PhD Dissertation

In Partial Fulfillment of the Requirements for

the Degree of Doctor of Philosophy from

Ecole Nationale Supérieure des Télécommunications

Computationally Efficient Adaptive Algorithms for Multicarrier Physical Layer

Presented by : **Asad Mahmood**

Defended on July 16$^{\text{th}}$ 2008, before the jury consisting of

Prof. Maurice BELLANGER	CNAM, Paris, France	Reviewer
Prof. Guillaume GELLE	Université de Reims, France	Reviewer
Prof. Jean-Claude BELFIORE	Telecom ParisTech, France	Thesis Supervisor
Prof. Alain SIBILLE	ENSTA ParisTech, France	Jury Member
Prof. Andreas CZYLWIK	University of Duisburg, Germany	Jury Member
Dr. Alban GOUPIL	Université de Reims, France	Jury Member

Contents

Acknowledgements

I would like to thank the director of my laboratory Professor Alain Sibille for extending his much needed support whenever needed. I would like to thank Dr. Omar Hammami for accepting me as a PhD student and giving me a chance to work at the prestigious institute of ENSTA. I would also like to thank Dr. Emmanuel Jaffrot and Professor Jean Claude Belfiore for the fruitful discussions concerning the directions and results of my research. I would also like to thank the administrative staff at UEI (Engr. Gille and Madam Darrozes) for always extending help in the administrative aspects and also my colleagues at ENSTA who were always there to lighten-up the tension involved in the work.

Last but not least, I would like to thank my parents, my siblings and my friends for always being a source of encouragement and without whose wishes and prayers, the completion of this thesis would not have been possible.

Asad Mahmood

ENSTA, Paris

Notations

Here is a list of the main notations and symbols used in this document. We have tried to keep consistent notations throughout the document, but some symbols have different definitions depending on when they occur in the text.

Chapter2 : Bit-Loading

N	Number of Sub-carriers
$Q(.)$	Q-function
n	subcarrier index
$\|h_n\|^2$	Channel Gain Energy
e_n	Energy Allocated to Sub-carrier n
b_n	Bo. of Bits Allocated to Subcarrier n
p_n^{err}	Probability of Error in Subcarrier n
B	Total No. of Bits to be Allocated
Δe_n^b	Energy req. to increase bits from b to b+1 in subcarrier n
Γ	Gap factor
γ_m	Performance Margin

Chapter3 : Peak-Energy Constrained Energy Allocation

∇_x	Differential with respect to variable x
$F(x, y, \lambda)$	Lagrangian Function
λ	Equality Constraint
ν_n	Inequality Constraints
M	Constellation Size

N_{max}	No. of Subcarriers with Peak/Max Energy
$\{h_{max}\}$	Set of indices of subcarriers with Peak/Max energy
λ_0	Initial Waterfilling Constant
ϵ	Difference between two Waterfilling Constants
E_{Surp}	Total Surplus Energy

Chapter 4: Irregular LDPC Codes Optimization

G	Generator Matrix
H	Parity-Check Matrix
s	original message
$\{.\}^T$	Matrix Transpose
$[.\|.]$	Matric Concatenation
$L()$	Likelihood function
r_{ji}	Message from check-node 'j' to variable-node 'i'
q_{ij}	Message from variable-node 'i' to check-node 'j'
d_v	degree of variable node
λ_x	Message Degree Distribution
ρ_x	Check Degree Distribution
$p^{err-corr}$	Probability of correcting an error
$p^{err-add}$	Probability of Addition of an error
k	Check-Degree
j	Variable-Degree
C_l^j	Binomial Coefficient

Acronyms

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first occurs in the text.

ALU	Arithematic Logic Unit
BPSK	Binary Phase Shift Keying
BER	Bit Error rate
BERM	Bit Error Rate Minimization
CSI	Channel State Information
DSL	Digital Subscriber Line
DMT	Discrete Multi-Tone
DSP	Digital Signal Processor
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FP	Floating Point
FCC	Federal Communications Commission
FPGA	Field Programmable Gate Array
GA	Genetic Algorithms
IEEE	Institute of Electronic and Electrical Engineers
ISI	Inter Symbol Interference
GIC	Greedy Irregularity Construction
ISR	Iterative Surplus Re-distribution
IWF	Iterative Water-Filling
KKT	Karush Kuhn Tucker
LDPC	Low Density Parity Check
MIPS	Million Instructions Per Second

MAC	Medium Access Layer
MCM	Multi Carrier Modulation
MM	Margin Maximization
MB	Majority-Based
MB-OFDM	Multi-Band Orthogonal Frequency Division Multiplexing
NLOS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
RTR	Run Time Re-configuration
RM	Rate-Maximization
RUU	Register Update Unit
SV	Saleh-Valenzuela
SPC	Sum of Product of Combinations
TDD	Time Division Duplex
UWB	Ultra Wide Band
WF	Water-Filling
WPAN	Wireless Personal Area Networks
WLAN	Wireless Local Area Network

Journals

- S. M. S. Sadough, A. Mahmood, E. Jaffrot, P. Duhamel, ”Multiband-OFDM: A New Physical Layer Proposal for Ultra Wideband Communications” , Journal of Iranian Association of Electrical and Electronics Engineers (IAEEE), November2007.
- A. Mahmood, J.C. Belfiore ”3-dB Subgroup based Algorithm for Optimal Discrete Bit-Loading”, Accepted with Revisions IEEE Transactions on Communications.

Conferences

- A. Mahmood, J.C. Belfiore ”Improved 3-dB Subgroup based Algorithm for Optimal Discrete Bit-Loading”, Proc. IEEE Sarnoff Symposium, Princeton Univ. USA, 27-30 April, 2008.
- A. Mahmood, E. Jaffrot, ”Computationally Efficient Algorithm for BER Optimized Power Allocation with Peak-Power Constraint”, Proc. IEEE VTC-Spring April 22-25 2007, Dublin, Ireland.
- A. Mahmood, E. Jaffrot, ”Greedy Check Allocation for Irregular LDPC Codes Optimization in Multicarrier Systems”, Proc. IEEE WCNC Mar 11-15 2007, Hong Kong.
- A. Mahmood, E. Jaffrot, ”New Efficient Method for Optimal Discrete Bit-Loading with Spectral Mask Constraints”, Proc. IEEE GLOBECOM Nov 27-Dec 1 2006, CA, USA.
- S. M. S. Sadough, A. Mahmood, E. Jaffrot, P. Duhamel, ”Performance Evaluation of IEEE 802.15.3a Physical Layer Proposal Based on Multiband-OFDM”, Proc. IEEE IST’05 Sept. 9-12 2005, Shiraz, Iran.
- A. Mahmood, O. Hammami, ”Hardware/Software Co-simulation of Adaptive MC-CDMA PHY layer for Wireless Communications”, Proc. IEEE ICM’05 Dec. 13-15 2005, Islamabad. Pakistan.

Résumé étendu en Français

Chapitre 1 - Introduction

L'usage accru de transmission de données dans nos jours, à faire augmenter la demande pour plus de débit. Les réseaux sans-fil essaient de répondre à la demande de haut débit dans la présence des plusieurs contraintes importantes concernant la transmission sans-fil. Un contrainte importante est sur la bande passante, imposées par les autorités réglementaires telles que la FCC (Fédéral Communications Commission) aux États-Unis. La deuxième contrainte vient du canal de transmission qui est l'air ou l'espace dans le cas de communications sans-fil. À haut débit, le montant de distorsion introduite à la transmission par la canal devient de plus en plus prononcée, ce qui rend la difficulté à compensation au niveau du récepteur. Puisque les exigences réglementaires du spectre ne peuvent être modifié ou changé, la but est d'améliorer la performance des systèmes de transmission numérique opérant dans divers canaux avec diverses techniques de traitement de signal. L'allocation adaptative des ressources [1–3] est une de ces méthodes pour améliorer la performance du système où les différents paramètres du système (taille de constellation, taux de codage, le puissance etc.) sont modifiées selon l'état de canal (CSI = Channel State Information). Cette allocation des ressources peut être effectués à plusieurs couches, mais nous sommes intéressées par les possibilités d'adaptation à la couche physique. Il a été démontré que l'adaptation de puissance et le taille de constellation pour un système M-QAM, peut donner un gain de près de 20 dB par rapport à la transmission non adaptative [4].

La modulation multi-porteuses (MCM = Multi-Carrier Modulation) [5] a révolutionné la technologie de couche physique pour des systèmes de communication, filaires ainsi que les sans-fil, au cours des deux dernières décennies. C'est la technologie de choix pour la couche physique des systèmes DSL [6],WLAN (IEEE 802.11x), WPAN (IEEE 802.15) et WiMAX (IEEE 802.16) parmi des nombreux systèmes utilisant MCM. MCM fonctionne selon un approche *diviser et conquérir*, c'est a dire la transmission est fait dans plusieurs sous-porteuses avec un débit de données plus faible, qui rendre simplification au l'égalisation à récepteur en traitant séparément la distorsion sur chaque sous-porteuse. En domaine fréquentiel, MCM transforme le canal sélectif en fréquence dans plusieurs petits canaux orthogonaux et sur lesquelles les données sont transmises en parallèle. Avec de nombreux avantages de MCM sur les systèmes de communications mono-porteuse (SCM = Single Carrier Modulation), telles que l'architecture simplifiées de récepteur, la capacité de la diversité en fréquence, la robustesse inhérente contre l'interférence inter-symbole (ISI), la répartition de bande passante en grand nombre de canaux bande étroite orthogonale donne la possibilité d'adaptation avec une granularité fine dans un canal sélective en fréquences.

Malgré tous les avantages de MCM cités ci-dessus, les systèmes MCM actuelle n'atteint pas leurs potentiel en raison de non-adaptation des paramètres de fonctionnement (e.g. taille de constellation, taux de codage, puissance émis etc.) par rapport l'état de canal (CSI) sur chaque sous-porteuse. Par conséquent, la probabilité d'erreur globale du système est dominée par les sous-porteuses les plus dégradées. Ainsi, pour améliorer les performances du système, l'impact de ces sous-porteuses devrait être atténués. C'est la raison d'être de l'adaptation des ressources ou de 'bit-loading' (lorsque le paramètre d'adaptation est la taille de constellation discret) pour MCM. Depuis l'usage des techniques d'adaptation dans systèmes MCM, un grand nombre des algorithmes, en utilisant toutes sortes de techniques d'optimisation, ont été élaborés et mis en place pour plusieurs systèmes MCM, y compris les modems DSL [7]. Malgré ces efforts, un grand nombre des contraintes importantes e.g. la délai de voie de retour, la qualité d'estimation du canal, le complexité des algorithmes etc. doivent être adressé pour l'utilisation de communications sans-fil en haut débit . Cette thèse aborde le problème de la complexité des algorithmes d'adaptation en proposant des nouveaux algorithmes d'optimisation pour MCM.

La complexité des algorithmes est ciblé sur le plan théorique / algorithmique ainsi que sur la coté architecture. Les contributions principales apportées dans cette thèse peuvent être énumérés comme suivants:

- La conception d'un nouvel algorithme de Bit-Loading (adaptation par rapport taille de constellation) basé sur un rythme d'allocation présent dans l'allocation optimale/greedy. Le nouveau algorithme a une complexité beaucoup plus faible que d'autres algorithmes

- Développements théoriques et conception d'un nouvel algorithme de répartition optimale de puissance totale en tenant compte la contrainte de puissance-maximale.

- La proposition d'une nouvelle méthode d'optimisation du profil d'irrégularité des codes LDPC irrégulière basé sur la quantification du phénomène de l'effet de vague (= 'Wave-Effect') et les développements théoriques conduisant à une méthode de calcul efficace.

- Une nouvelle méthodologie pour l'optimisation des ressources architecture pour un algorithme d'adaptation tenant en compte les contraintes temporel de la canal de la transmission en temps réel.

Chapitre 2 - Bit-Loading

Introduction

Modulation Adaptative se réfère généralement au adaptation de type de modulation/ taille de constellation (BPSK, QPSK, QAM-16 etc.) par rapport les conditions de la canal. C'est-à-dire, le mieux la canal, large le taille de constellation du symbole envoyé. Modulation adaptative dans les systèmes multi-porteuses est appelé **Bit-Loading** lorsque l'adaptation est fait dans les sous-porteuses indépendamment de l'un et l'autre et la taille de constellation est un nombre discret. La tâche d'optimiser le profil de bits discrets sur toutes les sous-porteuses, pour un état de canal donné, est un sujet important dans la recherche sous le nom de **Discrète Bit-Loading**. Il a été constaté que le profil optimale d'allocation des bits, pour un canal et nombre de sous-porteuses donnés, est différente pour différents objectifs (maximisation du débit, minimisation de puissance totale etc.) et contraintes (la puissance totale, nombre de bits totale, la puissance maximale etc.) associés à la problème d'optimisation.

L'objectif d'un algorithme d'allocation peut être variable: maximiser le débit, minimiser le taux d'erreur, minimiser la puissance totale etc. selon les besoins du système correspondant. Sur la base de l'objectif, il existe deux types de problèmes d'optimisation qui sont populaires dans la littérature: maximisation de débit et minimisation de puissance totale pour un débit donnée [8]. Nous définissons N comme la nombre totale des sous-porteuses et σ_n^2, $|H_n|^2$, e_n, b_n et p_n^{err} comme le puissance de bruit, le gain du canal, l'énergie attribué, le nombre de bits alloué et la probabilité d'erreur sur le sous-porteuse n. La problème de maximisation de débit traite de la maximisation du débit de données ($R = B/T_{sym}$) pour un ensemble des sous-canaux parallèles lorsque le débit symbole $1/T_{sym}$ est fixe. Cela exige la maximisation du nombre total de bits ($B = \sum_{n=1}^{N} b_n$) alloué sur tous les sous-porteuses, pour une énergie totale donné. Pour nombreux systèmes de communication pratique, un débit variable n'est pas souhaitable. Dans ce cas, au lieu de maximiser le débit, maximisation de la performance (*margin*) pour un débit donné, est ciblé. Pour maximiser le *margin*, l'énergie totale est minimiser pour un débit et taux d'erreur donnée.

État de l'Art

Concernant les méthodes existant d'allocation de bits, nous constatons que le solution *Water-filling* [9] et la solution *greedy* [10] sont bien connue comme les solutions optimale pour le cas continue et discret respectivement. Par contre, la recherche pour des algorithme moins complexe a entraîné le développement de diverses approches au cours des dernières années [11–15]. Hughes-Hartogs [10] a été le premier à établir l'algorithme *greedy* d'allocation des bits pour maximiser le débit pour une puissance totale donnée. Les propositions d'après [16–18] ont concentré sur *arrondissement* de la solution continue d'un problème d'optimisation pour réduire la complexité d'algorithme par rapport à l'algorithme classique *greedy* avec un minimum de différence de performance.

Les principaux travaux mathématique liés à allocation de bits discret ont été effectuées par Campello [11], qui a développés les *conditions* suffisantes pour un allocation discret optimale et a proposé des algorithmes d'allocation en utilisant les conditions *suffisant* et le facteur de *Gap* (Un approximation pour la prise en compte du comportement de type de modulation et de codage dans l'équation classique de capacité par Shannon [19]). Critiquant la validité de l'expression de 'Gap', Piazzo [12] a développé des algorithmes pour une allocation optimale des bits, basé sur un condition *nécessaire* pour l'optimalité. D'autres oeuvres importantes concernant l'allocation de bits discret ont été effectuée par Krongold [13] et Sonalkar [14], qui ont utilisé les techniques de *Lagrange discrète* et *Greedy-Bit Removal* pour arriver à la solution optimale. Bien que tous ces différents algorithmes fournissent une solution avec différent niveau de performance et celle de complexité, la contrainte de puissance maximale n'était pas exclusivement adressée. Les grands travaux liés à l'allocation de bits discret tenant en compte la contrainte de puissance maximale ont été effectuée par Baccarelli [20] et Papandreou [15]. Baccarelli utilise l'approche classique *greedy* pour respecter la contrainte de puissance maximale et Papandreou a proposé un méthode d'allocation en plusieurs phases pour converger vers la solution optimale respectant la contrainte de puissance maximale.

