as restrições do problema serão eliminados na seleção elitista. Serão selecionados para a próxima geração somente os melhores indivíduos dentre os pais os filhos.

3.5.9 Critério de convergência

Após estas operações, a seleção elitista garante que somente os melhores indivíduos passem para a próxima geração que evoluirá até que o critério de parada seja satisfeito:

- Ocorram 10 gerações seguidas sem melhorias no *fitness* do melhor indivíduo. Desta forma cada experimento tem no mínimo 10 gerações.

Como *fitness* muito pequenos são admitidos no problema, o AG geralmente convergia prematuramente. Desta forma foi criada uma variável auxiliar de convergência que quando multiplicada por 10^{10} possui a sua precisão aumentada. Como mostrado na Equação 3.4 onde é calculada a variável que controla a convergência:

$$conv = (f_n - f_{n+1}) \times 10^{10} \quad (3.4)$$

Onde f_{n+1} representam o *fitness* do melhor indivíduo criado na reprodução e f_n os seus pais. Se a variável *conv* for igual a 0 por gerações seguidas, um novo experimento será iniciado.

O procedimento é repetido por 10 experimentos e o melhor indivíduo é extraído de cada experimento, como mostrado nas Figuras 3.6 e 3.7 do apêndice que mostram os fluxogramas o algoritmo de treinamento de dicionários.

O dicionário obtido ao final do processo de treinamento é mostrado na Figura 3.8, que mostra o dicionário e o conjunto de treino que foi utilizado no seu treino. As regiões de *Voronoi* delimitam os pontos mais próximos de cada *codeword* ou gene.

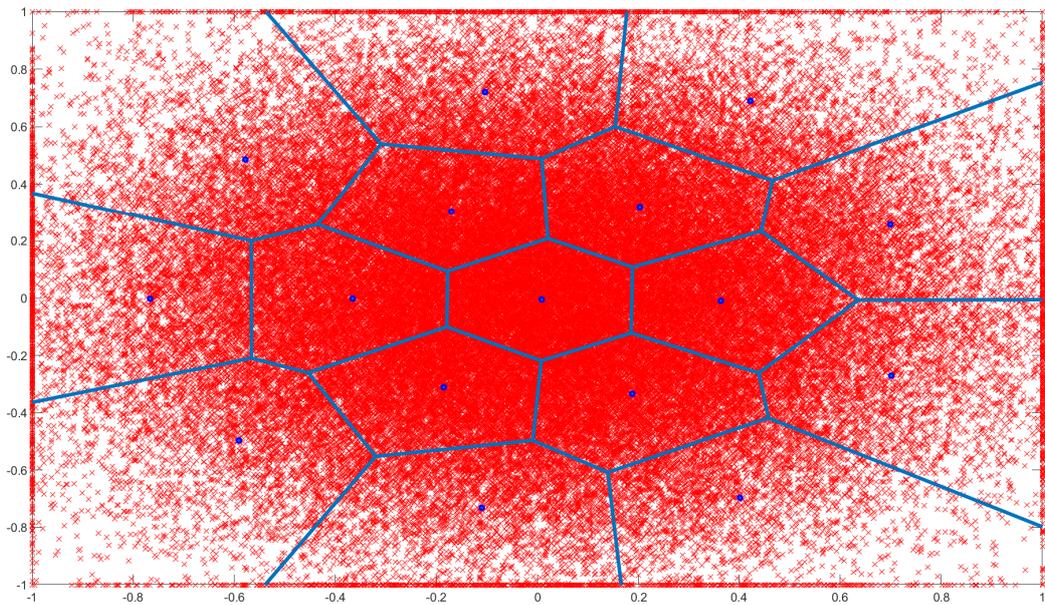


Figura 3.8: Dicionário treinado e conjunto de treino.

3.5.10 Métrica de avaliação do desempenho dos dicionários

Nesta parte do trabalho será avaliado o custo computacional de cada uma das soluções baseadas no número de gerações que o AG levou para alcançar tal resultado.

A qualidade das soluções é avaliada em relação ao *fitness* dos indivíduos produzidos, utilizando-se a MSE definida na Equação 3.3.

O desempenho dos AGs é dado em termos da Equação 3.5 que avalia a evolução do *fitness* pelos AGs, como mostrado na Equação 3.5:

$$P = \left(\frac{(f_i - f_{AG})}{f_i} \right) \times 100\% \quad (3.5)$$

Onde f_i representa o *fitness* da solução inicial e f_{AG} representa o *fitness* após o re-treinamento dos dicionários com o auxílio dos AGs. A Equação 3.5 busca avaliar o quanto um AG produziu de melhoras percentuais em relação ao *fitness* do dicionário inicial.

3.6 Conclusão

Neste trabalho foram desenvolvidos 2 softwares. Um treina dicionários para o processo de VQ e o outro faz compressão de sinais de LTE utilizando o dicionário treinado no primeiro.

Cada um dos *softwares* possui uma saída e uma métrica de avaliação, o primeiro é avaliado pelo *fitness* e o segundo pela EVM. Os resultados são descritos no capítulo a seguir.

Capítulo 4

Resultados

Introdução

Neste capítulo são discutidos os resultados obtidos nos algoritmos discutidos no Capítulo 3. Do algoritmo de treinamento que recebe como entrada os sinais I/Q de LTE, as variáveis de simulação e treina os dicionários treinados com o auxílio do Algoritmo Genético e do algoritmo de compressão que recebe os parâmetros de simulação e o dicionário treinado e retorna a EVM média das portadoras versus a compressão obtida.

Os resultados avaliam a eficiência do algoritmo sobre dois aspectos diferentes:

- Avaliando distorções médias obtidas nos treinamento dos dicionários versus complexidade da solução alcançada.
- Quantificando EVMs médias obtidas versus a compressão atingida nos testes.

4.1 Compressão utilizando somente sobre-amostragem

O algoritmo desenvolvido pode ser configurado de inúmeras formas. Nesta seção são apresentadas as simulações que utilizaram as seguintes etapas:

1. Remoção do prefixo cíclico.
2. Normalização em blocos.
3. Sobre-amostragem (ou *downsampling*).

Na etapa de sobre-amostragem as razões de $(\frac{K}{L})$ foram modificadas segundo [5] e [11]. Os resultados encontram-se na Tabela 4.1. As EVMs deste processo encontram-se descritas na Tabela 4.1.

Tabela 4.1: Valores de EVM reproduzidos utilizando somente sobre-amostragem.

$(\frac{K}{L})$	EVM (%)
$(\frac{15}{16})$	0.0098436
$(\frac{3}{4})$	0.0079349
$(\frac{2}{3})$	0.016105
$(\frac{5}{8})$	0.095875
$(\frac{15}{28})$	18.2302
$(\frac{15}{32})$	40.2909
$(\frac{5}{12})$	59.1062

A Tabela 4.1 examina o efeito da sobre-amostragem nas EVMs dos sinais de LTE. Estes sinais possuem redundâncias de aproximadamente $(1, 70 \times f_{Nyquist})$ no domínio da frequência [51]. Deste modo as razões de $(\frac{L}{K})$ inferiores a 1, 70 conduzem a EVMs próximas a 0, pois, removem somente redundâncias espectrais como explicitado no Capítulo 2.