Algorithme 3-dB d'Allocation Optimale des Bits Discrètes

En utilisant de le paramètre de 'Gap', le nombre maximum de bits (b_n) par symbole qui peuvent être envoyés sur sous-porteuse n, dans un système multi-porteuses sont donnés par

$$b_n = \log_2(1 + \frac{e_n \cdot CNR_n}{\Gamma}) \tag{1}$$

Où CNR_n est la rapport de gain du canal et bruit et b_n et e_n sont des nombre de bits et l'énergie attribué au sous-porteuse n, respectivement. Γ est le paramètre de 'Gap' pour l'estimation de comportement de la type de modulation et de codage pour un critère de performance souhaités. En utilisant l'équation ci-dessus, l'énergie nécessaire pour transmettre un bit supplémentaire sur le sous-porteuse n est donnée par :

$$\Delta e^+_{b_n} = e_n^{b_n+1} - e_n^{b_n} = \frac{2^{b_n} \cdot \Gamma}{CNR_n} \tag{2}$$

La base conceptuelle de notre algorithme d'allocation se trouve sur le principe facilement observable de l'équation ci-dessus, qui donne une relation récursive entre les énergies dont nous avons besoin pour allocations des bits successives sur la même sous-porteuse. C'est a dire que l'énergie nécessaire pour incrémenter bits de b_n à b_{n+1} est deux fois (nécessite plus de 3 dB) de l'énergie nécessaire pour incrémenter bits de b_{n-1} à b_n. Mathématiquement, cette relation récursive est représentés comme

$$\Delta e^+_{b_n} = 2 \cdot \Delta e^+_{b_{n-1}} \Rightarrow \Delta e^+_{b_n dB} = \Delta e^+_{b_{n-1} dB} + 3dB \tag{3}$$

En utilisant le relation ci-dessus, nous pouvons diviser toutes les sous-porteuses en groupes de 3-dB par rapport leur gains de canal. Ainsi, si nous imaginons tous les sous-porteuses triés dans ordre décroissant par rapport leur gain du canal, le sous-porteuse avec le plus petit gain de canal sera au premier groupe et la sous-porteuse avec la plus grande gain de canal est assignés au dernier groupe. Le nombre de sous-porteuses dans chaque groupe ne sera pas identiques et sera déterminés par la qualité de canal concerné. Utilisant ce tri des sous-porteuses dans les différents groupes de 3-dB, nous avons détecté un 'rythme' sous l'allocation optimal 'greedy' qui est faite bit par bit et finalement employé cette 'rythme' ou 'mode de comportement' pour la conception de notre algorithme d'allocation optimale de bits discrète.

Sur la base de ces principes, si tous les sous-porteuse sont divisés en différentes groupes de 3-dB, nous avons divisé notre algorithme d'allocation en deux phases : allocation initiale et allocation finale.

En allocation initiale, notre objectif principal est de déterminer le nombre maximal de groupes de 3-dB qui seront nécessaires pour allouer le cible nombre de bits B. Pour cela, nous définissons ns_l comme le nombre de sous-porteuses dans le groupe l de sorte que $\sum_{l=1}^{L} ns_l = N$. Maintenant, tout le processus d'allocation peuvent être divisés en petites *étapes* j. Une étape peut être définie comme tous les allocation qui se-font entre les allocations successives sur le première sous-porteuse H_{max}.

Si nous supposons que notre cible nombres de bits B sera atteint au cours de l'étape j^* , chaque étape jusqu'à $j^* - 1$ sera allouer un bit à tous les sous-porteuses de tous les groupes de 3-dB impliqués au cours de cette étape. Par contre, au cours de l'étape j^*,ce ne pas nécessaire que tous les sous-porteuses concernés sont attribué aussi un bit parce-que la procédure d'allocation s'arrêtera dès que $b_{total} = B$. Le phase initiale de notre algorithme d'allocation, se-concerne des allocation de l'étape 1 jusqu'à l'étape $j^* - 1$, alors que la phase finale d'allocation se-concerne aux allocation qui se-font à l'étape j^*. Une fois la valeur de j^* est déterminé, on sait que $j^* - 1$ sous-groupes seronts être employées pendant la phase d'allocation initiale, parce-que avec chaque nouvelle étape, un sous-groupe supplementaire est impliqué dans le procedure d'allocation. Seulement un bit est attribuée à chaque sous-porteuse impliqués dans une étape. Par conséquent, à la fin de l'étape $j^* - 1$, $j^* - 1$ bits seront etre allouer à tous les sous-porteuses dans le groupe 1, $j^* - 2$ bits pour chaque sous-porteuse de groupe 2 jusqu'à un seul bit au tous les sous-porteuses de groupe $j^* - 1$. Donc, si j^* représente le nombre de groupes 3-dB impliqué dans l'allocation initiale, après la phase d'allocation initiale (c'est-à-dire après $j^* - 1$ étapes de l'allocation), le nombre de bits alloués au chaque sous-porteuse de groupe j peut être donnée par

$$b_j^{initial} = j^* - j \quad \forall \ 1 \leq j \geq j^* - 1 \tag{4}$$

La phase finale d'allocation détermine les sous-porteuses qui devraient être alloués un bit au cours de l'étape j^* pour atteindre le nombre de bits ciblé (B) telle que la profil d'allocation est le même que celle qui est réalisé par l'algorithme optimale de Hughes-Hartogs [10]. Dans [21], nous avons proposé une méthode d'allocation finale basés sur le tri des sous-porteuse par rapport leur gain de canal. D'une part, cette méthode étant simple autrement rend complexité à la procédure d'allocation, en particulier, s'il reste peu nombre de bits à allouer dans la dernière phase. Sur la base de la 'rythme' sous la procedure d'allocation, nous avons constaté que la complexité algorithme peuvent être largement réduits en remplaçant la méthode ci-dessus avec une autre approche où chaque groupe de 3-dB est encore sous divisée en différents 'intervalles' [22].

Résultats Simulation

Le profil d'allocations de bit et la complexité de notre algorithme ont été comparée avec deux autres algorithmes optimales d'allocation de bit, Hughes-Hartogs [10] et celle de Papandreou [15] . L'algorithme proposé est valide pour n'importe quel système multi-porteuse mais pour un scénario d'application, on a pris les paramètres correspondants à système UWB basé sur MB-OFDM (MultiBand-OFDM), et par conséquent, le modèle de canal UWB [23] a été employé pour la simulation. Les profil d'allocations de trois algorithmes pour des scénarios différents de la canal ont été trouvée à être exactement identiques, vérifiant l'optimalité de notre algorithme. La complexité de notre algorithme a également été comparée avec celle des deux autres algorithmes, non seulement sur la base de la complexité théorique, mais aussi fondée sur le nombre exact des cycles d'exécution de chaque algorithme sur le processeur Simplescalar. Nombre total de cycles d'exécution a été évalué pour différents paramètres de systèmes, vérifiant le meilleur complexité de notre algorithme.

Conclusion

Nous avons proposé un nouvel algorithme d'allocations des bits discrets basé sur la répartition de toutes les sous-porteuses en groupe de 3-dB. L'analyse de complexité et de résultats de simulation a montré la convergence rapide de notre méthode vers la solution optimale par rapport à certains algorithmes récemment proposés pour optimiser la répartition de bits discrets. Pour une véritable analyse de complexité, nous avons mis en place l'ensemble des algorithmes comparées sur un processeur SimpleScalar pour effectuer une comparaison de complexité basé sur nombres de cycles d'exécution. Enfin, le travail peut être poursuivi en direction de développer la stratégie de répartition optimale de puissance et d'adaptation de codage canal basée sur la répartition des sous-porteuses en groupe de 3dB.

Chapitre 3 - Allocation d'énergie sous la contrainte d'énergie-maximale

Introduction

En générale, l'objectif d'un algorithme d'adaptation est de trouver la meilleure distribution d'un ou plusieurs paramètres (taille de constellation, taux de codage, l'énergie etc.) de systèmes sur tous les sous-porteuses afin d'optimiser un attribut de l'ensemble du système, e.g. l'énergie totale $E = \sum_{n=1}^{N} e_n$, le nombre de bits à allouer $B = \sum_{n=1}^{N} b_n$ ou taux d'erreur globale $= BER_{avg}$ avec contraintes sur autres attributs. Dans chapitre 2, nous avons traité avec la meilleure distribution / répartition du paramètre b_n afin d'optimiser (minimiser) l'énergie totale, avec la stricte limitation de valeur discret pour b_n. Bien que cette contrainte discrète est vrai pour l'optimisation du paramètre b_n, l'optimisation de la puissance / énergie (e_n) peut se faire libre de cette contrainte et, par conséquent, les techniques d'optimisation qui existent pour les variables continues peuvent être adoptées dans ce contexte. Le but de ce chapitre est donc d'employer un tel technique classique d'optimisation des variables continues pour trouver la distribution optimale du paramètre e_n, en tenant compte nos propres objectifs et contraintes.

Nous avons vu dans ce chapitre q'un changement d'objectif ou des contraintes du problème d'optimisation, conduit à un changement du profile de répartition optimale de la paramètre concerné. Contrairement water-filling, où l'objectif est de maximiser la capacité (débit) du système, notre objectif dans ce chapitre est d'arriver a la répartition de l'énergie totale E sur tous les sous-porteuses afin d'optimiser/minimiser le taux d'erreur globale (BER) du système, quand le taille de constellation est la même dans tous les sous-porteuses.

La distribution de l'énergie totale est ensuite optimisée par rapport le BER en présence de la contrainte de l'énergie maximale. Ce type de contrainte est présent pour éviter l'interférence entre plusieurs systèmes, qui partagent la même bande passante. Cette contrainte est définie en termes d'un masque de la Densité Spectrale de Puissance (DSP), qui détermine la puissance/énergie maximale à transmettre à chaque fréquence de la bande-passante. Puisque cette contrainte de DSP peut se varie dépendamment des règlements locale, modification de théorie et les algorithmes d'adaptation, afin de tenir compte cette contrainte supplémentaire de l'energie-maximale, est d'un intérêt particulier. Enfin, un algorithme efficace de l'allocation optimale d'énergie qui respecte la contrainte d'énergie maximale est présenté et la complexité réduite de notre algorithme par rapport la solution classique de Waterfilling-Itératif, est démontré au travers le temps de simulation sur plusieurs cas.

État de l'Art

Un des premiers travaux concernant le répartition optimale d'énergie afin d'optimiser le BER globale, est celui de Fischer [18] qui aborde le problème plus pratique de maximiser le SNR pour un débit donnée et le contrainte sur puissance totale. En utilisant l'optimisation Lagrange, la fonction objective de SNR moyenne de système est maximisée afin de partager le débit total sur toutes les sous-porteuses. Goldfeld [24] a utilisé le facteur de *partitionnement* pour quantifier la répartition de l'énergie totale sur les différentes sous-porteuses. La fonction de taux d'erreur moyenne de système est optimisée sous la contrainte de l'énergie totale. La solution optimale a été trouvée, composée de N équations transcendantales et puis, une solution plus simple et quasi-optimale a été proposé. Une bonne analyse sur les caractéristiques du BER-optimale répartition d'énergie a été présenté par Chang [25], qui a utilisé la borne exponentielle [26] de la fonction de taux d'erreur binaire et optimisés sous la contrainte d'énergie totale. Il a observer que les caractéristiques du profil d'allocation d'énergie optimisé pour BER est diffèrent de celle de la waterfilling [27] qui maximise la capacité. A haute SNR, le système optimisé par rapport BER alloue plus d'énergie à la sous-porteuse le plus atténué, ce qui est contraire au comportement de la waterfilling.

Allocation d'énergie avec contrainte de l'énergie-maximale

Lagrange optimisation est la meilleur méthode pour trouver la répartition optimale de e_n sur les sous-porteuses afin de minimiser notre fonction objectif de BER avec les contraintes de l'énergie totale et l'energie-maximale. Cette problème d'optimisation peut etre transformer dans la forme classique d'une problème d'optimisation Lagrange comme suivant

$$\begin{aligned} minimise\ \ p_{avg}^{err} &= \frac{1}{N}\sum_{n=1}^{N} f\left(h_n \cdot e_n\right) \\ subject\ \ to\ \ \sum_{n=1}^{N} e_n - E_{total} &= 0 \\ e_n - \overline{e} &\leq 0 \end{aligned} \tag{5}$$

Introduisant λ comme multiplicateur de Lagrange pour la contrainte de l'égalité et ν_n comme les multiplicateurs de l'inégalité, la fonction Lagrangian peut être exprimé sous la forme:

$$\begin{aligned} L\left(e_n, \lambda, \nu_n\right) &= \\ &\frac{1}{N}\sum_{n=1}^{N} f_{BER}\left(h_n \cdot e_n\right) + \lambda\left(\sum_{n=1}^{N} e_n - E_{total}\right) + \nu_n\left(e_n - \overline{e}\right) \end{aligned} \tag{6}$$

Le problème d'optimisation ci-dessus impliquant à la fois les contraintes de l'égalité et l'inégalité, peut être résolus en utilisant les conditions KKT [28], dont la solution et celle du problème d'optimisation souhaité. Solution des conditions KKT demande la différenciation de la fonction objective impliqué, qui est la fonction BER $f_{BER}\left(h_n \cdot e_n\right)$ dans notre cas. Pour un modulation M-PSK ou M-QAM, le fonction de probabilité d'erreur exacte est généralement exprimée au travers d'un *fonction-Q* [29]. Il ne serait pas donc possible de trouver une solution finale, si la fonction-Q est utilisée comme la fonction objective dans l'equation Lagrangian. Par conséquent, il est plus approprié d'utiliser une approximation de l'expression exacte de BER. Pour la modulation M-QAM, la fonction exacte de BER peut être estimé en utilisant une borne supérieure exponentielle [26]. Après la résolution du Lagrangian définis ci-dessus au travers des conditions KKT, l'allocation optimale d'énergie par rapport le BER globale et respectant la contrainte de l'energie-maximale est représentée par:

$$e_n = \begin{cases} 0 & h_n < \exp\left(\frac{-b\lambda_0}{2}\right) \\ \overline{e} & h_n \geq \exp\left(\frac{-b\lambda_0}{2}\right),\ h_n\overline{e} + \left(\frac{2}{b}\right) ln\left(\frac{1}{h_n}\right) \leq \lambda_0 \\ \frac{\lambda_0}{h_n} - \left(\frac{2}{b}\right)\left(\frac{1}{h_n}\right)\ln\left(\frac{1}{h_n}\right) & h_n \geq \exp\left(\frac{-b\lambda_0}{2}\right),\ h_n\overline{e} + \left(\frac{2}{b}\right) ln\left(\frac{1}{h_n}\right) > \lambda_0 \end{cases}$$

La répartition optimale d'énergie, optimisé par rapport à un objectif particulier (débit, BER etc.) ne respecte pas le contrainte d'énergie-maximale ($\overline{e}$) nécessairement sur tous les sous-porteuses. *Water-Filling itératif* (IWF) est proposé [20] dans la littérature comme une méthode d'allocation d'énergie respectant la contrainte d'énergie maximale sur toutes les sous-porteuses. En IWF, la routine de Water-Filling (WF) est re-exécuter un certain nombre de fois afin de parvenir à une allocation respectant la contrainte de l'energie-maximale. Cette application itératif de la routine de Water-Filling rend une complexité large au l'algorithme et nécessite donc une simplification pour que ça soit implémenter dans les systèmes a très haut débit. Nous avons proposé une méthode simplifiée sous la forme de l'algorithme *(*Iterative Surplus Redistribution) (ISR) qui réduire considérablement la complexité de la routine de IWF.

L'algorithme ISR est basé sur l'idée q'une fois une allocation optimale a été effectué et tous les sous-porteuses dépassant la limite de $\overline{e}$ sont de la procédure d'allocation pour la prochaine itération, au lieu de réallocation de nouveau l'énergie totale E_{new} (comme ce fait dans l'approche IWF), seul le sum d'énergie en surplus de $\overline{e}$ (E_{surp}) sur les sous-porteuses dépassant le $\overline{e}$ limite, est re-allouer. Cette action est répété jusqu'aucun d'entre les sous-porteuses dépasse le limite de $\overline{e}$. Donc, E_{surp} est distribué répétitivement telles qu'à chaque itération, sa montant diminue jusqu'à ce qu'il est complètement ré-attribuer avec la non violation de la limite de $\overline{e}$. Cette méthode évite l'exécution de la routine complète de Water-Filling dans chaque itération, ce qui réduit la complexité éventuelle de l'algorithme. À chaque itération, l'objectif est de trouver l'incrément dans la valeur de la constante initiale de WF (λ_0^{ini}), qui est égal à ($\epsilon = \lambda_0^{nouveau} - \lambda_0^{ini}$) et qui pourra répartir proprement le $E_{surplus}$ entre la sous-porteuses appropriés.

Conclusion

Dans ce chapitre, nous avons abordé le problème de la distribution optimale de l'énergie sur un système multi-porteuses pour une canal sélective en fréquence, de sorte que BER moyenne est minimisé avec les contraintes sur l'énergie totale, l'énergie maximale et la taille de constellation. Notre contribution principale est l'extension de développements théoriques en tenant compte la contrainte de l'énergie maximale et la proposition d'un algorithme pour l'allocation d'énergie respectant une contrainte d'énergie maximale. La complexité de notre algorithme a été comparée avec la solution classique de Water-Filling Itératif (IWF = Iterative Water-Filling). Notre algorithme a été trouvée nettement moins complexe que la solution IWF, par la comparaison de complexité théorique. Pour vérifier l'analyse de complexité théorique, nous avons comparé le temps de simulation des deux algorithmes pour les différents paramètres afin de vérifier la complexité réduite de notre algorithme.

Chapitre 4 - Optimisation des Codes LDPC Irréguliers

Introduction

En ce qui concerne l'adaptation dans la couche physique, les paramètres comme l'énergie de symbole [27], taille de constellation [5, 30], taux de codage [31] ou une combinaison de ces paramètres [32], [33], est modifiées en réponse d'un canal sélectif en temps et/ou fréquence. Dans les deux précédents chapitres nous avons traité les algorithmes d'adaptation concernant la paramètres du taille de constellation /type de modulation et l'énergie du symbole. Dans ce chapitre, nous allons jouer avec l'adaptation du paramètre de taux de codage canal en modifiant le montant de la redondance par rapport l'état de canal.

Les codes LDPC (Low-Density Parity-Check) ont été inventés par R.G. Gallager [34] en 1962 et ils sont un type de codes en bloc linéaires. Gallager a découvert un algorithme de décodage itératif qu'il a appliqué à cette nouvelle classe de codes. Mais, les codes LDPC ont été ignorés pendant longtemps essentiellement en raison d'une grande complexité de calcul, surtout si la taille des codes est trés long. En 1993, C. Berrou a inventé les codes turbo [35] et leur algorithme de décodage itératif. Les performances remarquables observées avec les codes-turbo à rajeuni l'intérêt pour les techniques de décodage itératif et en 1995, MacKay et Neal [36] ont redécouvert les codes LDPC par la mise en place d'un lien entre leur algorithme itératif et l'algorithme de propagation de croyance [37].

Comme tous les codes linéaires, LDPC codes sont définis en termes des matrices de générateur (G) et de parité (H) avec les colonnes et de lignes représentant bits/noeuds de message et de parité, respectivement. Si le poids/degrés (nombres des '1') de tous les lignes de matrice de parité H est même ainsi que celle de tous les colonnes de H, le code LDPC est appelé un code LDPC régulière et autrement un code LDPC irréguliers. Si la distribution d'irrégularité d'un code LDPC irrégulière est bien choisie, les codes LDPC irréguliers montre les performances supérieures à code régulier. Pour une grande taille de bloc, ils atteindre les performances le plus proches de la capacité et parfois mieux même que le meilleur codes-turbo connue.

Dans son travail sur LDPC irrégulière, Luby [38] a donné la raison intuitif derrière les meilleures performances des codes LDPC irréguliers, ce qu'il a appelé comme le ***Wave***

Effet. Il en a déduit qu'il existe la raison de croire qu'une large distribution de degrés, au moins pour le noeuds de message, pourrait être utile. Les noeuds de message avec un degré élevé ont tendance à corriger rapidement leur valeur. Par conséquent, ces noeuds fournisses des informations bonnes aux noeuds de parité, qui par la suite fournisse une information meilleur aux noeuds-méssages avec un degré faible. Les constructions irrégulier ont donc le potentiel de conduire à un effet de vague, où les noeuds avec un degré élevé, ont tendance à se corriger en premier lieu, suivi par le correction des noeuds avec moins de degré.

État de l'Art

Le processus de conception de codes LDPC irréguliers est divisé en deux étapes de 1. Construction d'une famille des codes (Optimisation de distribution des degrés). 2. La construction d'un code particulier d'une famille des codes (Les 'connections' entre les différents noeuds). En littérature, tous les deux, les méthodes d'optimisation de profil d'irrégularité ainsi que les techniques pour la construction d'un code pour une distribution de degré donnée, sont présenté séparément et les deux se combinent pour définir la construction complète de codes irréguliers.

Deux algorithmes ont généralement employées pour concevoir une famille de codes LDPC irréguliers 1. L'algorithme d'évolution de densité (Density Evolution = DE) [39] et la technique de cartes d'EXIT (EXtrinsic Information Transfer) [40]. Richardson et al. [39] a montré la performance des codes irrégulière, crée avec l'algorithme d'évolution de densité, s'approche au capacité de canal. L'algorithme DE suit la fonction de densité de probabilité (pdf) des messages échangé entre les noeuds-message et noeuds-parité, avec l'hypothèse asymptotique (longueur infinie de code). Les techniques de l'évolution différentielle [41] et l'approximation gaussienne [42] sont deux techniques proposés dans le littérature, permettant la mise en oeuvre faisable de l'algorithme de DE.

Les méthodes pour la construction des codes LDPC pratiques (longueur court) sont également présentes dans la littérature. Parmi les techniques le plus simples sont des méthodes complètement aléatoire [43] ou semi-aléatoire [34]. Des approches déterministes, qui tendent à améliorer la performance du graphe, inclus le méthode de 'Bit-Filling' [44] et d'allocation progressif des 'edges' (PEG) [45]. Ainsi, il a été montré dans [46] , que tous les 'cycles' de longueur court n'ont pas le même impact deteriotive sur la performance et, par conséquent, une méthode d'éviter les cycles sélectif a été proposé.

Optimisation des Codes LDPC Irréguliers avec 'Wave-Quantization'

Le décodage de codes LDPC, réguliers ou irréguliers, est généralement fait avec la méthode classique de 'Message Passing' ou la propagation de croyance (Belief propagation = BP) algorithme ou la probabilité/vraisemblance des bits sont échangées comme message dans le décodage itératif. Puisque, une analyse mathématique de décodage probabiliste pour un nombre d'itérations est difficile, Gallager [47] a proposé une borne sur la performance d'un décodage probabiliste au travers de décodage-hard (Bit-Flipping = BF) [48] qui sont connu sous le nom de Gallager A et Gallager B algorithmes de décodage, ainsi. Luby [38] a ensuite prologé l'analyse de décodage-hard pour le cas des codes irrégulière. Ensuite, [49] a proposé l'usage de décodage-hard basé sur la décision de majorité (Majority-Based =MB) et qui a été employé pour notre analyse de quantification de vague.

L'idée principale derrière notre proposition est l'utilisation de la phénomène de l'effet de vague (Wave-Effect) qui existe dans les codes LDPC irréguliers et qui est définir au-dessus. Luby et al. a introduit cette phénomène en cherchant la raison intuitif derrière les meilleures performances de codes irréguliers que les codes réguliers. Notre but est d'aller plus loin et de quantifier cet effet de vague, de sorte qu'il peut éventuellement être utilisé comme un moyen pour l'optimisation et la construction des codes LDPC irrégulière de longueur finie. Il s'agit de quantifier le changement (l'incrément/décrément) dans la probabilité d'erreur (p_i) d'une noeud-message à cause de l'addition / soustraction d'un noeud de parité attaché. Il a été remarqués que la méthode classiques de calcul de changement de la probabilité d'érreur en utilisant un décodeur-hard pour des codes LDPC irrégulier, en raison de calcul prohibitif, ne parviennent pas à nous fournir la quantification exacte de l'effet de 'vague'. Nous avons présenté une méthode fondée sur la somme des produits de combinaisons (Sum of Product of Combinations = SPC), par l'aide duquel, la quantification de l'effet du vague devient faisable. Une comparaison a été faite entre le temps d'exécution des deux méthodes de calcul et notre méthode a été trouvé nettement moins complexe, et faisable pour implémentation même pour le cas lorsque la méthode classique de calcul devient irréalisable.

Enfin, nous proposons un algorithme où *irrégularité* est ajouté progressivement au un code régulier. La performance du code est mesurée et quantifiés pour des différentes possibilités de l'irrégularité, et finalement l'irrégularité est rajoutée ou la performance du code éventuelle est meilleure. La **performance** pour différentes options d'irrégularité est mesurée/quantifiée est comparé par le quantification de la changement dans le probabilité d'erreur, comme expliqué précédemment. Si la probabilité d'érreur d'un noeud-message 'n' après itérations 'i' est représenté par p_i^n, est qui est composé de facteur x_i et y_i, on peut représenter/quantifier le changement (Δ) dans le probabilité d'érreur originale (p_0) comme suivant

$$x_i = \left[\frac{1 + \prod_{k'=1}^{k}(1 - 2.p_{k'})}{2}\right] \qquad y_i = \left[\frac{1 - \prod_{k'=1}^{k}(1 - 2.p_{k'})}{2}\right] \tag{7}$$

$$x_i = \left[\frac{1 + \prod_{k'=1}^{k}(1 - 2.(p_0 + \Delta_{k'}))}{2}\right] \qquad y_i = \left[\frac{1 - \prod_{k'=1}^{k}(1 - 2.(p_0 + \Delta_{k'}))}{2}\right] \tag{8}$$

Qui est égal à

$$x_i = \underbrace{\left[\frac{1 + (1 - 2.p_0)^k}{2}\right]}_{\mathbf{x}} + \underbrace{\sum_{k'=1}^{k}(1 - 2.p_0)^{k-k'}.\prod_{l'=1}^{k'}(-2\Delta_{l'})}_{\Delta_{neigh}} \tag{9}$$

And

$$y_i = \underbrace{\left[\frac{1 - (1 - 2.p_0)^k}{2}\right]}_{\mathbf{y}} + \underbrace{\sum_{k'=1}^{k}(1 - 2.p_0)^{k-k'}.\prod_{l'=1}^{k'}(2\Delta_{l'})}_{\Delta_{neigh}} \tag{10}$$

Conclusion

Ce chapitre a proposé un nouveau méthode (Greedy Irregularity Construction =GIC) de construction des codes LDPC irrégulier, en rajoutant l'irrégularité dans un code régulier, avec une manière progressive. Cette approche est basée sur la quantification du l'effet de vague qui est un phénomène bien-connue pour des codes irrégulier. La méthodologie de quantification repose sur l'utilisation d'analyse probabiliste d'un décodeur-hard basé sur la majorité (Majority-Based = MB). Le méthode classique de calcul, pour analyse probabiliste de décodeur-hard pour des codes irrégulière, a une large complexité et donc n'est pas faisable pour utilisation dans notre cas. Nous avons proposé une méthode de calcul, basé sur la somme de produit des combinaisons (Sum of Product of Combinations = SPC), qui est beaucoup plus simple et qui rendre son utilisation faisable pour notre cas. À notre connaissance, il s'agit de la première tentative pour réaliser la 'quantification' de l'effet de vague pour l'optimisation des codes irréguliers. Nous pensons que la poursuite de ce travail pour le différent type de décodage et des canaux, peut aboutir à la production des résultats encore plus significatifs.

Chapter 5 - Algorithme-Architecture Co-Optimisation

Introduction et l'État de l'Art

Le temps d'exécution d'un algorithme dépend de sa complexité algorithmique, ainsi que sur la plateforme d'implémentation sur laquelle l'algorithme est implémenté. L'utilisation de la meilleure (plus rapide) plateforme d'implémentation n'est pas toujours la solution surtout pour les cas où la contrainte sur le temps d'exécution de l'algorithme est variable en temps, comme la notre c'est-à-dire un système de communication sans-fil. La souplesse/flexibilité offrir par les systèmes re-configurable semble bien à répondre aux besoins variables de calcul. Ces dernières années ont vu un intérêt accru dans l'utilisation des plateformes re-configurable pour les applications de traitement du signal [50–52]. Sa motivation provient de plusieurs directions dont la popularité des radios logiciel/cognitives [53, 54] est une des motivations principales de l'intérêt dans les plateformes re-configurable pour le traitement du signal. Parmi les caractéristiques souhaitées de radio logiciel, la mode multifonctionnalité rend la possibilité aux fonctionnement avec plusieurs normes et protocoles de communication aux moments/l'endroits différents. Puisque des exigences de calcul des différentes normes de communications varient de manière significative, l'utilisation des plateformes re-configurables dans ce contexte conduit à une économie d'énergie et de ressources importante.

Notre but était d'exploitées l'architecture re-configurable dans le contexte des algorithmes d'adaptation. Ca peut servir pour réduire encore le temps d'exécution des algorithmes d'adaptation, qui peut conduire à des gains de performance ou même permettre l'utilisation des algorithmes d'adaptation dans certains cas où leur utilisation n'aurait pas été possible autrement. Dans se chapitre, on explore le contexte où les ressources de l'architecture / processeur sont gérées à temps réel et un lien entre les paramètres théoriques (temps de cohérence etc.) et les ressources matérielles (cache, ALU etc.) est créé pour accroître la performance globale du système.

Dans la travail de Zhang [55], un architecture MP SoC (Multiprocesseur System on Chip) a été utilisée comme plateforme de mise en oeuvre pour un radio cognitive OFDM. Dans [56], l'implémentation des divers algorithmes de bande de base dans le protocole d'Hiperlan, a été étudiée sur une architecture flexible, et les ressources (cache, ALUs etc.) ont été répartis par rapport les besoins des algorithmes. De même, [57] enquête sur l'utilisation des FPGAs pour la mise en place d'un système OFDM.

Concernant la mise en oeuvre des algorithmes de 'bit-loading', Cudnoch [58] a donné un analyse d'implémentation d'un algorithme de 'bit-loading', destinés à un système sans fil multi-porteuses. La performance de la mise en oeuvre a été étudiée en termes de débit sur diverses conditions de canal de la norme IEEE 802.11a. Dans [59], la faisabilité de mise en oeuvre des algorithmes d'adaptation sur une architecture configurable, a été analysé dans le contexte d'un système WiMax. Dans [60], l'implémentation des algorithmes adaptif est optimisée pour des systèmes mobiles OFDMA à large bande et leur faisabilité est évaluée dans le cadre de spécifications WiMax. De même, en [59] la possibilité de la re-configuration dynamique partielle d'un FPGA est analysé pour un système OFDM avec les spécifications WiMax. Toutefois, aucun effort n'a pas été fait pour allouer dynamiquement les ressources d'architecture par rapport les contraintes de la canal (temps de cohérence etc.), qui est le but de ce chapitre.

Algorithme-Architecture Co-Optimisation pour Systèmes Mobile Adaptatives

Les canaux sans fil, en raison de la présence d'interférence inter-symbole, sont sélectifs dans le temps et fréquence. Si en plus, un utilisateur est en mobilité avec une vitesse large, un doppler importante est présent et le temps de cohérence est largement réduit. Pour le bonne fonctionnement des systèmes adaptatifs dans ce cas, l'adaptation doit se faire très rapidement correspondant avec les variations de la canal. A pars des réseaux où les utilisateurs ont une faible mobilité (WiFi), nouveaux systèmes e.g. WiMAX, cible des scénarios où la vitesse des utilisateurs peut varier de la vitesse des piétons à la vitesse des trains à grande vitesse (200 Km/h). Cela renforce la nécessité d'accroître la temps de convergence des algorithmes d'adaptation/'bit-loading' ou le développement des méthodes qui répondent aux diverses contraintes du temps sur différents canaux.

Notre première réponse à cette problématique était d'améliorer le temps de convergence des algorithmes de 'bit-loading' d'un point de vue algorithmiques, comme expliqués dans les chapitres précédents. Par contre, pour le cas où même le meilleur type d'un algorithme ne répond pas au besoin des contraintes (temps de cohérence etc.) d'un canal qui se-varient en temps, d'autres méthodes doivent être examinées pour améliorer la temps de convergence de l'algorithme. Une telle possibilité est d'explorer les paramètres optimaux d'une architecture par rapport à un algorithme et, par conséquent, obtenir une réduction dans le temps d'exécution d'un algorithme par l'optimisation des paramètres de l'architecture, sur laquelle l'algorithme est implémenté. Avec l'utilisation croissante des plateforme re-configurable (FPGA, processeurs programmables etc.), l'idée d'optimiser les ressources d'une architecture en temp-réel, est devenu réalisable.