No entanto, razões de $(\frac{L}{K})$ superiores a 1, 70 conduzem a EVMs notáveis, superiores a 20% indicando que houve perda de informação útil do sinal de LTE, a qual não poderá ser re-inserida no processo de descompressão do sinal.

4.2 Resultados

Nesta seção são avaliadas soluções, analisando-se as distorções e as taxas de compressão obtidas em cada um dos casos abordados.

Na subseção 4.2.1 é apresentado o critério de decisão das taxas de mutação e cruzamento. A subseção 4.2.2 discute o treinamento dos dicionários e as melhoras percentuais do erro de treino do AG, a subseção 4.2.3 discute os resultados do algoritmo de compressão.

As simulações a subseção 4.2.3 utilizaram as seguintes etapas:

1. Remoção do prefixo cíclico.

Tabela 4.2: Variáveis e valores utilizados nas simulações

Parâmetros	Valores
T_{am}	20.000 amostras
Num_{gen}	16
N_{ind}	20 indivíduos
N_{tor}	2
δ	0,2
T_{cruz}	0 : 0,1 : 1
T_{mut}	0 : 0,05 : 1%

2. Normalização em blocos.
3. Sobre-amostragem (ou *downsampling*).
4. Quantização Vetorial.
5. Codificação por Entropia.

4.2.1 Cenário final

O algoritmo é executado preliminarmente com a configuração da Tabela 4.2. As taxas de mutação e cruzamento são modificadas a cada simulação. As taxas de cruzamento variam de 0 á 100%, com o passo de 10%, as de mutação variam de 0 á 100%, com o passo de 5%. Assim foram executados 200 testes com o objetivo de decidir qual a melhor configuração média de AG.

Onde T_{am} representa o número de amostras reais de LTE sem o prefixo cíclico, normalizadas e sobre-amostradas de um fator de $(\frac{5}{8})$ utilizadas no treino. As constantes Num_{gen} , N_{ind} , N_{tor} e δ representam respectivamente o número de genes por simulação, o número de indivíduos por experimento, o número de indivíduos por rodada de torneio e o desvio padrão na mutação. As variáveis T_{cruz} e T_{mut} representam as taxas de cruzamento e mutação do AG que serão variadas nestas simulações a fim de encontrar a melhor configuração para o AG.

A decisão dos parâmetros utilizou como métrica a média do *fitness* melhor indivíduo. O procedimento para cada uma das taxas de mutação foi o seguinte:

Dada uma simulação com T_{cruz} e T_{mut} , calculado-se o *fitness* do melhor indivíduo em cada um dos 20 experimentos. E então é retirada média dos *fitness* dos 20 indivíduos escolhidos no passo anterior e realizado o logaritmo na base 10 da média dos indivíduos.

De forma que temos a Equação 4.1 onde k representa o experimento atual, f_k representa o *fitness* do melhor indivíduo do experimento k e N o número de experimentos total.

$$V = \log_{10} \left(\frac{\sum_{k=1}^N f_k}{N} \right) \quad (4.1)$$

Onde V é uma variável de avaliação que define comportamento médio de cada uma das simulações, como mostrado nas Figuras 4.1 e 4.2 que mostram os testes efetuados e os valores de V obtidos em cada uma das simulações de 16 genes.

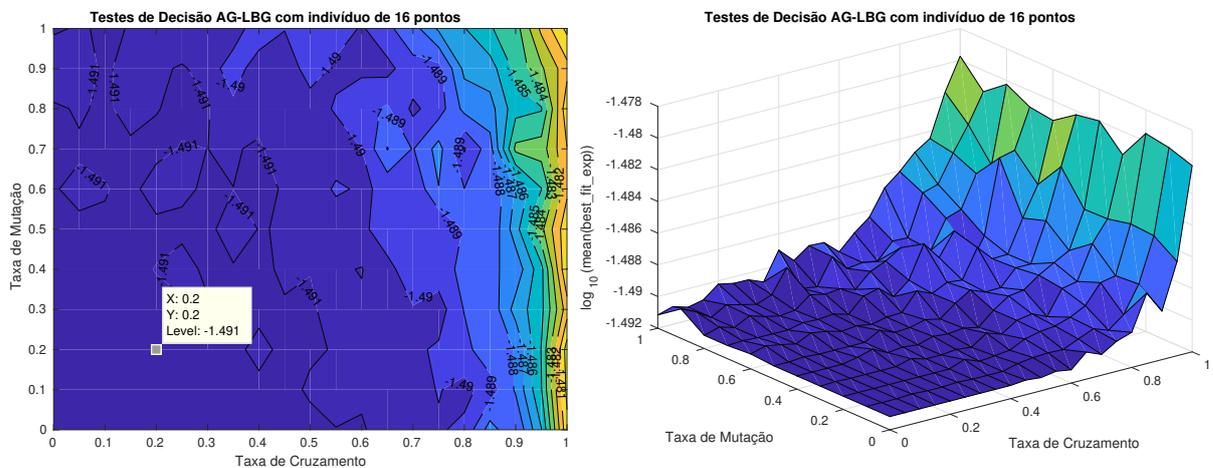


Figura 4.1: Testes de decisão de parâmetros para o AG-LBG.

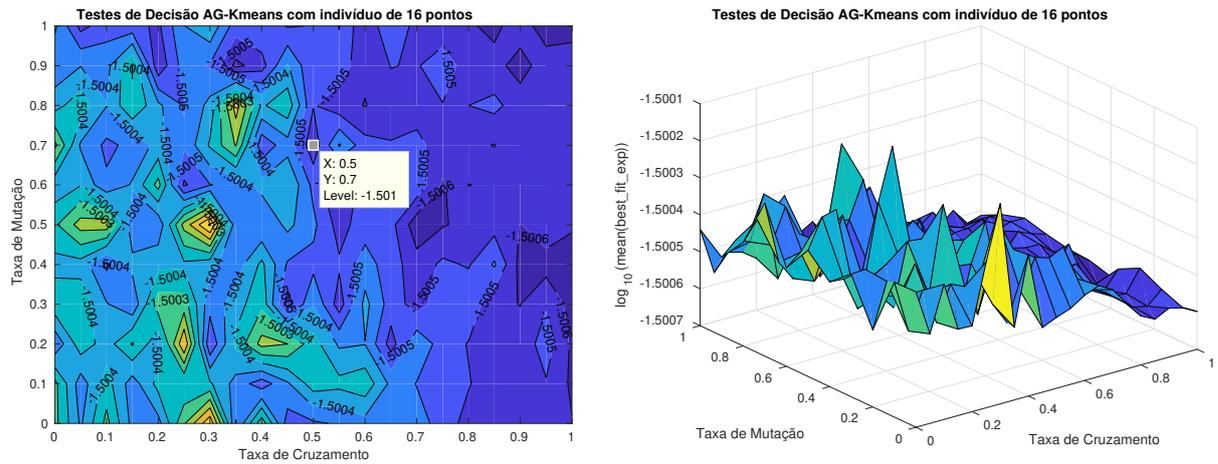


Figura 4.2: Testes de decisão de parâmetros para AG Kmeans.

Desta forma ao final de 200 simulações foram escolhidos os parâmetros que conduziram aos menores valores de V , como mostrado nas Figuras 4.2 e 4.1, logo os parâmetros escolhidos foram:

- Para o AG-LBG: $T_{cruz} = 0, 2$, $T_{mut} = 0, 3$ com o $V_{min} = -1, 491$
- Para o AG-Kmeans: $T_{cruz} = 0, 5$, $T_{mut} = 0, 7$ com $V_{min} = -1.501$

O parâmetro V foi utilizado para decidir as taxas de mutação de cruzamento a serem utilizadas. Pois foi admitido que os AGs com 1024,4096 e 16384 genes tenham aproximadamente o mesmo comportamento médio que o de 16 genes.

4.2.2 Avaliação do Algoritmo de Treinamento

Decorrida a seleção das taxas de mutação e cruzamento, ocorre o treinamento dos dicionários em AGs de 1024,4096 e 16384 genes assumindo-se que estes possuem o menor erro de compressão.

No treinamento dos dicionários foram utilizados os parâmetros mostrados na Tabela 4.3:

Tabela 4.3: Variáveis e valores utilizados nas simulações

Parâmetros	Valores
T_{am}	20.000 amostras
Num_{gen}	256, 1024, 4096, 16384.
N_{ind}	20 indivíduos
N_{tor}	2
δ	0,2
T_{cruz} AG LBG	20%
T_{mut} AG LBG	30%
T_{cruz} AG Kmeans	50%
T_{mut} AG Kmeans	70%
Q_a	8 bits por amplitude maxima

Onde as variáveis T_{am} , Num_{gen} , N_{ind} , N_{tor} e δ são as as respectivamente o número de amostras de treino, número de genes, número de indivíduos no torneio e o desvio padrão da variável gaussiana. Os parâmetros T_{mut} e T_{cross} representam respectivamente as taxas de mutação e cruzamento descritas na subsecção 4.2.1.

Nas Figuras 4.3 e 4.4 temos o treinamento de 2 dicionários de 1024 genes. Podemos notar que o processo de reprodução seguido de seleção elitista conduziu a indivíduos evoluídos em relação aos dicionários iniciais, segundo Equação 3.3. Os indivíduos que são inseridos na população e os que são obtidos ao final das simulações, são mostrados nas Tabelas 4.4 e 4.5.