Dans notre méthodologie, nous proposons que dans un premier temps, sur la base des paramètres du système (nombre de sous-porteuses, nombre de bits à attribuer, la temps du cohérence etc.), une base de données pour les performances des différentes configurations (paramètres) de l'architecture du processeur est créé pour tous les types d'algorithmes disponibles. Nous avons utilisé l'outil de SimpleScalar [61] et un architecture superscalar, pour calculer le nombre exacte des cycles d'exécution pour différent algorithmes. L'algorithme génétique est utilisé comme un outil d'optimisation, où la fonction objective est le temps d'exécution d'un algorithme sur une architecture particulière. Puisque l'exécution et l'évaluation d'un grand nombre de configurations n'est pas possible en temp-réel, une base de données est créé initialement, reliant les différentes configurations aux leurs temps d'exécution correspondante. Cela donne la possibilité de choisir au temps réel la meilleure configuration de processeur pour un algorithme donnée et correspondant à la contrainte de temps de la cohérence du canal à cet instant. À cet égard, il est important de mentionner qu'avec l'amélioration de la technologie des DSP programmables, il sera possible même de trouver le temps d'exécution correspondants aux différentes configurations de processeur au temps réel, au lieu de le tous faire initialement.

Ensuite, un algorithme avec des paramètres d'architecture par défaut, est attribué pour le processus d'allocation, et vérifier s'il répond à la contrainte de temps de cohérence, sinon, un autre algorithme est choisi. Une fois que toutes les options, de types des algorithmes, ont été épuisées, la meilleure configuration répondant à la contrainte de temps de la cohérence est sélectionnée et appliquée au processus d'allocations des bits. Le critère de temps d'exécution de l'algorithme est seulement examiné dans ce travail, qui est la plus pertinente par rapport des contraintes retard temporel (temps du cohérence) de canal. Toutefois, la même méthode peut être étendue à d'autres critères d'optimisation telles que la consommation d'énergie et de surface de silicium. À cet égard, l'utilisation de l'algorithme génétique mono objectif dans notre proposition, devra être remplacé par l'algorithme génétique multi objectifs, qui prend en compte plusieurs objectifs et contraintes en même temps.

Conclusion

Dans ce travail, nous avons exploré les options où le temps de convergence d'un algorithme adaptif peut être améliorée par l'optimisation des ressources architectural en temps réel, correspondant aux contraintes temporel (temps de cohérence etc.) d'un canal donnée. On a observé que l'optimisation des ressources architecturales permettre le fonctionnement d'algorithme d'adaptation aux plus hautes vitesses de la mobilité. L'utilisation de l'algorithme génétique permet d'éviter le grand temps de simulation qui serait par ailleurs nécessaire pour faire une optimisation pour un espace de conception correspondant aux paramètres d'un processeur. Enfin, la méthodologie proposée, d'exploration conjointe des paramètres des architectures et algorithmes en temps réels, nous permettre le bon fonctionnement des algorithmes d'adaptation dans les cas avec plus dur contraintes sur le temps de cohérence, où leur fonctionnement n'aurait pas été possible autrement.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Adaptive Communication Systems - Resource Allocation

Data transmission has become an integral and ubiquitous component of today's world. Everyday actions, such as using a bank machine, making a phone call, watching television, and doing grocery shopping, all involve some sort of data transmission that makes these actions more convenient, cost effective, or feasible. This data transmission can be performed over a wireline infrastructure, a wireless network, or a combination of the two infrastructures. A consequence of this increased integration of data transmission in our day-to-day life is the demand for more throughput. As the level of integration increases and more people are connected, the amount of data generated grows. Therefore, the data rates of the transmission systems must increase to keep up with the increase in information.

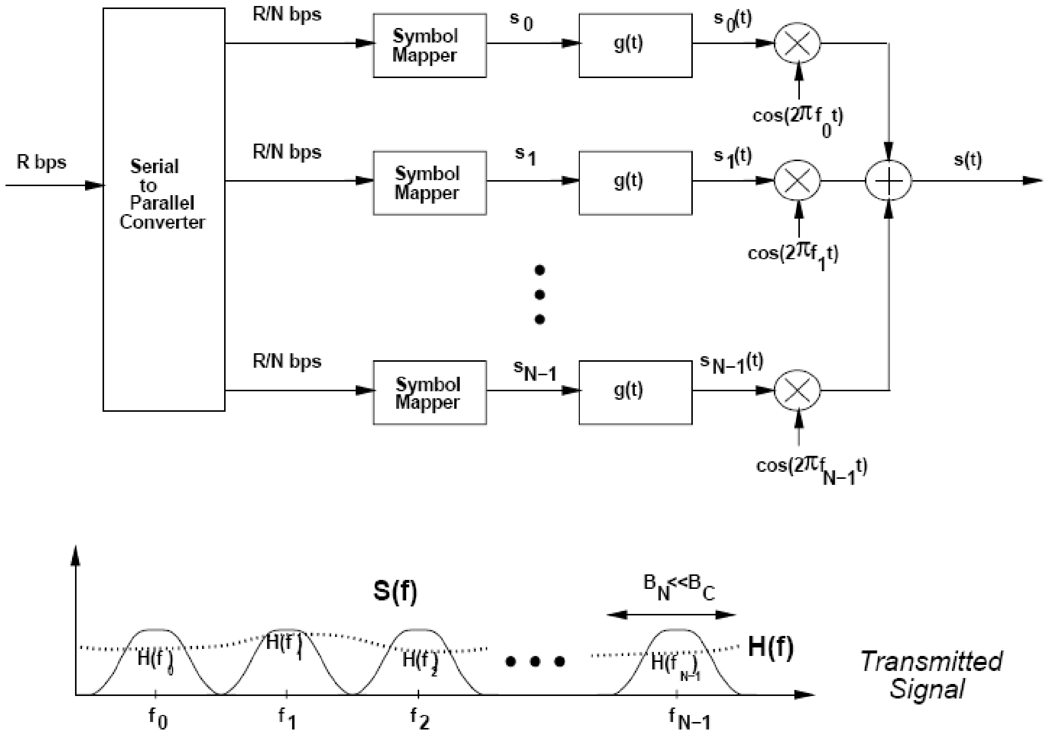
Although the throughput supported by wireline networks are enormous, due to fiber optics and other technologies, the base station/mobile user interfaces of wireless networks are still trying to keep up with the demand for more throughput. Moreover, there are several significant restrictions when wireless modems transmit at high data rates. The first is bandwidth usage. Since the spectrum that wireless systems use to transmit data is regulated by government agencies, such as the Federal Communications Commission (FCC) in the United States, each operator of a wireless data transmission infrastructure must abide by the established guidelines. This is done in order to avoid interference between different wireless operators. Therefore, the rate is constrained by the maximum bandwidth allocated to the operator. The second constraint is the channel environment which the data transmission system is operating through. At higher data rates, the amount of distortion introduced to the transmission becomes more pronounced, making it more difficult to compensate at the receiver.

Since the regulatory requirements of the spectrum cannot be modified or changed, researchers are investigating techniques for enhancing the performance of digital transmission systems operating in various channel conditions (e.g., additive white Gaussian noise, multipath fading, impulse noise). Adaptive resource allocation [1–3] is one such method to enhance the system performance where different system parameters (modulation size, coding rate, power etc.) are modified according to the Channel State Information. This resource allocation can be performed at multiple layers but our concern will be considering the adaptation possibilities at physical layer. It was shown that adaptive power and rate M-QAM system can give a power gain of almost 20dB relative to non-adaptive transmission [4]

1.2 Resource Allocation in Single and Multicarrier Physical Layer

Multicarrier modulation (MCM) [5] has almost revolutionized the physical layer technology for communication systems, both wireline and wireless, for the past two decades. It is the technology of choice for digital subscriber lines (DSL) systems [6] along with Wireless Local Area Networks (WLAN) , Multiband-Orthogonal Frequency Division Multiplexing (MB-OFDM) based Wireless Personal Area Networks (WPAN) and as well the emerging WiMax (IEEE 802.16) systems for broadband wireless communications. MCM operates according to a *divide-and-conquer* approach: by transmitting the data across the channel at a lower data rate in several frequency subcarriers which eventually makes the process of distortion compensation simpler by treating each subcarrier separately. From a time-domain perspective, this translates the wideband transmission system into a collection of parallel narrowband transmission systems each operating at a lower data rate [19]. From the frequency-domain perspective, MCM transforms the frequency-selective channel, i.e., non-flat spectrum across the frequency band of interest, into a collection of approximately flat subchannels which the data gets transmitted over in parallel. Thus, MCM has be-

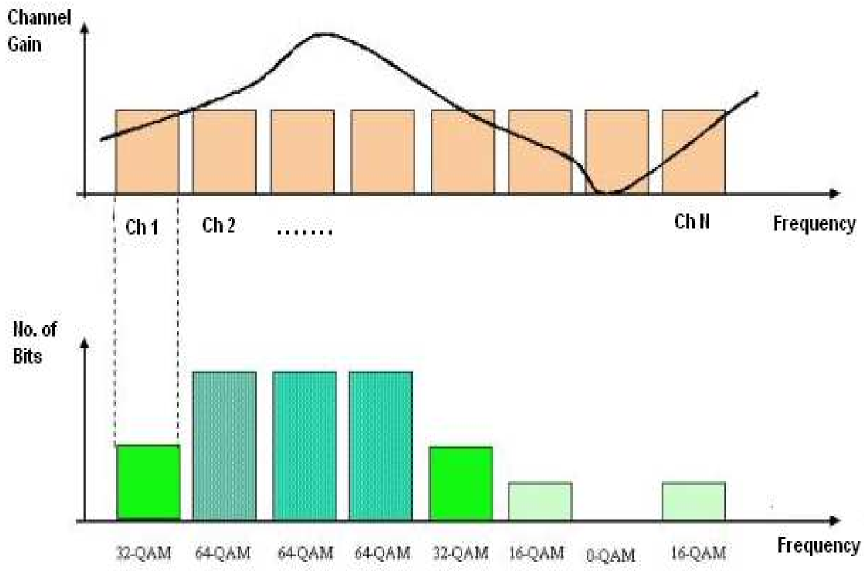
Figure 1.1 Multi-Carrier Modulation



come the technology of choice to combat the frequency-selective fading channel. A basic transceiver structure of MCM is show in figure 1.1 along with the resultant division of the entire operating bandwidth spectrum into a large number of orthogonal subcarriers.

Although the modulation and demodulation stages of an MCM system are usually more complex relative to a single carrier system, MCM systems possess a number of advantages due to the division of the used spectrum into a large number of subcarriers. Since the channel usually does not have a flat frequency response, it is easier to compensate for the channel distortion on a per-subcarrier basis rather than on the entire received signal. Moreover, since the channel distortion may not be equivalent for all subcarriers, adapting the transmission parameters per subcarrier (i.e., signal constellation and transmit power levels) would allow for increased throughput while guaranteeing a prescribed error performance. This phenomenon of transmission parameters adaptation per subcarrier can be represented by figure 1.2, where the parameter of constellation-size (modulation type) is varied on each subcarrier based upon the channel conditions (Signal to Noise Ratio (SNR)) on each subcarrier. This adaptive transmission, depending upon the parameter that is being adapted/optimized, is known in literature by a number of terminologies namely Link-Adaptation, Adaptive-Modulation (adapting the modulation

Figure 1.2 Adapting Modulation-Type on Different Subcarriers of a MultiCarrier System



type or constellation-size), Bit-Loading (Adaptive Modulation when constellation size is strictly discrete), Power Allocation (Adaptation of the power-level attributed to each subcarrier), Adaptive Coding (Adaptation of the amount of redundancy added) etc.

A thorough comparison between single carrier and multicarrier systems was performed by Saltzberg using a number of criteria, as summarized in table 1.1 [62]. There is little difference in performance between single carrier and multicarrier systems since the latter can be interpreted as a linear reversible transformation of the former. However, there are a number of practical differences. For instance, multicarrier systems can perform adaptive bit loading per subcarrier/frequency based upon the channel response over that particular frequency in a straight-forward fashion, which eventually enhances system performance, either in terms of maximizing overall system throughput or by increasing error robustness of the system. On the other hand, multicarrier systems are more sensitive to the effects of narrowband noise, amplitude clipping, timing jitter, and delay. With respect to the computational complexity, FFT-based multicarrier systems employing frequency-domain single-tap subcarrier equalizers usually use fewer multiplications and additions per unit time relative to single carrier systems, which require lengthy equalizers to eliminate the distortion introduced by the channel. As a result, multicarrier systems have fewer computations per unit time. Similarly other advantages as the ability to efficiently capture multipath energy, simplified transceiver/equalizer architecture, enhanced capability to exploit frequency diversity, increased interference mitigation

Table 1.1: Comparison between Single-Carrier and Multi-Carrier Systems

Issue	Single-Carrier	Multi-Carrier	Same
Performance in Gaussian Noise			Yes
Sensitivity to Impulse Noise		Yes	
Sensitivity to Narrowband Noise	Yes		
Sensitivity to Clipping	Yes		
Sensitivity to Timing Jitter	Yes		
Latency (Delay)	Yes		
Need for Echo Cancellation	Yes		
Complexity of Algorithm	Yes		
High Cost and Power Consumption	Yes		
Sensitivity to Impulse Noise (Analog)	Yes		
Adaptability of Bit-Rate		Yes	

capability, inherent robustness to Inter-Symbol-Interference (ISI) and spectral flexibility to avoid low quality sub-bands and to cope with local regulations, all have made the Multicarrier communications phenomenon as the technology of choice for the majority of the state-of-the-art wireline (DSL, Powerline) and Wireless (WiFi, DAB/DVB, WiMax, UWB etc.) communication systems.

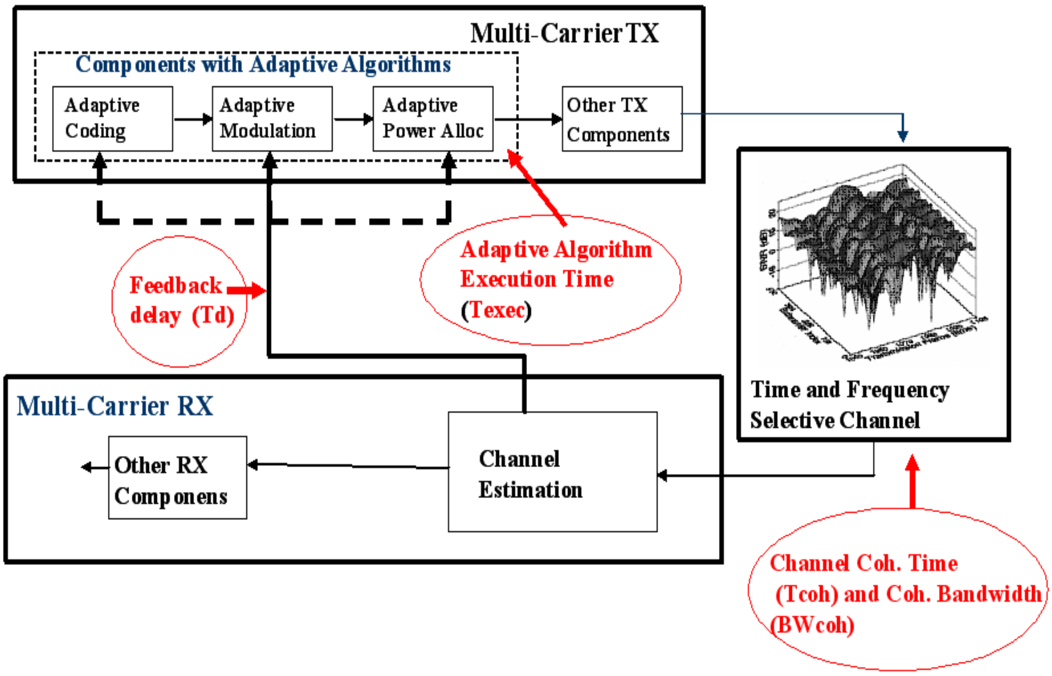
1.3 Problem Definition

Despite all the above cited advantages of multicarrier modulation, many conventional WLAN systems do not fully exploit its potential, unlike DSL modems. Rather, conventional WLAN systems employing MCM use the same operating parameters across all subcarriers, including modulation scheme, coding rate, and transmit power level. However, the effects of the channel may vary on a subcarrier basis, and thus the overall error probability of the system is dominated by the error probabilities of the subcarriers with the worst performance [19]. For instance, systems that try to keep the error rate low usually transmit with the smallest subcarrier signal constellation possible. Equivalently, systems that require a high throughput have error probabilities dominated by the largest subcarrier error probability. Thus, to enhance system performance, the impact of these poorly-performing subcarriers should be mitigated. This is the rationale behind adaptive resource allocation/loading(when resource is discrete e.g. constellation-size) for MCM systems.