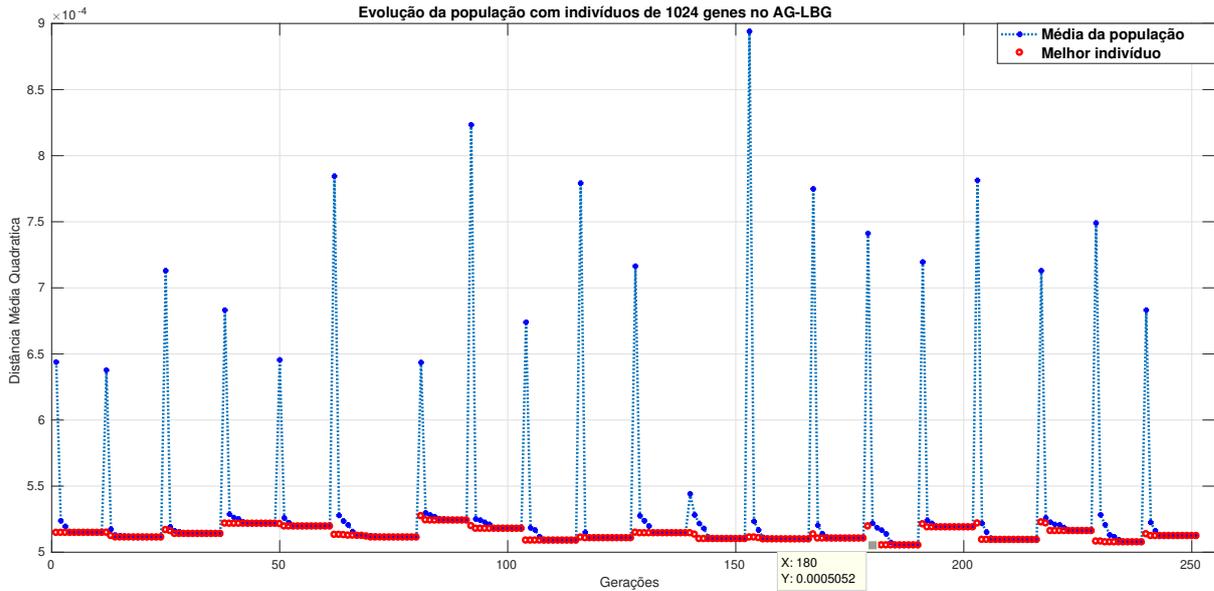


Figura 4.3: Treinamento do dicionário de 1024 pontos com AG-LBG

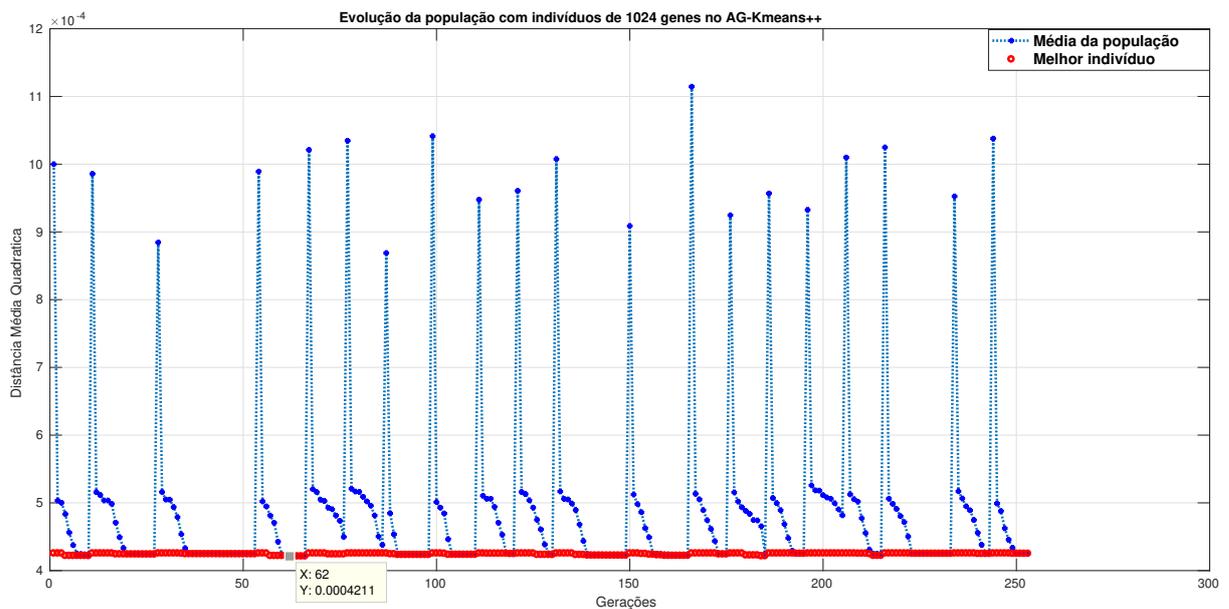


Figura 4.4: Treinamento do dicionário de 1024 pontos com AG-Kmeans

As Figuras 4.5 e 4.6 elucidam os erros no processo de treinamento de 2 dicionários de 4096 genes. No qual o processo de reprodução seguido de seleção elitista conduziu a indivíduos evoluídos em relação aos dicionários iniciais, segundo Equação 3.3. Os indivíduos que são treinados externamente e inseridos na população e os que são obtidos ao final das simulações, são mostrados nas Tabelas 4.4 e 4.5.

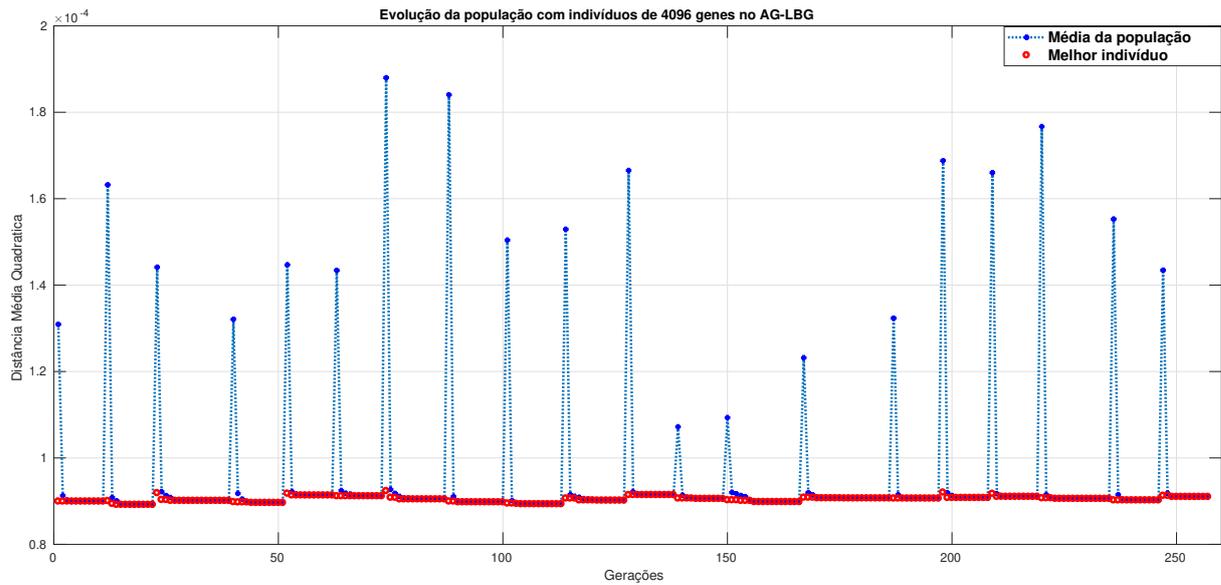


Figura 4.5: Treinamento do dicionário de 4096 pontos com AG-LBG

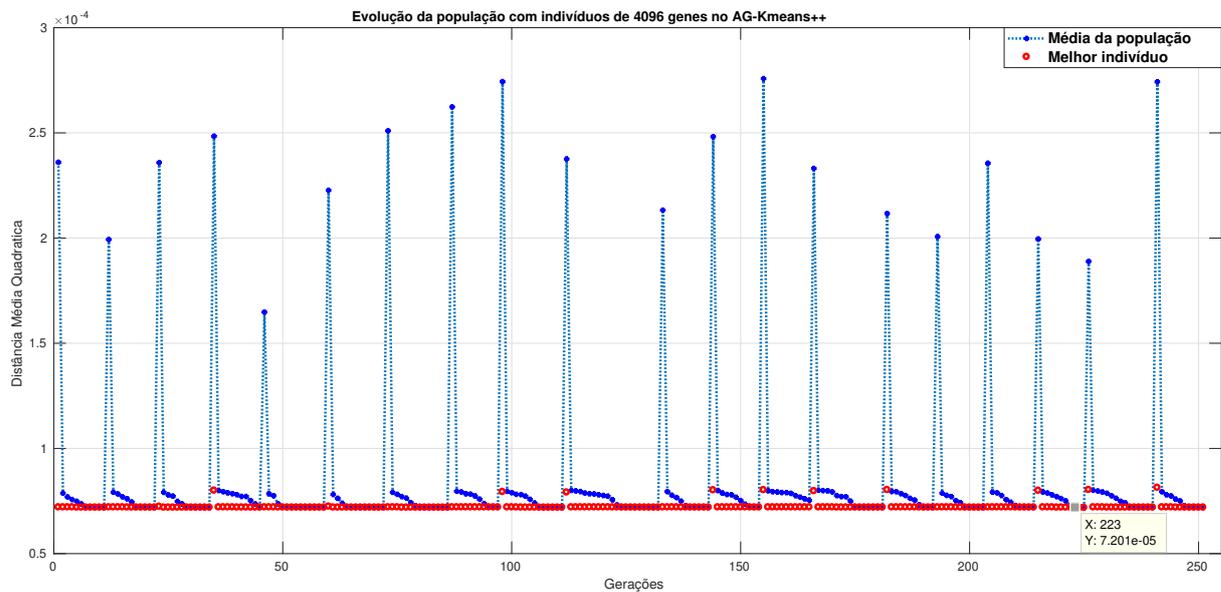


Figura 4.6: Treinamento do dicionário de 4096 pontos com AG-Kmeans

As Figuras 4.5 e 4.6 elucidam os erros no processo de treinamento de 2 dicionários de 4096 genes. No qual o processo de reprodução seguido de seleção elitista conduziu a indivíduos evoluídos em relação aos dicionários iniciais, segundo Equação 3.3.

As Tabelas 4.4 e 4.4 sintetizam os resultados obtidos no treinamento de dicionários utilizando o algoritmo proposto na seção 3.4. A métrica P , definida na Equação 3.5, avalia a

melhora percentual que o algoritmo conduziu em cada uma das simulações, como mostrado a seguir.

Tabela 4.4: MSE obtidos nos treinos de dicionários com LBG.

Tam. do dic.	LBG	AG-LBG	Num. de Ger.	P
1024	$6,2979 \times 10^{-4}$	$5,0524 \times 10^{-4}$	251	19,77%
4096	$1,8732 \times 10^{-4}$	$8,19 \times 10^{-5}$	257	56,27%
16384	$5,45 \times 10^{-5}$	$3,785 \times 10^{-6}$	371*	93,05%

O treinamento de dicionários com 16384 genes, em conjunto com o LBG só utilizou uma geração devido ao elevado custo computacional. O ganho de compressão obtido não justificou o gasto computacional com mais experimentos.

Tabela 4.5: MSE obtidos nos treinos de dicionários com Kmeans.

Tam. do dic.	Kmeans	AG-Kmeans	Num. de Ger.	P
1024	$4,255 \times 10^{-4}$	$4,188 \times 10^{-4}$	253	1,5746%
4096	$7,1933 \times 10^{-5}$	$7,1812 \times 10^{-5}$	251	0,1682%
16384	$1,0261 \times 10^{-6}$	$1,0149 \times 10^{-6}$	220	1,0915%

As Tabelas 4.4 e 4.5 demonstram que o AG foi capaz de melhorar o treinamento de dicionários em relação ao conjunto de treino inicial, ou seja, em relação a amostras reais dispostas em 2 dimensões.

4.2.3 Avaliação do algoritmo de Compressão

Os dicionários treinados com o auxílio do algoritmo desenvolvido e explicado no Capítulo 3, os resultados constam na Figura 4.7.

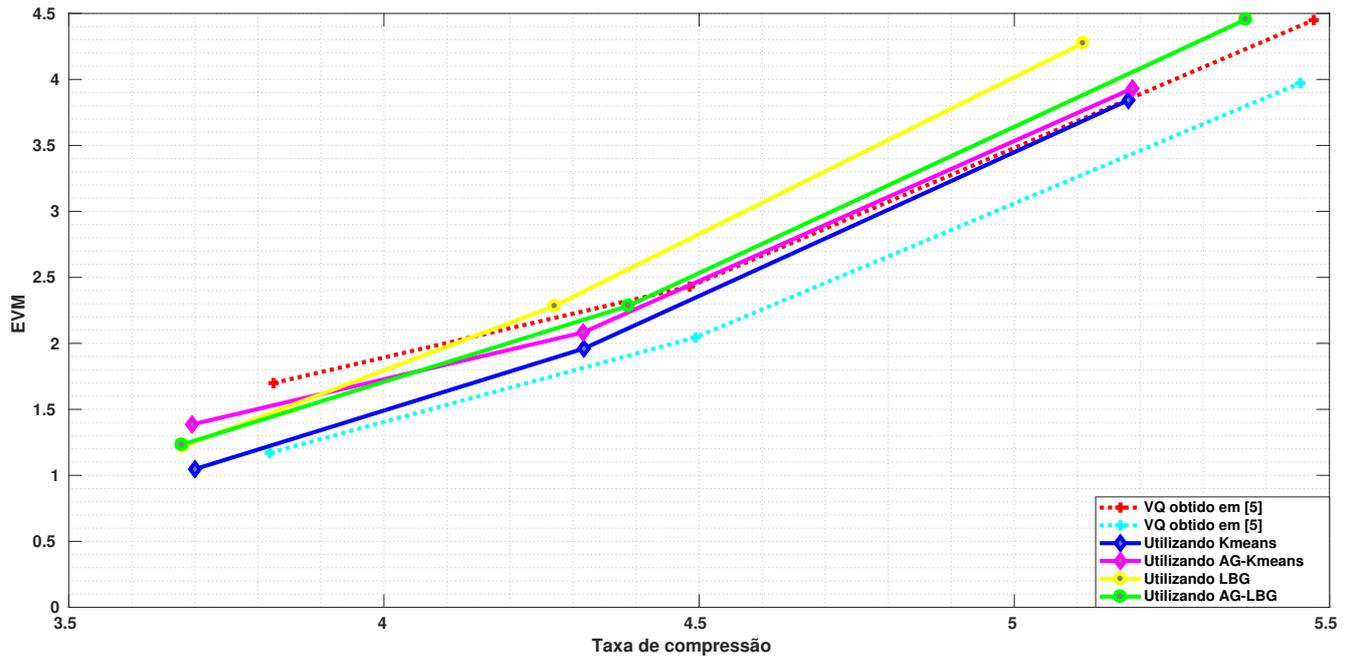


Figura 4.7: EVMs obtidas utilizando o método proposto.

A Figura 4.7 indica que treinamento com AG foi capaz de ampliar as taxas de compressão tanto do K-means++ quanto LBG.

No entanto este aumento, proporcionou um aumento nas EVMs dos sinais em estudo.

4.2.4 Dicionários Treinados

Nesta seção são apresentados os dicionários treinados com o auxílio dos AGs descritos na seção 4.2 são exibidos nas Figuras 4.8, 4.9, 4.10, 4.11, 4.12 e 4.13., como forma de reforçar as os resultados obtidos na Figura 4.7 .

As cores dos dicionários variam na escala colorida conforme a densidade de cada *code-word* em relação ao conjunto de testes.

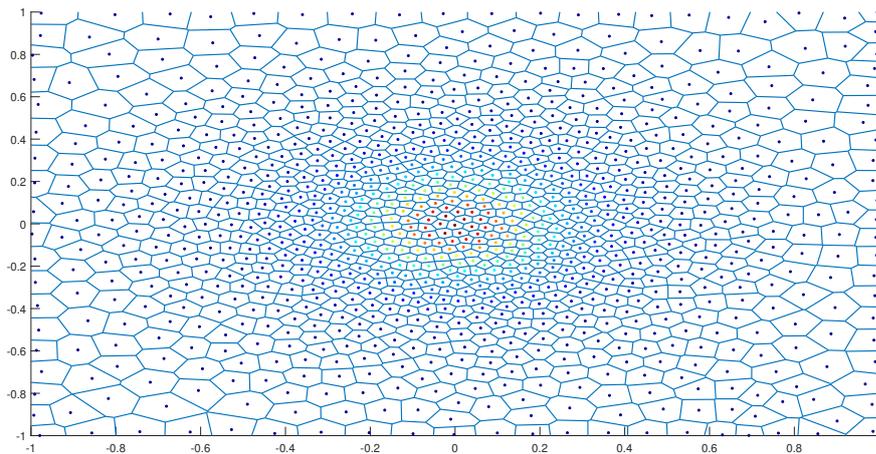


Figura 4.8: Dicionário de 1024 genes treinado com LBG e AG.