Resources/Parameters that are commonly varied/adapted on different subcarriers are modulation size and power and rarely coding rate. Since the channel which the data transmission system is operating in is usually frequency-selective, each subcarrier will have a different signal-to-noise ratio (SNR). Thus, tailoring the operating parameters on per-subcarrier basis has shown to largely improve performance [63]. A complete model for an adaptive system is shown in figure 1.3, where the feedback path from the receiver to the transmitter is used to transmit the information on the conditions of the channel (Channel State Information- CSI), based on which it has received the current symbols and based on which the transmitter must adapt the transmission of the next frame, assuming that the channel-conditions remain constant on a number of frames. The techniques for loading originated from other areas, including financial analysis and quantizer design [64]. However, since the advent of the use of adaptive resource allocation in multicarrier systems, a large number of loading algorithms, using all sorts of optimization techniques, have been developed and implemented for several data transmission systems, including DSL modems [7].

Despite these efforts, a number of issues remain unresolved or require better solutions. For example, although for more than a decade a huge amount of research has been dedicated to reduce the computational complexity of the per-subcarrier (fine-granular) based adaptive algorithms. However, in practical wireless systems e.g. WiFi, WiMax, until now the adaptive resource allocation is performed such that the transmission parameters are varied at different time-instants but for a particular time-instant the transmission parameter is same over all the subcarriers or in other words whenever a parameter (modulation-size, power, redundancy etc.) is adapted, the same parameter is allocated to all of the subcarriers (Coarse-granular). Such coarse-granular adaptation leads to performance improvement but not to the extent where the fine-granular adaptation is performed. Similarly other questions like what form of adaptation is best or what com-

Figure 1.3 Adaptive Multicarrier System



bination of different forms (adaptive modulation, adaptive power-allocation, adaptive coding etc.). Another important question which this thesis seeks to answer is what benefits can be extracted from the hardware implementation aspects of the link-adaptation algorithms especially with the availability of *adaptivity* in the implementation platforms as well (FPGAs, configurable processors etc.)

1.4 Our Contributions and Thesis Presentation

However, the main objective of this research was/is to reduce the complexity of the existing adaptive algorithms for multicarrier systems, both, from a theoretical/algorithmic perspective as well as taking advantage from the adaptive nature of the state-of-the-art underlying implementation platforms

Therefore, to reach this main objective, several sub-objectives have been achieved in this dissertation, namely:

- The first insight into the inherent pattern, which exists in the optimal bit-allocation, and based upon this pattern the design of a Novel Optimal Discrete Bit Loading algorithm which has a complexity significantly lower than existing algorithms for discrete bit-loading

- Theoretical developments for an optimal power-allocation including the Peak-Power Constraint (explained in Chapter3) and the design of a novel algorithm for Peak-Power Constrained Optimal Power Allocation which involves a complexity significantly lower than the classical method of *Iterative- Waterfilling*, which is conventionally employed for such constrained power allocations.

- The proposal of the novel idea of *Wave-Quantization* in Irregular-LDPC Codes, the theoretical developments in this regard leading to a simplistic calculation method of the *elite-effect* in Irregular LDPC-codes which is not possible with classical methods. Finally a proposal of the design and optimization of Finite-Length Irregular LDPC Codes based on this *Wave-Quantization* methodology.

- The optimization of a flexible implementation architecture for the algorithms of link-adaptation, where a methodology to tune the hardware resources at run-time based upon the needs of the channel and system is proposed. This results in the use of the link-adaptation algorithms at those ranges of doppler-frequencies/user-mobility, which are not possible to operate upon, otherwise.

Chapter 2 introduces the reader to the proposed bit loading algorithm. An extensive state-of-the-art on the loading algorithms is presented along with the basic concepts of the theoretical foundations of the algorithm. The algorithm's performance in terms of complexity is done with respect to other proposed algorithms, both theoretically and in terms of exact number of execution cycles over a Superscalar architecture.

Chapter 3 introduces the reader to the proposed power loading algorithm. A state-of-the-art on different power optimization methods with different goals and constraints is given. Our concerned problem is formulated for Bit Error Rate (BER) minimization, for a given throughput and with a peak-power constraint using the lagrange optimization method. Then, *Iterative Surplus Re-distribution (ISR)* based power-allocation algorithm is presented as a simplification to the classical *Iterative-Waterfilling* algorithm, and its complexity analysis is presented.

Chapter 4 presents our developments regarding the development of a simplified methodology to update the irregularity profile of an irregular LDPC code based upon the channel-gains. State-of-the-art on irregularity profile optimization is presented along with the our proposed idea of Wave-Effect Quantization. Corresponding developments to make it computationally feasible and finally the *Progressive-Irregularity-Construction* (PIC) based methodology for the design and optimization of finite-length irregular codes.

Chapter 5 will present our work related to the implementation aspects of different algorithms. The set-up used for performing an exhaustive search over a configurable processor architecture for the bit-loading algorithm is presented. Then, genetic algorithms were employed to converge faster toward the optimal hardware configuration, for a given algorithm. A methodology to link the run-time tuning of the processor architecture parameters based upon the time-varying channel needs is finally proposed, to enhance the operating doppler-range of a wireless system.

Finally, in chapter 6, the research achievements of this work are outlined, conclusions are drawn from this research and perspectives of interest regarding this work are presented which may be taken as research-directions in the future.

Chapter 2

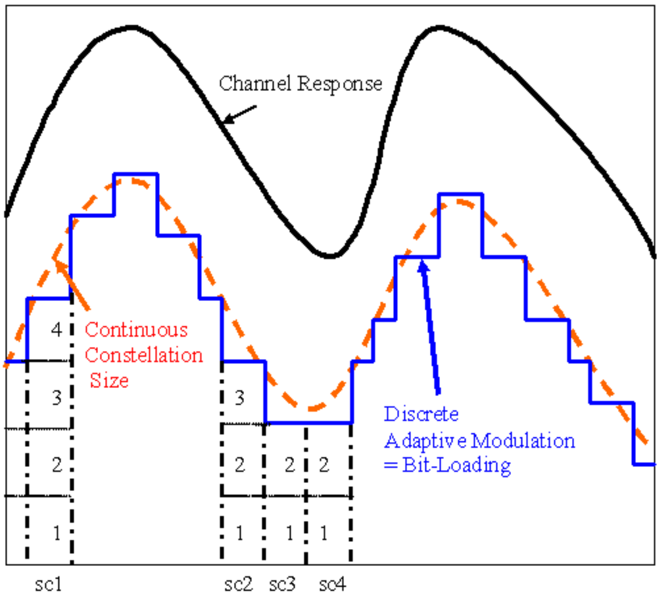
Bit-Loading Algorithm for Multicarrier Systems

2.1 Introduction

We explained in chapter 1 that an *adaptive* selection of system parameters (modulation, power, coding etc.) with respect to the channel response results in an overall improvement in system performance. This *adaptation* can be done in the dimension of time (corresponding to different channel conditions at different time-slots), in the dimension of frequency (corresponding to channel gains at different frequencies), in the dimension of space (corresponding to channel conditions at different branches of a multi-antenna system) or any given combination of the above three possibilities. Adaptation in the dimension of frequency is greatly facilitated by means of a Multi-carrier system as we explained in the previous chapter because of the presence of a large number of subcarriers at unique frequencies, which eventually allows fine-tuning of system parameters at different frequencies. **Adaptive Modulation** generally refers to the adaptation policy where modulation type/constellation size (BPSK, QPSK, QAM-16 etc.) is allowed to be varied with respect to the channel conditions, i.e. the better the channel, the higher the modulation size. In multicarrier systems, since a different modulation size may be allowed for each sub-carrier, the phenomenon of adaptive modulation with respect to each sub-carrier is termed as **bit-loading** when constellation-size is only allowed discrete number of bits as shown in figure 2.1

Since the multicarrier transmission technique was first employed in the wireline DSL

Figure 2.1 Adaptive Modulation : Continuous and Discrete (Bit-Loading) Constellation Size



technology, much of the literature regarding the 'bit-loading' algorithm comes under the paradigm of Discrete Multitone (DMT), which is the variant of multicarrier transmission with bit-loading applied to the DSL technology. The task of optimizing the bit-profile i.e. how many discrete bits should be allocated to each subcarrier for a given channel condition and a given number number of sub-carriers has been of great research interest in the past under the title of **Discrete Bit-Loading Problem**.It has been found that the optimized bit-profile for the same channel and sub-carriers is different for different goals (maximizing the system throughput, margin, etc) and constraints (maximum available power, maximum no. of bits to be allocated, peak-power constraint etc.) associated with the optimization problem. Whereas a large number of optimization techniques from convex optimization to combinatorial methods (as will be indicated in the next state-of-the-art section), the aim has been to achieve the optimal-profile with minimal involved complexity. Complexity of the bit-loading optimization problem is of utmost importance because of the fact that in the fast-varying wireless channel, the optimization process would be required to be re-performed each time channel conditions are varied and hence be required to finish under a hard time-constraint. Hence, the basic aim of this chapter is to introduce to the reader the bit-loading problem in detail, the existing solutions in this regard and our proposed 3-dB subgrouping based bit-loading method which claims to achieve the optimal profile with complexity lesser than any other available algorithm.

2.2 Types of the Bit-Loading Problem

As discussed above, the objective of an allocation algorithm can be variable: maximizing the throughput, minimizing the BER or minimizing the total power etc., depending upon the need of the corresponding system. Based upon the objective, there are two typical loading problems which are popular in the literature: rate maximization and margin maximization (= power minimization) [8]. Each problem has the following optimization parameters. We define N as the total no. of subcarriers and σ_n^2, $|H_n|^2$, e_n, b_n and p_n^{err} as the noise power, channel gain, allocated energy, allocated no. of bits and probability of error, all with respect to subcarrier n, respectively. Based upon these notations the two major classes of Bit-Loading problem are listed below.

2.2.1 Rate Maximization

Maximization of the data rate ($R = B/T_{sym}$) for a set of parallel subchannels when the symbol rate $1/T_{sym}$ is fixed, requires maximization of the achievable total no. of bits ($B = \sum_{n=1}^{N} b_n$) over all the subcarriers. Hence, the largest number of bits that can be transmitted over a parallel set of subchannels must maximize the sum of the bits transmitted over each of the subcarriers given a constraint on the total available energy E. Thus the Rate Maximization problem can be formally expressed as

$$max \quad B = \sum_{n=1}^{N} b_n \ \ \forall\, b_n \in Z^+; \tag{2.1}$$

such that

$$E = \textstyle\sum_{n=1}^{N} e_n \qquad \forall e_n \in \ \Re^+$$
$$p_n^{err} \leq p_{target}^{err}$$

The problem can be re-stated as how much energy (e_i) (corresponding to a discrete no. of bits) must be allocated to each of the subcarriers such that the resultant throughput is maximized and the sum of the energies in all the subcarriers is equal to the total available energy. ‘Water-filling’ [9] is the classical solution of the above mentioned energy-constrained throughput-maximization problem, when energy (and the corresponding modulation-size) is allowed to have continuous values. The term **water-filling** arises from the representation of the curve of inverse channel gains being considered as a bowl into which water (energy) is poured, filling the bowl until there is no more energy to use

Figure 2.2 Waterfilling phenomenon in continuous domain

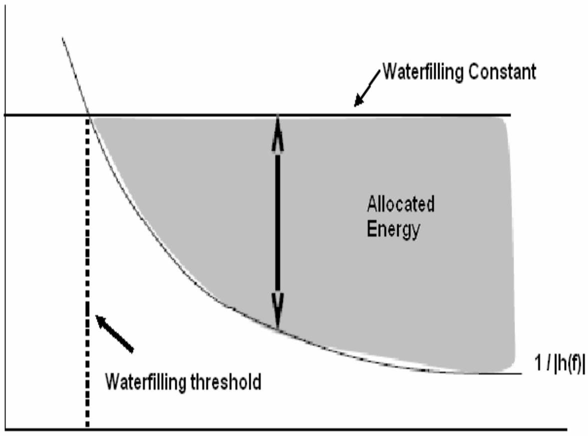
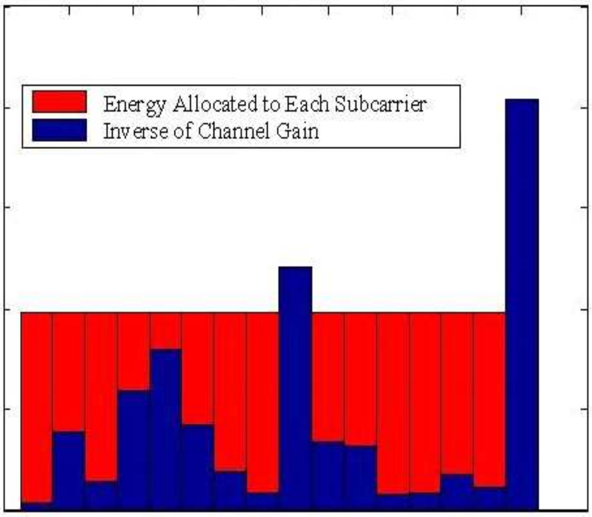


Figure 2.3 Discretized Waterfilling



as shown in figure 2.2. The water will rise to a constant flat level in the bowl. The amount of water/energy in any subchannel is the depth of the water at the corresponding point in the bowl (which hence indicates that the higher the channel gain, the more the energy allocated to it).The water-filling solution is unique because the function being minimized is convex, so there is a unique optimum energy distribution (and corresponding set of subchannel data rates) for each ISI channel with multi-channel modulation.

With the constraint of discrete no. of bits over each subcarrier, as in our case, the solution becomes a *discretized* version of the classical 'water-filling' solution. Figure 2.3 illustrates the discrete equivalent of water-filling phenomenon.

2.2.2 Margin Maximization

For many transmission systems, variable data-rate is not desirable. In this case, instead of maximizing the throughput, the best design will maximize the performance *margin* at a given fixed data-rate B^{target}. The margin, γ_m, for transmission on a (sub)channel with a given Signal to Noise Ratio (SNR), a given number of bits per dimension b', and a given coding-scheme target probability of error p^{err}_{target}, can be defined as the amount by which the SNR can be reduced (increased for negative margin in dB) and still maintain a probability of error at or below the target p^{err}_{target}. The margin maximization problem can be formally stated as

$$max \quad \gamma_m \tag{2.2}$$

such that

$$\begin{gathered} \textstyle\sum_{n=1}^{N} b_n = B^{target} \qquad b_n \in Z^+ \\ p_i^{err} \leq p^{err}_{target} \qquad E = \textstyle\sum_{n=1}^{N} e_n \quad e_n \in \Re^+ \end{gathered}$$

This problem can be simplified by realizing that performance margin is simply a scaling of the allocated energy in each subchannel by a constant amount (exploiting the fact that initially the probability of error in that subchannel is far below the required p^{err}_{target}). The minimum total energy allocation needed to meet the rate target with zero margin ($p_n^{err} = p^{err}_{target}$) can be found, and the resulting energies can be scaled, such that the total energy budget is used. Thus, to maximize fixed-rate margin, the designer equivalently minimizes the total energy (which is also called energy-minimizing loading) and which can be formally stated as

$$min \quad E_{total} = \sum_{n=1}^{N} e_n \;\; \forall\, e_n \in \Re^+; \tag{2.3}$$

such that

$$\begin{gathered} \textstyle\sum_{n=1}^{N} b_n = B \quad b_n \in Z^+ \\ p_n^{err} \leq p^{err}_{target} \end{gathered}$$

The solution to the energy minimization problem after correct differentiation is again a sort of water-filling. However, in this case water/energy is poured as long as the total number of allocated bits does not exceed the target number of bits to be allocated (=

B_{target}), corresponding to a given BER and modulation and coding scheme. This energy minimizing form of the loading problem is known in mathematics as *the dual form* of the rate-maximization formulation [19].