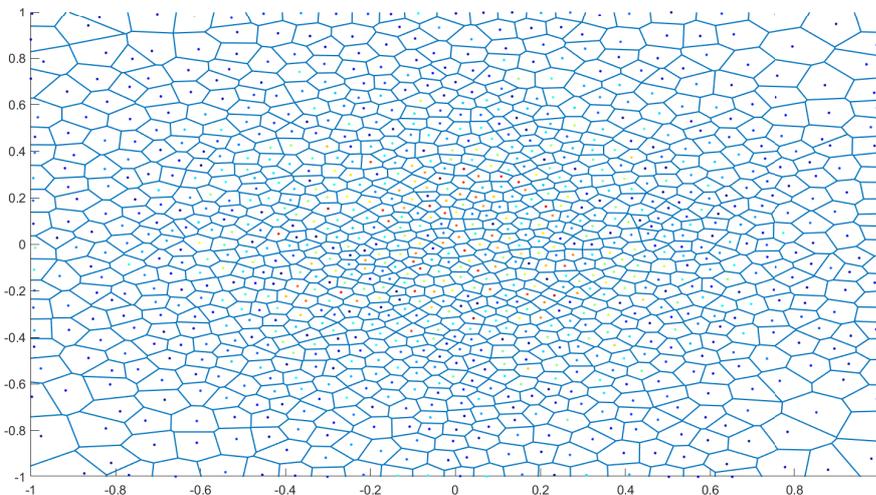


Figura 4.9: Dicionário de 1024 genes treinado com AG-Kmeans.

As Figuras 4.8 e 4.9 apresentam os dicionários de 1024 genes treinados com o auxílio dos AGs. Pode-se notar que o dicionário exibido na Figura 4.9, conduz a menores erros de quantização do que o dicionário da Figura 4.8, pois os dicionários treinados pelo AG K-means++ conduzem a regiões melhor espaçadas, reduzindo assim distorções.

No entanto os dicionários treinados com AG-LBG, conduzem a maiores erros de compressão como na Figura 4.8 que alcançou ótimas taxas de compressão, exibidas na Figura 4.7. Pois AG-LBG consegue criar mais regiões de menor entropia o que conduz a menores taxas e consequentemente a maiores ruídos de compressão.

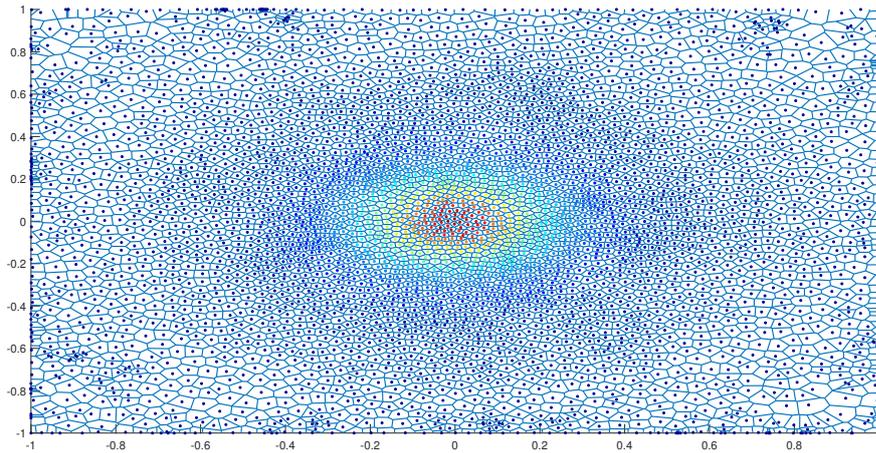


Figura 4.10: Dicionário de 4096 genes treinado com LBG e AG.

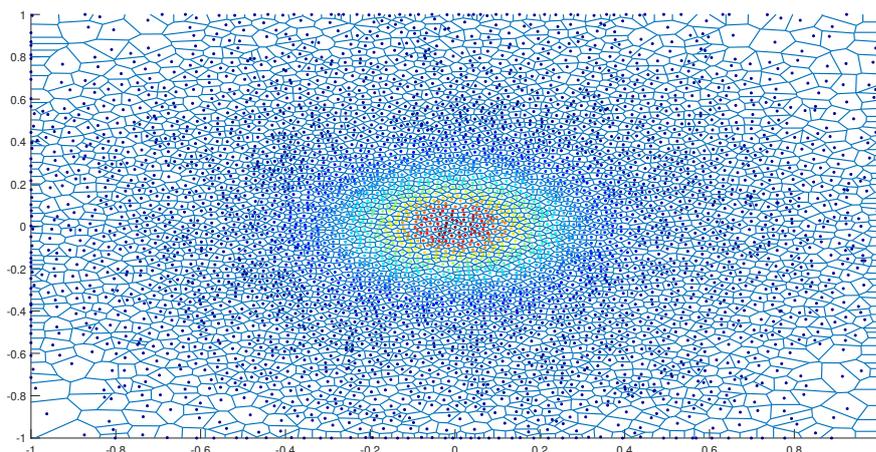


Figura 4.11: Dicionário de 4096 genes treinado com AG-Kmeans.

As Figuras 4.10 e 4.11 exibem os dicionários de 4096 genes treinados com o auxílio dos AGs. O comportamento apresentado nos dicionários de 1024 é novamente evidenciado. No qual os dicionários treinados com o auxílio do K-means++ conduzem a uma menor distorção e os dicionários treinados com o auxílio de LBG conduzem a maiores taxas de compressão.

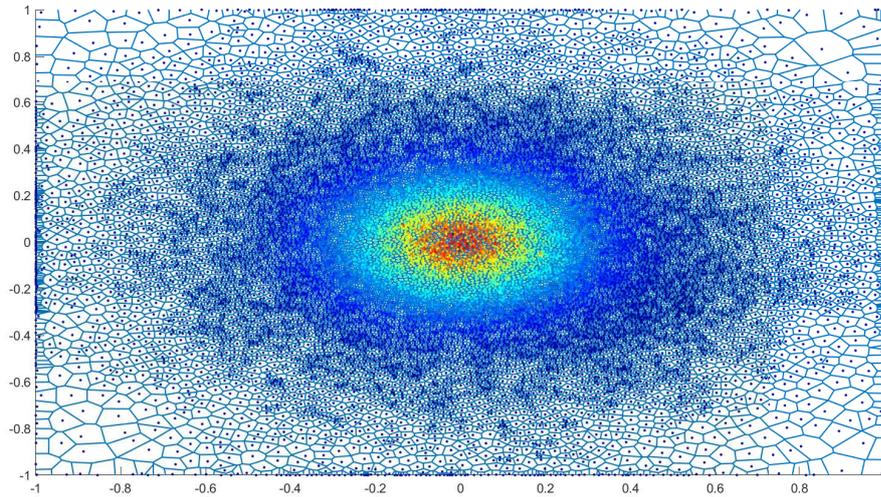


Figura 4.12: Dicionário de 16384 genes treinado com LBG e AG.

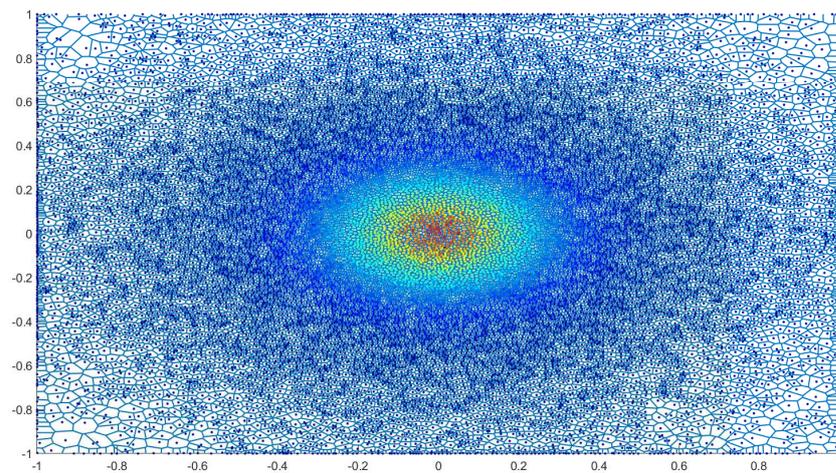


Figura 4.13: Dicionário de 16384 genes treinado com AG-Kmeans.

Nas Figuras 4.12 e 4.13 são exibidos os dicionários de 16384 treinados com AG-LBG e AG-Kmeans++. O dicionário da Figura 4.13 conduziu a menores distorções visto que o K-means++ efetua a divisão das regiões com o intuito de minimizar o erro total. Entretanto o dicionário treinado com o auxílio do AG- LBG exibido na Figura 4.12, conduziu a maiores distorções, e ampliou significativamente a taxa de compressão, com EVMs aceitáveis.

Isto ocorreu porque o AG-LBG proporcionou o treinamento de um dicionários de menor entropia, reduzindo assim o tamanho da informação a ser transmitida, resultando assim em taxas maiores de compressão e conseqüentemente maiores erros de compressão.

4.3 Conclusão

O uso dos AGs no treinamento de dicionários mostrou-se vantajoso pois consegue ampliação as taxas de compressão dos dicionários treinados. Conquanto é necessário um estudo acerca do treinamento, da especialização dos dicionários, para que seja estabelecido um AG padrão que conduza aos melhores resultados possíveis, com menores distorções e maiores taxas de compressão.

Capítulo 5

Conclusão

O método de treinamento de dicionários mostrou-se eficaz na tarefa de criar um dicionário sub-ótimo para o caso ideal.

Os resultados acima mencionados indicam que a aplicação que a performance do K-means em conjunto com os AGs conduziu a melhores resultados no quesito distorção imposta e com um custo computacional reduzido ao alcançar o critério de convergência em menos gerações, provando assim a eficiência da sua inicialização.

Os resultados apresentados indicam que com uso de AG para geração de *codebooks* no processo de VQ proposto para a compressão de dados em enlaces *fronthaul* possibilitam ampliar a capacidade de transmissão em cerca de 5,4 vezes, mantendo o EVM de aproximadamente 4,4%, respectivamente.

O uso do AG em conjunto com o LBG proporcionou o aumento da distorção, no entanto aumentou a capacidade de compressão em 0,2585 vezes em relação ao LBG. Provando a aplicabilidade no contexto das redes de alta capacidade, como por exemplo taxas da ordem de 100 Gbps necessária na expansão do *fronthaul*.

5.1 Sugestões para trabalhos futuros

Este trabalho mostrou que é vantajoso o uso de Algoritmos Genéticos no re-treinamento de dicionários sub-ótimos no caso de amostras I/Q de sinais de LTE então ficam como sugestões para trabalhos futuros implementar o treinamento de dicionários em cenários com distorções semelhantes as encontradas no *fronthaul* CPRI, e implementar a equalização dos símbolos transmitidos.

Uma outra tarefa futuras relevantes são:

- Realizar o treinamento com sinais de LTE com outras modulações digitais disponíveis.
- Modificar a *fitness* para que o problema busque reduzir o tamanho da informação e o ruído imposto no processo.
- Investigar algoritmos multi-objetivos e técnicas e otimização de problemas multi-objetivo que possam ser alternativas aos AGs.
- Fazer aprimoramento do algoritmo genético fazendo um estudo acerca do processo de especialização do dicionário e também, fazer o uso de técnicas que reduzam o tempo de treinamento dos dicionários e testar outras técnicas de clusterização ou EAs.
- Implementar e otimizar outras técnicas de VQ, como MSVQ e otimizar estas outras técnicas.

5.2 Publicações

Os resultados iniciais deste trabalho foram publicados com o trabalho intitulado: Algoritmo Computacional para Compressão de Sinais LTE [52]. Este trabalho foi premiado com o prêmio de melhor artigo na VIII Conferência Nacional em Comunicações, Redes e Segurança da Informação que ocorreu na Universidade Federal da Bahia nos dias 19, 20 e 21 de Outubro de 2018.

Os resultados finais foram submetidos ao IEEE *Latin-American Conference on Communications* e aguardam análise.

5.3 Conclusão

O aumento da centralização da arquitetura CRAN, bem como a crescente requisição de taxas tem motivado diversos autores a proporem técnicas de compressão de sinais I/Q de LTE ou amostras de CPRI.

Os sinais de LTE possuem diversas redundâncias que podem ser retiradas antes do *fronthaul* e reinseridas após o mesmo com o auxílio de técnicas de processamento de sinais.

Há um acordo entre capacidade de compressão e ruído imposto na compressão. Soluções que utilizam SQ conduzem a maiores distorções e menores atrasos. No entanto soluções baseadas em VQ introduzem erros menores e atrasos maiores.

O método implementado neste trabalho introduz pouco atraso na decodificação que é feita através de tabelas de busca e mostrou-se bem eficaz ao conceber taxas de compressão de 5,4 vezes com EVMs de aproximadamente 4,4%.

Bibliografia

- [1] A. N. C. J. e. C. L. C. Ranaweera, E. Wong, “5g c-ran architecture: A comparison of multiple optical fronthaul networks,” pp. 1–6, May 2017.
- [2] W. C. S. V. H. A. L. A. C. K. S. e. J. C. Z. J. G. Andrews, S. Buzzi, “What will 5g be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.
- [3] S. H. Z. X. G. L. e. Z. P. C. I, C. Rowell, “Toward green and soft: a 5g perspective,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 66–73, February 2014.
- [4] T. Q. e W. Yu, *Cloud Radio Access Networks: Principles, Technologies, and Applications*. Cambridge University Press, 2017.
- [5] M. S. R. e. J. Z. H. Si, B. L. Ng, “A novel and efficient vector quantization based cpri compression algorithm,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7061–7071, Aug 2017.
- [6] S. Park, O. Simeone, O. Sahin, and S. Shamai Shitz, “Fronthaul compression for cloud radio access networks: Signal processing advances inspired by network information theory,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 69–79, Nov 2014.
- [7] A. G. e R. M. Gray, *Vector Quantization and Signal Compression*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 2012.
- [8] M. M. S. W. e. R. V. D. Samardzija, J. Pastalan, “Compressed transport of baseband signals in radio access networks,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 9, pp. 3216–3225, September 2012.
- [9] A. T. e. D. S. B. Guo, W. Cao, “Lte/lte-a signal compression on the cpri interface,” *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 117–133, Sept 2013.
- [10] A. T. D. S. B. Guo, W. Cao, “Cpri compression transport for lte and lte-a signal in c-ran,” in *7th International Conference on Communications and Networking in China*, Aug 2012, pp. 843–849.