The energy minimization is equivalent to margin maximization because any other bit distribution would have required more energy to reach the same given data rate. That greater energy would then clearly leave less energy to be distributed (uniformly) among the remaining subchannels. Since increasing the energy for a fixed data rate, in general, decreases an error probability metric, it is a waste of resources to allocate energy to a subchannel such that the error probability p_n^{err} is already less than p_{target}^{err}. Therefore, it is appropriate to meet the subchannel error probability constraint with equality if this is feasible, thereby leaving more energy avalailable to be distributed amongst the sub-channels.

2.3 State of the Art on Discrete Bit-Loading Algorithms

Using the classical Shannon equation for capacity, Gallager [9] was one of the first ones to establish the fact which the power distribution that maximizes the total capacity over multi-channel with variable gains is the 'Waterfilling' distribution. From the multicarrier transmission perspective over a frequency selective channel, Kalet's contribution [27] was the first to theoretical investigate the power optimization problem for multi-carrier system. It employs Lagrange optimization method to analytically prove that the solution for optimal allocation of power over a multi-carrier system has the same form as that of 'Waterfilling' solution of information theory. Water-filling algorithms result in bit distributions where the number of bits over a subcarrier($= b_n$) can be any real number. Realization of bit distributions with non-integer values can be difficult and not possible at times, because most modulation schemes (M-ary QAM, M- PSK etc.) have integer number of bits per symbol. Alternative loading algorithms allow the computation of bit distributions that are more amenable to implementation with a finite granularity. Mostly integer-based granularity is selected for the problem of bit-distribution, which is also commonly known as the 'discrete bit-loading' problem. In discrete bit-loading, only integer number of bits can be allocated to a sub-carrier which simplies the eventual implementation of the corresponding modulation technique. Over the past twenty years a huge amount of techniques/methods have been proposed for optimal discrete bit distribution problem, ranging from heuristic greedy approaches to approximation of optimal continuous solutions and from rigorous mathematical analysis to utilization of discrete lagrange optimization techniques . In this section, we will try to summarize the major contributions in this regard by classifying methods similar to each other in groups.

2.3.1 Greedy Approach based techniques

These techniques make use of the greedy optimization technique in mathematics. The very first algorithm regarding discrete bit-allocation over a multicarrier transmission was developed using this method and is known as the Hughes-Hartogs [10] algorithm. This algorithm assigns bits one-by-one to sub-carriers, taking into consideration the best option for a bit allocation at that particular step, and hence is called 'greedy'. The algorithm is not water-filling in the classical sense, but since it puts every increment of transmit power where it will be most effective, it appears to be optimum for multi-carrier transmission. This algorithm on one hand provides the overall optimal solution but on the other hand is computationally complex because of the marginal powers calculation for each sub-carrier, and the corresponding sorting of sub-carriers involved. The algorithm essentially consists of two phases:

- Calculate Bit-Incremental Energy for each sub-carrier. $\Delta e_n^b = \Delta e_n^b - \Delta e_n^{b-1}$ Where $E_{m,n}$ is the transmit energy needed in sub-channel n to transmit m bits per symbol at some pre-defined error rate.

- Assign one bit to the sub-carrier that requires the least incremental energy for a bit addition and update its Bit-Incremental energy.

The above loop is repeated till available energy is exhausted (for data-rate maximization) or the desired data-rate is achieved (for fixed data-rate). Similar to this technique of adding a bit one-by-one (= bit-filling), another approach known as greedy discrete bit-removal for Bit Error Rate (BER) constrained throughput maximization has also been proposed in [14]. The algorithm is based on greedy de-allocation(bit-removal) procedure where initially all the sub-carriers are loaded with the maximum constellation size respecting the closed-form BER equation at respective sub-carriers. Then constellation size is iteratively adjusted to make mean BER just less than the threshold BER. The comparisons made with competitive bit-allocation algorithms showed that the algorithm gave near-optimal performance while involving low computational complexity. Also based on this iterative bit-removal philosophy, a total power constrained throughput maximizing algorithm has been proposed [65] with additional constraints with min/max number of bits per bin which showed the computational advantages of a bit-removal algorithm over a bit-filling algorithm.

2.3.2 Approximation techniques

In order to tackle the high-complexity involved in the 'greedy' implementations of the discrete-bit loading problem, Peter S. Chow proposed [6, 19] a simpler channel-capacity based sub-optimal solution. The basic idea is to assign bits to the sub-carriers using the standard capacity equation [66] using 'performance margin' (γ_m) parameter, as shown in eq. 2.4, where performance margin is the same as defined previously.

$$b_n = \log_2 \left(1 + \frac{e_n \cdot CNR_n}{\Gamma + \gamma_m(dB)} \right) \tag{2.4}$$

where Γ is the gap factor as explained in the next section. In Chow's algorithm, for a fixed-rate application, γ_m is iteratively adjusted till the desired data-rate is achieved. If the algorithm does not converge after a pre-defined no. of iterations, it is made to converge in a forced manner. The algorithm claims to offer significant implementation advantages compared to the discrete-bit allocation optimal solution, [10] while suffering negligible performance degradation in comparison. Availability of 'Forced convergence' results in convergence within finite number of iterations. Energy distribution has the typical 'saw-tooth' shape with 3dB peak-to-peak difference because of integer-bit granularity. The basic structure of the algorithm can be expressed in the following three steps:

- Calculate the difference ($\Delta b_n = b_n^* - b_n$) between the 'actual' (float value) value of bit capacity (b_n) for each subcarrier (based upon shannon capacity equation) and its 'rounded' (integer value) version (b_n^*=round (b_n) and then remove sub-carriers with rounded capacity value equal to 0.

- Performance margin (γ_m) is increased/decreased by a specific amount if total number of allocated bits ($\sum_{n=1}^{N} b_n = B_{total}$) is greater/lesser than the desired number of bits (B_{target}).

- If $B_{total} \neq B_{target}$ and IterateCount is less than a pre-determined number of iterations (MaxCount), repeat the above loop, else perform forced convergence to B_{target}. This forced convergence is done by greedy addition/removal of bits till B_{target} is achieved, which renders complexity to the algorithm [5].

Another approximation based algorithm was proposed by Czylwik [17] which relates to the application of QAM-constellations based adaptive modulation with and without optimal power distribution for fixed data-rate applications in frequency and time selective channels. The basic idea behind Czylwik's algorithm is very much the same as that of Chow's algorithm since it is also based on bit allocations, making use of standard capacity equations for individual sub-carriers. However, after initial bit allocation, the convergence towards the desired data-rate is based upon the criterion of 'minimization of error probability' unlike Chow's 'rapid convergence' approach. Minimum overall error-probability is achieved by making the error probability for all used sub-carriers approximately equal.

Unlike the prevailing capacity-based bit allocation ideas of Chow and Czylwik, Fischer [18] allocated the bits over sub-carriers in order to maximize the SNR (minimize the probability of error). Since the solution can be derived from the closed-form expressions, it is claimed to be of low complexity as well.

2.3.3 Mathematical Analysis/Optimization methods based techniques

A complete and mathematically verifiable framework, as well as an approach that circumvents many drawbacks in the original Hughes-Hartogs methods was developed independently by Jorge Campello de Souza [67] and Howard Levin and are now known as Levin-Campello (LC) Algorithms. These algorithms solve the finite-granularity loading problem directly, which is not a water-filling solution.

Campello developed different mathematically verifiable conditions for a discrete allocation to be optimal with respect to a particular objective (throughput, energy etc.). Using these conditions and the mathematical theory of discrete optimization, efficient optimal algorithms for discrete bit allocation for multi-carrier systems were devised. If $\Delta e_n^{b_n}$ is defined as the energy required to increase the number of bits on sub-channel n from $(b_n - 1)$ to b_n, then by denoting $\mathbf{b}=\{b_n\}$ as a certain bit-allocation profile vector, Campello's conditions can be formally stated as :

- Definition 1: **b** is said to be efficient if

$$\Delta e_n^{b_n} \ \leq \ \Delta e_m^{b_m+1} \qquad \forall n, m = 1, 2..., N \ , \ m \neq n$$

which simply says that there should not exist another allocation with the same number of total bits but with less required energy.

- Definition 2: **b** is said to be E-tight if

$$0 \ \leq \ E \ - \ \sum_{n=1}^{N} e_n^{b_n} \ < \ min \ \Delta e_m^{b_m+1} \quad \forall n, m = 1, 2..., N$$

where E is the total available energy, which simply says that it is not possible to allocate one more bit while respecting the energy constraint.

- Definition 3: An allocation b is said to be B-tight if

$$\sum_{n=1}^{N} b_n \ = \ B_{target} \quad \forall n, m = 1, 2..., N$$

where B_{target} is the target number of bits to be allocated.

Similarly, by making use of the necessary conditions of optimality, Piazzo [12] developed theorems of equivalence between the systems optimal with power, rate and BER. Using Campello's sufficient condition for optimal bit allocation, Piazzo also developed algorithms for optimal discrete bit allocation which claimed better performances than those of Campello, as they were not solely confined to the *Gap approximation* based analysis. Another technique based on discrete lagrange optimization was proposed by Krongold [13] for solving both, the rate-maximization and the margin-maximization problems. Krongold observed that both, the *optimal* bit profile as well as the optimal energies to be allocated, can be found by making use of the discrete langrange optimization approach.

2.3.4 Subcarrier Bit-Incremental Energy Relationship

Finally, some latest published works make use of the subcarriers bit-incremental energy relationship which can be established by making use of Gap Approximation for capacity formulation. In this regard Papandreou [15] first proposed a *multi-phase* allocation procedure for converging towards the optimal bit-allocation for DMT applications. Later, we proposed a 3-dB subgroup classification of subcarriers, [Asad06] based on this bit-incremental energy relationship of subcarriers, which is described in the following section.Another recent contribution making use of this bit-incremental relationship is that of Esfahani [68], which proposes the two-step based allocation of bits based on the classification of the subcarriers with respect to their gains. Its final step consists of iterative (greedy) refinement of the allocation profile, so as to reach the target rate, which renders complexity to the algorithm.

Apart from the above mentioned proposals for the simplest case, some recent works on bit-loading have tried to tackle other extended problems like that of multi-user scenario [69], Peak-Power Constrained Bit-Loading, [20], MIMO-OFDM, [70] real-time related issues [71, 72] and LDPC-coded OFDM systems [73].Our work, in essence, attempts to contribute in two directions at the same time: firstly by proposing a novel optimal discrete bit-allocation methodology, which converges faster than the existing methods, and secondly through development of a *sufficient* condition for maximum bit allocation conforming to the peak-power-constraint. Significant works related to peak-power constrained discrete bit allocation can be found in [20, 74, 75].

2.4 $3dB$-Subgroup Classification of Subcarriers

2.4.1 Gap Approximation

Cioffi [19] was the first to establish the performance approximation of the QAM modulation in form of the classical shannon capacity equation by means of a *Gap Approximation* factor. The Gap parameter helps model and simplify the system analysis with bearable performance trade-off [19, 76], especially for large constellation size and at high SNR. The Gap parameter comes directly from the expression for symbol error rate for M-QAM in an AWGN channel which is tightly upper-bounded by

$$p_{sym}^{err} \leq 4 \cdot Q\left(\sqrt{\frac{3 \cdot SNR_{sym}}{M-1}}\right) \tag{2.5}$$

where Q function is given by

$$Q(x) = \int_{x}^{\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}} du \tag{2.6}$$

If we replace $M = 2^b$ in the above equation, where b represents the number of bits per symbol, then the above equation can be written as

$$b \leq \log_2 \left(1 + \frac{3.SNR_{sym}}{\left[Q^{-1} \left(\frac{p^{err}_{sym}}{4} \right) \right]^2} \right) \tag{2.7}$$

If we represent $\Gamma = \frac{\left[Q^{-1} \left(\frac{p^{err}_{sym}}{4} \right) \right]^2}{3}$ in the above, it can be written as

$$b \leq \log_2 \left(1 + \frac{SNR_{sym}}{\Gamma} \right) \tag{2.8}$$

which is exactly the same as Shannon's equation for the capacity in AWGN channel except for the SNR gap parameter (Γ). The quantity Γ is called the 'SNR Gap' because we see in the above equation that the number of bits that can be reliably transmitted is less than the theoretical capacity. It is thus, the capacity of the channel with a factor of $\Gamma(dB)$ less $SNR_{sym}(dB)$. As Γ approaches 1 (0 dB), then the achievable data-rate of the QAM system approaches capacity. An important thing to note is that it depends on the error probability requirements. The above expression is most accurate for higher order M-ary QAM constellations. Although it is not always exact, it constitutes a good approximation that simplifies bit loading algorithms and extended analysis of this gap parameter for M-ary PSK constellations have been carried out in [76]. Furthermore if we define γ_m as the performance margin (as defined previously) and γ_c as the coding gain for a given channel coding scheme, then the new SNR gap factor can be calculated as

$$\Gamma_{new}(dB) = \Gamma_{old}(dB) + \gamma_m(dB) - \gamma_c(dB) \tag{2.9}$$

2.4.2 Bit-Incremental Energy Relationship and 3-dB subgroup Classification

Using the Gap parameter, the maximum number of bits(b_n) per symbol that can be sent over subcarrier n in a multicarrier system are given by

$$b_n = \log_2(1 + \frac{e_n \cdot CNR_n}{\Gamma}) \tag{2.10}$$

where $CNR_n = |H_n|^2 / N_n$ is the channel gain to noise ratio with H_n and N_n representing the channel gain factor and noise power at subcarrier n respectively. Γ is the Gap parameter as described earlier for approximating behavior of the employed modulation and coding for a desired performance criterion and e_n represents the energy allocated to sub-carrier n. Using eq. 2.10, the energy required to transmit an additional bit on subcarrier n is given by

$$\Delta e^{+}_{b_n} = e_n^{b_n+1} - e_n^{b_n} = \frac{2^{b_n} \cdot \Gamma}{CNR_n} \tag{2.11}$$

The conceptual basis of our allocation algorithm lies on the simple principle easily observable from equation 2.11, which describes the recursive relationship between the required incremental energies for successive bit-allocations over the same sub-carrier. In other words the energy required to increment bits from $b_n \rightarrow b_{n+1}$ is twice (requires 3dB more) the energy required to increment bits from $b_{n-1} \rightarrow b_n$. Mathematically, this recursive relation is represented as

$$\Delta e^{+}_{b_n} = 2 \cdot \Delta e^{+}_{b_{n-1}} \Rightarrow \Delta e^{+}_{b_n dB} = \Delta e^{+}_{b_{n-1} dB} + 3dB \tag{2.12}$$

Let Δe_n^{0+} denote the energy increment required to allocate the first bit over subcarrier H_n. Then the first subcarrier that will be allocated a bit will be the one requiring minimum Δe_n^{0+} (Δe_{min}^{0+}) or in other words with maximum H_i (H_{max}). 2.12 implies that before incrementing bits from 1 to 2 over H_{max}, a bit is allocated over all the subcarriers with Δe_j^{0+} within the 3dB range of Δe_{min}^{0+}. In other words k bits are allocated to H_{max} before the first bit is added to subcarrier j such that

$$2^{k-1} \cdot \Delta e_{min}^{0+} \leq \Delta e_j^{0+} < 2^k \cdot \Delta e_{min}^{0+} \tag{2.13}$$

$$\forall \;\; 0 < j \leq N \; ; 0 < k$$

If $L = \lceil \Delta e_{max}^{0+} / \Delta e_{min}^{0+} \rceil$[1], we can divide all the subcarrier in L groups of $3dB$ width each with respect to $\Delta e_n^{0+}{}_{dB}$. This means that a subcarrier n belongs to the subgroup l if