- [11] M. S. R. e. J. Z. H. Si, B. L. Ng, “A vector quantization based compression algorithm for cpri link,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.
- [12] S. Nanba and A. Agata, “A new iq data compression scheme for front-haul link in centralized ran,” in *2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, Sep. 2013, pp. 210–214.
- [13] “Free Lossless Audio Codec,” <https://xiph.org/flac/>, July 2019.
- [14] T. P. e. C. M. Z. He, C. Hu, “A compression scheme for lte baseband signal in c-ran,” in *9th International Conference on Communications and Networking in China*, Aug 2014, pp. 202–207.
- [15] R. Salman, V. Kecman, Q. Li, R. Strack, and E. Test, “Fast k-means algorithm clustering,” *International Journal of Computer Networks Communications*, vol. 3, 08 2011.
- [16] K. F. N. e B. L. Evans, “Time-domain compression of complex-baseband lte signals for cloud radio access networks,” in *2013 IEEE Global Conference on Signal and Information Processing*, Dec 2013, pp. 1198–1201.
- [17] M. M. Ebrahimi, M. Helaoui, and F. M. Ghannouchi, “Delta-sigma-based transmitters: Advantages and disadvantages,” *IEEE Microwave Magazine*, vol. 14, no. 1, pp. 68–78, Jan 2013.
- [18] D. A. e S. Vassilvitskii, “K-means++: The advantages of careful seeding,” vol. 8, 01 2007, pp. 1027–1035.
- [19] A. B. e. R. G. Y. Linde, “An algorithm for vector quantizer design,” *Communications, IEEE Transactions on*, vol. 28, pp. 84–95, 01 1980.
- [20] A. Company, “Lte resource guide,” Tech. Rep., 2015, version 1. [Online]. Available: www.cs.columbia.edu/6181/hw/anritsu.pdf
- [21] I. T. e. M. B. S. Sesia, *LTE - The UMTS Long Term Evolution: From Theory to Practice*. Wiley, 2011.
- [22] C. Cox, *An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications*. Wiley, 2012.

- [23] S. P. e. J. S. E. Dahlman, *4G: LTE/LTE-Advanced for Mobile Broadband*. Elsevier Science, 2011.
- [24] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*. Cambridge University Press, 2009.
- [25] C. Johnson, *Long Term Evolution in Bullets*. Johnson, 2012.
- [26] G. X. e. Q. H. Y. Ren, Y. Wang, "A compression method for lte-a signals transported in radio access networks," in *2014 21st International Conference on Telecommunications (ICT)*, May 2014, pp. 293–297.
- [27] T. I. Inc., "Lte in a nutshell:the physical layer."
- [28] S. N. Bernstein, "On the work of p. l. chebyshev in probability theory," in *The Scientific Legacy of P. L. Chebyshev*, no. First Part: Mathematics, 1945, p. 174.
- [29] T. Henk, *Understanding Probability: Chance Rules in Everyday Life*. Cambridge University Press, 2004.
- [30] D. L. e. A. A. A. de la Oliva, J. A. Hernandez, "An overview of the cpri specification and its application to c-ran-based lte scenarios," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 152–159, February 2016.
- [31] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.
- [32] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2012.
- [33] "Common public radio interface (cpri) specification v6.1," http://www.cpri.info/downloads/CPRI_v_6_1_2014-07-01.pdf, jul, 2014.
- [34] Z. M. e D. Dasgupta, *Evolutionary Algorithms in Engineering Applications*, 1st ed. Berlin, Heidelberg: Springer-Verlag, 1997.
- [35] K. D. Jong, *Evolutionary Computation: A Unified Approach*, ser. A Bradford Book. MIT Press, 2006.
- [36] C. W. Ahn, *Advances in Evolutionary Algorithms: Theory, Design and Practice*, 1st ed., ser. Studies in Computational Intelligence. Springer, 2006.

- [37] J. S. A.E. Eiben, *Introduction to Evolutionary Computing*, 2nd ed., ser. Natural Computing Series. Springer-Verlag Berlin Heidelberg, 2015.
- [38] R. L. H. e S. E. Haupt, *Practical Genetic Algorithms with CD-ROM*. Wiley-Interscience, 2004.
- [39] S. N. D. S.N. Sivanandam, *Introduction to Genetic Algorithms*. Springer, 2008.
- [40] G. P. e M. Russo, “The enhanced lbg algorithm,” *Neural Netw.*, vol. 14, no. 9, pp. 1219–1237, nov 2001.
- [41] “Vector quantizer design,” <https://www.mathworks.com/help/dsp/ref/vectorquantizerdesign.html>, 2018.
- [42] “Kmeans,” <https://www.mathworks.com/help/stats/kmeans.html>, 2018.
- [43] X. W. e V. Kumar, *The Top Ten Algorithms in Data Mining*, ser. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2009.
- [44] 3rd Generation Partnership Project (3GPP), “Evolved universal terrestrial radio access (e-utra); base station (bs) conformance testing,” Technical Specification (TS) 36.141, 2017, version 14.3.0. [Online]. Available: <http://www.3gpp.org/dynareport/36141.htm>
- [45] —, “Evolved universal terrestrial radio access (e-utra); base station (bs) radio transmission and reception,” Technical Specification (TS) 36.104, 2017, version 14.3.0. [Online]. Available: <http://www.3gpp.org/dynareport/36104.htm>
- [46] P. V. e A. B. Bradley, “An improved algorithm for vector quantizer design,” *IEEE Signal Processing Letters*, vol. 7, no. 9, pp. 250–252, Sep 2000.
- [47] T. K. e. O. N. P. Fränti, J. Kivijärvi, “Genetic algorithms for codebook generation in vector quantization,” 1996.
- [48] D. J. e. H. W. M. Garey, “The complexity of the generalized lloyd - max problem (corresp.),” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 255–256, March 1982.
- [49] W. M. e Y. Siqing, “Improved k-means clustering based on genetic algorithm,” in *2010 International Conference on Computer Application and System Modeling (ICCA SM 2010)*, vol. 6, Oct 2010, pp. V6–636–V6–639.

- [50] A. H. M. S. A. Soliman, *Modern Optimization Techniques with Applications in Electric Power Systems*, 1st ed., ser. Energy Systems. Springer-Verlag New York, 2012.
- [51] P. Diniz, E. da Silva, and S. Netto, *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2010. [Online]. Available: <https://books.google.com.br/books?id=-CIPeHVI9ZEC>
- [52] V. S. e J. C. W. A. Costa, “Algoritmo computacional para compressão de sinais 4g/5g,” *VIII Conferência Nacional em Comunicações, Redes e Segurança da Informação*, Oct 2018.