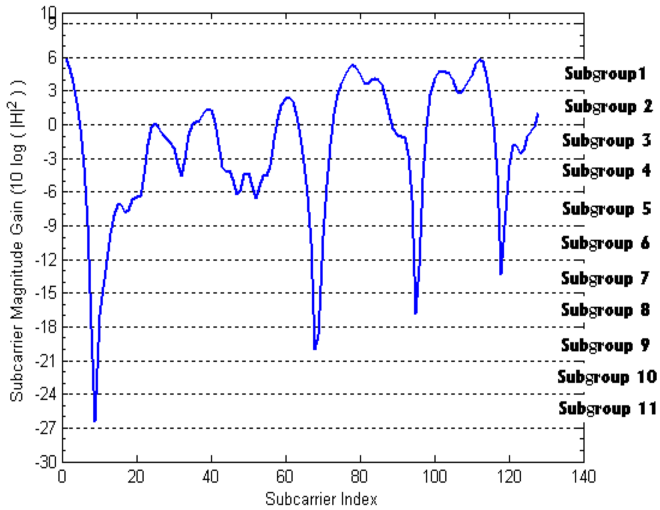
$$2^{l-1} \cdot \Delta e_{min}^{0+} \leq \Delta e_n^{0+} < 2^l \cdot \Delta e_{min}^{0+} \tag{2.14}$$

$$\forall \;\;\; 1 \leq l \leq L \; ; \; 0 < i \leq N$$

Thus if we imagine all the subcarriers sorted in descending order with respect to their channel gain factors, then the subcarriers with the smallest channel gain are assigned to

[1] $\lceil \; \rceil$ denotes the 'ceil' function

Figure 2.4 Classification of Sub-carriers in Subgroups

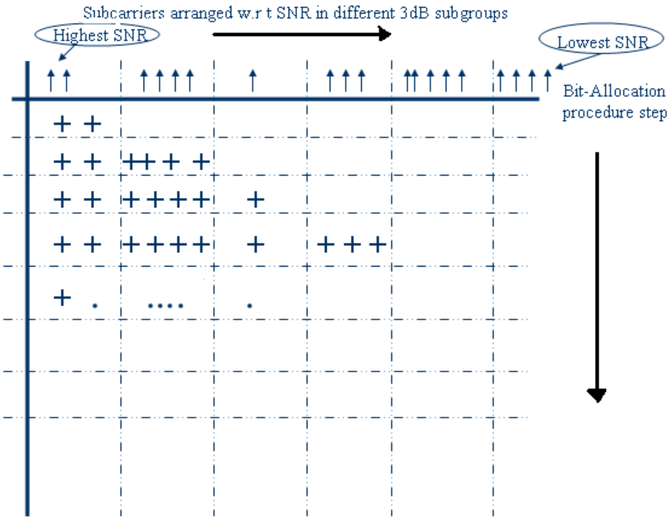


the first group, and the subcarriers with the largest channel gain are assigned to the last (Lth) group. An example of such subcarrier partitioning will be shown in figure 2.4. The number of subcarriers in each group will not be same and will be determined by channel frequency response of the concerned channel. Using the above mentioned bit-incremental energy relationship coming from the Gap approximation equation and this sorting of subcarriers in different subgroups in the ascending order with respect to the respective channel gains, we can detect the orderly manner with which the optimal discrete bit-allocation takes place and which can help us devise a simplistic allocation algorithm. It is important, however, to mention here that this 'sorting' of the subcarriers is only for imagining the underlying rythm, and for the actual allocation, the sorting of the subcarriers with respect to their incremental-energies is not required by the algorithm itself as will be evident by the algorithm details in the next section.

2.5 $3dB$-Subgroup allocation methodology

Based upon the 3-dB subgroup classification of subcarriers as explained above, an allocation rythm that is underneath the classical *greedy* allocation method can be traced. This allocation rythm can be best explained by means of the figure 2.5, where the subcarriers are represented by means of small vertical arrows and are assumed to be placed along the horizontal axis after sorting in descending order with respect to their magnitude gains, which means the first sub-carrier will be the one with the highest magnitude gain fac-

Figure 2.5 Bit-Allocation Rythm with respect to 3-dB subgroups



tor. The dashed vertical lines represent the classification of subcarriers in different 3-dB sub-groups as explained in the last section, and the number of subcarriers in different subgroups can vary as depicted in the figure. Now we divide the complete optimal bit-allocation process into smaller steps where these steps are represented by means of the horizontal dashed lines.

We found out that using this arrangement, the optimal bit-allocation procedure can be seen to follow a beautiful rythm not only for the allocations occurring during a single step but also a pattern exists for allocations across different steps of bit-allocation process.

2.5.1 Allocation Rythm across Different Steps of Bit-Allocation

It was observed that in the first step of the allocation process, only the subcarriers belonging to the first 3-dB subgroup are allocated a single bit. In the second step, both the first and the second sub-group are allocated a single bit and so in each new step a new sub-group's subcarriers are included in the allocation process. In other words from 1 till 2 bit allocation over n_{max} (i.e. the subcarrier with maximum channel gain and which is located in the first 3-dB subgroup), a bit will be allocated to subcarriers belonging to subgroup 1 only , while moving from 2 to 3 bits over n_{max}, a bit will be allocated to subcarriers of subgroup 1 and 2. If the target total number of bits B_{target} is not achieved even after L number of allocation steps where L represents the total number of 3-dB subgroups, then each new step will involve all the subcarriers till the desired total number of bits B_{target} is achieved.

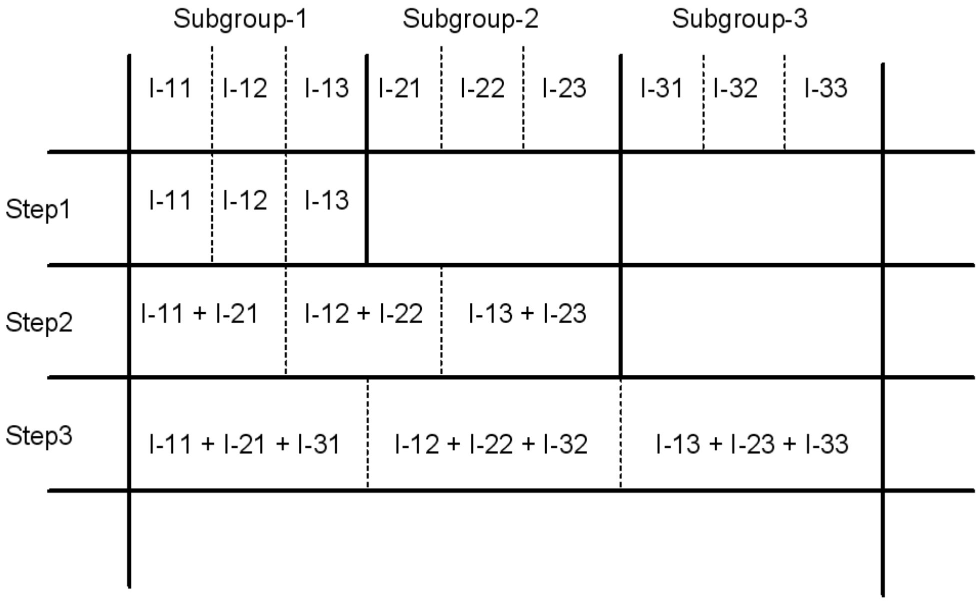
2.5.2 Allocation Rythm Within a Single Step of Bit-Allocation

As stated earlier and as can be easily observed from the figure that the number of 3-dB subgroups involved in a particular step's allocation is directly related to the number of that step i.e. in the first step only the first subgroup's subcarriers are used in allocation, in second step both the first two subgroups are used whereas in the step three, all the first three 3-dB subgroup's are employed for the bit-allocation during that step and so on. The reason for this is that after allocation of a bit to the subcarriers in the first subgroup, their bit-incremental powers get increased by a factor of $2(= 2^1)$ and thus they come within the 3-dB range of the next subgroup i.e. subgroup 2. Similarly after allocation of b bits to the subcarriers of the first subgroup, they come within the 3-dB range of the subgroup number b. For any step in which a total of $\bar{l}$ 3-dB subgroups are involved in the allocation in that step where $1 \leq l < \bar{l}$, the number of bits allocated to subgroup l will be one more than the number of bits allocated to subgroup $l + 1$. Also at $\bar{l}$ step, subcarriers belonging to all the subgroups from 1 till $\bar{l} - 1$ will come within the 3-dB range of the subgroup $\bar{l}$ alongwith the the subcarriers belonging to $\bar{l}$ subgroup.

The interesting thing to observe is that the original *distance* ratio of a subcarrier belonging to a subgroup is maintained even when it is projected to another subgroup's 3-dB range, where *distance* is defined by considering that all the subcarriers are arranged in a horizontal line with respect to their SNR in descending order, as shown in figure 2.6. Hence, the distance between the actual position of a subcarrier within a subgroup and the minimum possible value in that subgroup represented by the left vertical dashed line to a subgroup as shown in figure 2.6. This means that a subcarrier in the exact middle of a subgroup n will be found at the exact middle of the subgroup j, when projected to that subgroup.

Based upon the above arguments, it was observed that during a single step , since all the subcarriers of all the concerned subgroups have been projected in the same 3-dB range (the 3-dB range of the last 3-dB subgroup involved in bit-allocation during that step), the bit allocation starts with the subcarrier with the least *distance* ratio, where distance is as defined above and so the bits are incremently allocated to subcarriers with increasing *distance* without regard to what subgroup they originally belonged to. Thus the subcarriers with least *distance* ratio of no-matter what subgroup they initially be-

Figure 2.6 Classification of Sub-carriers in 3 Intervals for Each Subgroup



longed to gets allocated first and the subcarriers with higher *distance* ratio gets allocated near the end of that step's allocation procedure.

Based upon these principles, we have divided our allocation algorithm in two phases of initial and final allocation.

2.5.2.1 Initial allocation

Our main goal in this phase is to determine the maximum number of 3-dB subgroups that will be required to allocate B_{target} number of bits. For this, we define ns_l as the number of subcarriers in subgroup l, such that $\sum_{l=1}^{L} ns_l = N$. Now the whole allocation process can be divided in smaller allocation steps j as explained above. A step may be defined as the allocations between successive bit allocations over H_{max} (first 3-dB subgroup) i.e. step $j = 1$ refers to the allocations from the first bit allocation and before the second bit allocation on H_{max} with the total number of bits allocated in step 1 denoted by B_1. Therefore, step j in general refers to the bit allocation done in-between $j \rightarrow j+1$ bits over H_{max}, with B_j representing the total bits allocated till step j. It was found that the total number of bits allocated till step $j^{'}$ can be recursively calculated as

$$B_{j'} = B_{j'-1} + \sum_{j=1}^{j'} ns_j \;\; \forall \;\; j \geq 1 \tag{2.15}$$

$$(B_0 = 0) \; ; \; (ns_j = ns_L \; \forall \; j > L)$$

We can state that j^* is the number of maximum step number till which the allocation procedure goes in order to allocate B_{target} bits, if and only if

$$B_{j^*-1} < B_{target} \leq B_{j^*} \tag{2.16}$$

The above equation implies that the target number of bits B_{target} gets allocated after the completion of step $j^* - 1$ and before the completion of step j^*. Since B_{target} will be achieved during the j^* step, each step till $j^* - 1$ will allocate a bit to all the subcarriers of all the subgroups involved during that step but during the step j^* all the subcarriers of the concerned subgroups may or may not be allocated a bit as the bit allocation procedure will stop as soon as $B_{total} = B_{target}$. The initial allocation phase of our algorithm deals with the allocation from step 1 till step $j^* - 1$, while the final allocation phase deals with the allocation during step j^*. Once the value of j^* is determined, it is known that $j^* - 1$ subgroups will be employed during the initial allocation phase, since with each step increment an additional subgroup is involved, starting from only subgroup 1 usage at step 1. Since only one bit is allocated to all the subcarriers involved in a step, at the completion of $j^* - 1$ steps, $j^* - 1$ bits would have been allocated to all the subcarriers in subgroup 1, $j^* - 2$ bits to subcarriers in subgroup 2 and similarly one additional bit less for each next subgroup till subgroup $j^* - 1$, whose subcarriers would have been allocated a single bit. So if j represents the 3-dB subgroup number, then after initial allocation phase (i.e. after $j^* - 1$ steps of allocation), the number of bits allocated to subgroup j can be given by

$$b_j^{initial} = j^* - j \quad \forall \ \ 1 \leq j \geq j^* - 1 \tag{2.17}$$

Now if the 3-dB subgroup number $j(n)$ of any given subcarrier n can be determined as follows

$$j(n) = \left\lfloor \log_2 \left(\Delta e_n^{0+} / \Delta e_{min}^{0+} \right) + 1 \right\rfloor \tag{2.18}$$

then the bits allocated to subcarrier n in the initial allocation phase are given by

$$b_n^{initial} = j^* - \left\lfloor (\log_2 (\Delta e_n^{0+} / \Delta e_{min}^{0+}) + 1) \right\rfloor_0 \tag{2.19}$$

where $\lfloor \rfloor_0$ indicates the floor function with the minimum value of 0.

2.5.2.2 Final Allocation

The initial allocation phase results in a bit-profile with a maximum one bit difference per subcarrier with respect to optimal allocation profile .The final allocation phase determines which subcarriers should be allocated an additional bit during the j^* step in order to achieve the desired (B_{target}) number of total bits with optimal bit-allocation profile i.e. the same profile which will be achieved through the *Greedy* or the *Hughes-Hartogs* method. This allocation of remaining bits in the j^* step can be achieved by a number of ways.

1. *Complete Sorting* : In [21] we had proposed the allocation of the remaining bits to be allocated in the final phase based upon the sorting of all of the subcarriers. If $\Delta p_n^{initial+} = (2^{b_n^{initial}+1} - 2^{b_n^{initial}}) * \Delta p_n^{0+}$ represents the energy increment required to add a bit over subcarrier n after the initial allocation phase, and $B_{rem} = B_{target} - \sum_{N_{sub}}^{n=1} b_n^{initial}$ represents the total number of bits that remains to be allocated after the initial phase so as to achieve the target number of bits b_{target}, then the remaining bits can be allocated as

$$\overline{\mathbf{b}}_{sort}(n) = \overline{\mathbf{b}}_{sort}(n) + 1 \ \ \forall \ \ 1 \leq i \leq B_{left} \tag{2.20}$$

where $\overline{\mathbf{b}}_{sort}$ refers to the subcarriers bit profile vector after the initial allocation phase, sorted with respect to the ascending order of $\Delta e_n^{initial+}$. Thus, the final allocation step or the step j^* requires allocation of a single bit to the B_{rem} subcarriers which require the least amount of energy for a bit-increment or in other words B_{rem} subcarriers with minimum $\Delta e_n^{initial+}$. Hence, in order to achieve that, in this method we have sorted all of the subcarriers in ascending order with respect to $\Delta e_n^{initial+}$ and then allocated a bit to the first B_{rem} subcarriers thereby resulting in the optimal profile. On one hand this method being simple otherwise renders unnecessary complexity to the allocation process e.g. if only a few bits are left to be allocated in the final step, a complete sorting of all the subcarriers is not the best of the approaches and hence we observed that the overall algorithm complexity can be largely reduced by replacing the above method with the following method.

2. *Interval Classification based Final Allocation*

   We introduced the term *distance* in the start of this section. Based upon this definition of distance, we classified the subcarriers belonging to a subgroup in $I = 8$ intervals. The number of intervals can be selected as a function of the total number of subcarriers in a system, such that assuming that subcarriers in a subgroups have equal probability of lying in different intervals, the possible number of subcarriers in an interval comes out to be a modest number. Subcarrier n belongs to the interval

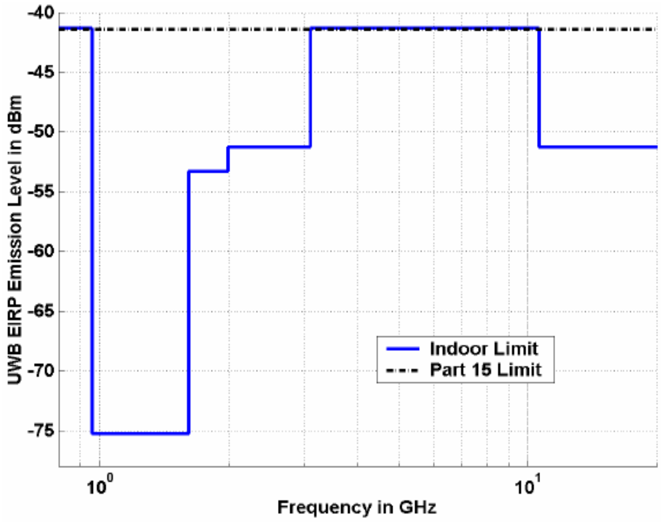
i (where $i \geq 1$) of a subgroup l if

$$\left(1 + \frac{(i-1)}{8}\right) \leq (\Delta e_n^{0+}/\Delta e_l^{min0+} < \left(1 + \frac{i}{8}\right) \qquad (2.21)$$

In order words it means that the subcarriers will belong to the interval 1 if they are within 1/8th of the distance from the minimum possible value in a subgroup till the next subgroup and will belong to the interval 2 if they are in-between the 1/8th and the 2/8th of the distance from the minimum possible value in a subgroup till the next subgroup and so on. Once this classification is performed, it can be easily deduced that during the allocation process of a step, first the subcarriers belong to the first interval of all the concerned subgroups are allocated a bit, then a bit is allocated to the next interval of all the subgroups and so on till all the intervals of all the subgroups have been served. Based upon this rythm, we formulated the final allocation process of the j^* step of the allocation process. As we have established earlier that all the subcarriers of all the concerned subgroups will be within the 3-dB range of the j^* subgroup at j^* step, which implies that all the concerned subcarriers for this group can be classified into a total of 8 intervals.

Now the method becomes very much same as the approach used in the initial allocation. In initial allocation, our basic goal was to find the number of vertical bit-allocation *steps* that are needed so as to allocate any given target number of bits B_{target}. Considering again the figure showing the interval classification where the subcarriers are classified in different intervals, our basic goal in this step will be to find the total number of intervals that will be required for B_{rem} number of bits to be allocated. If i^* represents the total number of intervals that will be used to allocate B_{rem} number of bits in the final allocation step where $1 \leq i^* \leq 8$, then it implies that all the subcarriers of all the intervals till $i^* - 1$ will be allocated a bit in this step and some or all of the subcarriers of the i^* interval will be allocated a bit depending on the remaining B_{rem}. Finally the bits remaining to be allocated to the last interval i^* can be allocated by iteratively searching for the subcarrier with minimum $\Delta e_{i^*}^{initial+}$ amongst the subcarriers of the interval i^* and removing this subcarrier from the search space for the newt iteration after allocating a bit to it. It is important however to mention here that this iterative or greedy search over the subcarriers of the i^* interval is far less complex than if the search is made over all of the intervals.

Figure 2.7 Spectral Mask Defining Peak-Power Emission Allowed for UWB Communications in US



As simulation results will reveal, after the final phase allocation the bit profile will resemble that of the optimal bit allocation and the complexity involved will be compared to the recently proposed Papandreou's [15] algorithm for optimal discrete bit allocation alongwith the classical greedy solution for different scenarios showing that our proposed algorithm can achieve the exact same optimal bit profile with much less complexity.

2.6 Optimal Allocation with Peak Power/Energy Per Subcarrier Constraint

The scarcity of the available bandwidth along with the increasing throughput demands often results in different systems operating over overlapping frequency. To avoid interference in the overlapping frequency band, the regulatory authorities impose a 'Spectral Mask' which dictates the maximum power/energy i.e. $\overline{P}$ ($\overline{e}$) that should be transmitted per Hz over a particular frequency-range, and is also termed as the peak-energy constraint in literature [20]. The spectral mask for the MB-OFDM based Ultra Wide Band (UWB) system is shown in the figure 2.7.

This $\overline{e}$ dictates the maximum number of bits ($\overline{b}_n$) that could be allocated to a subcarrier for a given QoS such that the Spectral Mask is respected. Hence upon introducing the peak-energy constraint, our bit allocation procedure must take into account this new peak-energy constraint.

The maximum number of bits that could be transmitted over a sub-carrier in presence of a peak-energy constraint can be represented as

$$2^{\overline{b}_n+1} - 1 > \frac{\overline{e}_n}{\Delta e_n^{0+}} \geq 2^{\overline{b}_n} - 1 \tag{2.22}$$

Its proof comes directly from the expression for the valid range of bits that could be allocated over a subcarrier. If $\overline{e}_n$ and $\overline{b}_n$ denote the maximum energy and bits that could be transmitted over subcarrier n and Δe_n^{0+} is the energy required to add a single bit over n, then $\overline{b}_n$ is equal to the maximum value of n^* which satisfies the following expression

$$\frac{\overline{e}_n}{(1+2+4+\ldots+2^{n^*-1})\cdot \Delta e_n^{0+}} = \frac{\overline{e}_n}{(\sum_{n=1}^{n^*} 2^{n-1})\Delta e_n^{0+}} \geq 1 \tag{2.23}$$

The multiplying factor in the denominator has the form of a geometric series. Since $\sum_{i=1}^{i^*+1} a^{i-1} = \frac{a^{i^*+1}-1}{a-1}$ For geometric series in equation 2.23, $a = 2$ and $n^* = i^* - 1$ and $\overline{b}_n = n^*$, thus giving

$$\frac{\overline{e}_n}{(2^{n^*-1})\cdot \Delta e_n^{0+}} \geq 1 \;\; \Rightarrow \;\; \frac{\overline{e}_n}{\Delta e_n^{0+}} \geq 2^{\overline{b}_n - 1} \tag{2.24}$$

thus proving the right hand side of equation 2.22, while the left hand side is easily verified by observing that the value of $(\overline{e}_n/\Delta e_n^{0+})$ in equation 2.22 has to be lower than the $(2^{\overline{b}_n+1} - 1)$ value in order not to increment the value of $\overline{b}_n$ to $\overline{b}_n + 1$. Equation 2.24 also leads to the expression for maximum bit allocation conforming to the peak-energy-constraint $\overline{e}_n$ and is given by

$$\overline{b}_n = \left\lfloor \log_2 \left(\frac{\overline{e}_n}{\Delta e_n^{0+}} + 1 \right) \right\rfloor \tag{2.25}$$

We can expect three different cases, if a peak-energy constraint is present: 1) Same energy constraint for all the subcarriers, 2) Different peak-energy constraint possible for different 3-dB subgroups but same for all the subcarriers within a single subgroup and 3) Different peak-energy constraint possible for different 3-dB subgroups as well as for all the subcarriers within a single subgroup. In any case, for our concerned margin maximization problem, first it is required to verify that that our required B number of bits can be allocated in presence of such a peak-energy constraint without its violation. Once this is verified, the allocation is performed as prescribed in the previous section. This condition can be easily verified by

$$B \leq \sum_{n=1}^{N} \bar{b}_n \qquad (2.26)$$

We can also determine the least possible value for $\bar{e}$ that is required for B number of bits to be allocated without violation of the spectral-mask for case 1 above.Based upon the earlier concepts and notations of subgroups and interval classification, if we require a total of j^* number of steps of allocation and i^* number of intervals for the allocation of B bits, then the peak-energy constraint will not be violated if and only if

$$2^{j*-1} \cdot \left(1 + \frac{i^*}{8}\right) \cdot \Delta e_{min}^{0+} \leq \bar{e} \qquad (2.27)$$

Once the feasibility of allocation is established the allocation is performed in a normal manner for the above mentioned case 1. However for the above cases of 2 and 3, where the peak-constraint may vary from subgroup to subgroup or even from subcarrier to subcarrier, an iterative initial allocation will be required to be performed where in each iteration the subcarriers achieving their spectral-mask limit are allocated the maximum permissible number of bits and are then excluded from the allocation process and then this process is repeated till enough bits are left that can be allocated in the final allocation phase.

2.7 Simulation and Results

2.7.1 Simulation Scenario

Our bit-loading algorithm is valid for any multicarrier system, be it wireline or wireless. As an application scenario we used the parameters and the channel models corresponding to the MultiBand-OFDM (MB-OFDM) based UWB system [77], which has recently been adopted by the WiMedia Alliance, a consortium of leading telecom companies, as a standard for wireless communications consumer application products, targeting very short range ($< 10m$) and high data-rate ($> 500Mbps$).

MB-OFDM system, as like other proposals for IEEE 802.15 standard, employed Saleh-Valenzuela model [78] as the reference indoor UWB channel model for performance evaluation purposes.The real valued model is based on the empirical measurements originally carried out in indoor environments in 1987 [78]. Due to the clustering phenomena observed at the measured UWB indoor channel data, a clustered-based channel model was proposed which makes use of a log-normal distribution rather than an original Rayleigh distribution for the multipath gain magnitudes. An independent fading mechanism is assumed for each cluster as for each ray within the cluster and both the cluster and ray arrival times are modeled independently by Poisson processes.

Based upon SV channel model and different indoor UWB communication scenarios, four different channel scenarios were standardized namely $CM1$(LOS[2] 0-4m), $CM2$(NLOS[3] 0-4m), $CM3$(NLOS 4-10m) and $CM4$ (Extreme NLOS) as proposed in [23]. The quasi-static nature of these channel scenarios especially CM1 and CM2 makes the MB-OFDM system feasible for adaptive resource allocation and hence the research in this regard [79].Although the bit-loading algorithm complexity is not significantly affected by the underlying channel model, but for the sake of realistic values, we employed channel gains corresponding to the CM1 channel model for our bit-profile analysis purposes.

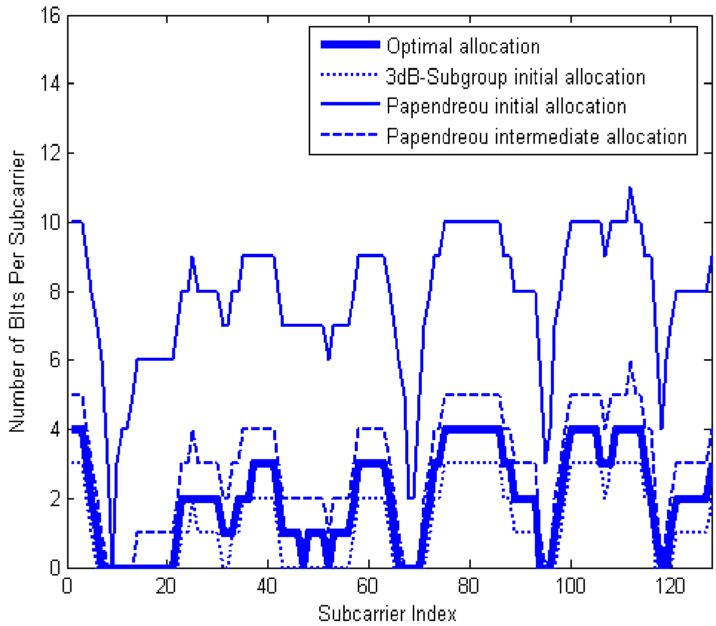
2.7.2 Bit-Allocation Profile

The performance and complexity of our allocation algorithm was compared with two other discrete bit-loading algorithms leading to the optimal solution with 'Gap' approximation, one the classical greedy approach first proposed by Hughes Hartogs [10] and the other a recently proposed algorithm for bit-loading based on multi-phase allocation [15]. The bit allocation profile during the intermediate stages of our 3dB subgroup algorithm and that of Papandreou are given in figure 2.8 alongwith the optimal profile for the concerned channel. The total number of subcarriers were taken as 128 and the total number of bits to be allocated as 256 (assuming QPSK on each subcarrier with no loading) based on the MB-OFDM system parameters and the channel gains were that corresponding to the CM1 channel.

The Hughes Hartogs algorithm performs the costly bit-by-bit allocation and directly converges to the optimal. The Papandreou algorithm performs the bit-allocation in three phases with finally converging to the optimal. The initial step allocates bits to each subcarrier with respect to the channel gain at the most attenuated subcarrier in the initial phase of the algorithm. In wireless channel environments, which are susceptible to deep fades, the latter approach can lead to an allocation far different from the desired bit-profile. For this reason, that method needs an intermediate step to approach the

[2]Line Of Sight
[3]Non Line Of Sight

Figure 2.8 Bit-allocation profiles for different Algorithms



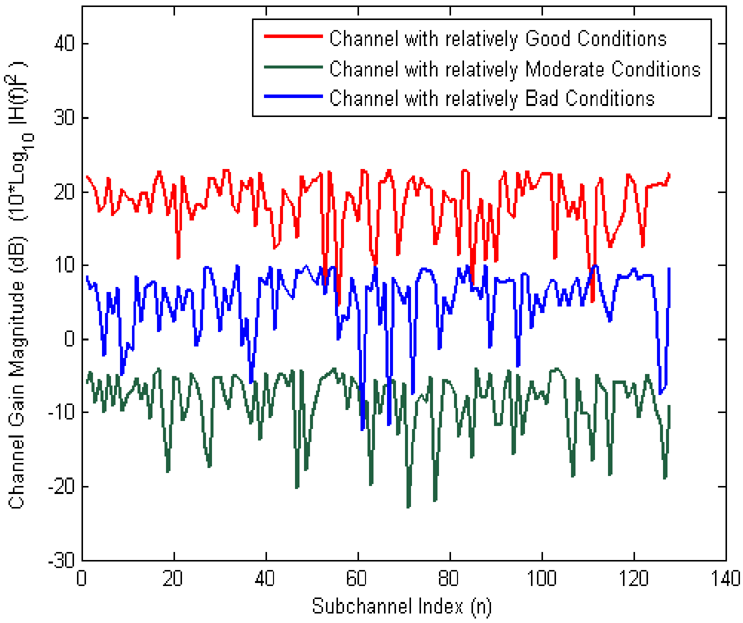
target profile, before finally converging towards the optimal profile in the third and final allocation phase. The final step of the allocation performs a Hughes-Hartogs like greedy allocation for the bits remaining to be allocated over the entire set of subcarriers, which when large number of subcarriers are present can prove to be very costly even if the number of bits remained to be allocated is few.

In contrast, our initial allocation being at a maximum difference of one bit per subcarrier from the optimal bit-profile, converges directly to the optimal solution in the second step, which is indicative of its faster convergence. The hierarchical classifical of all of the subcarriers first into 3-dB subgroups and then into the *intervals* makes the subspace of the greedy allocation involved in our final allocation very small, hence reducing the overall algorithm complexity.

2.7.3 Total Energy Improvement Factor

As mentioned earlier, the proposed bit-loading algorthm is applicable to any multicarrier communication system, wireline or wireless. Many state-of-the-art communication systems (DSL, DVB, WiFi, UWB, WiMax etc.) that employ multicarrier technology at their physical layer, inherently operate on largely diverse channel conditions. The exact gain in performance by the application of the bit-loading (adaptive modulation) technique

Figure 2.9 A particular instance of relatively Bad, Moderate and Good channel conditions



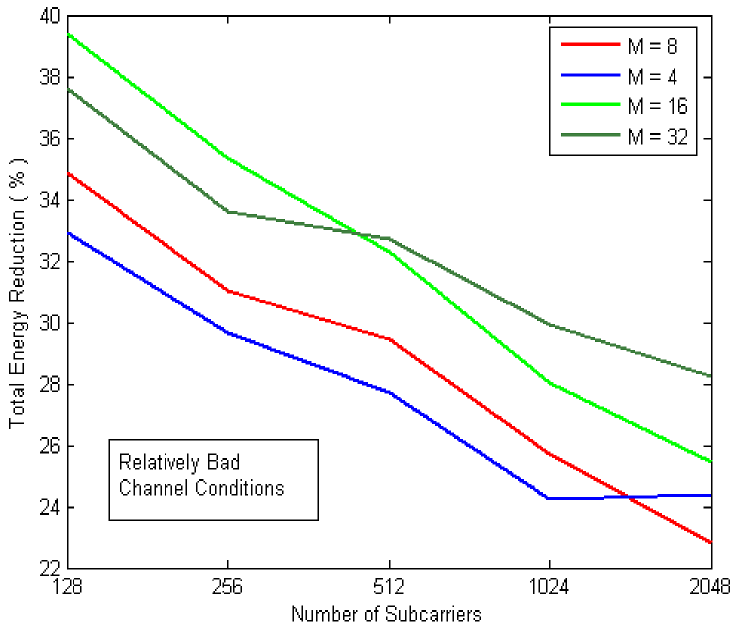
is thus dependant upon the particular system's operating parameters and channel conditions. Our main contribution, which relates to the complexity reduction of the bit-loading algorithm, is valid for any given channel conditions and system parameters.

Hence we give the energy reduction estimate due to the application of bit-loading, for three different types of channel conditions, that cover the range of channel conditions of most of the communication systems. The channel states of these arbitrary channels were made to lie between [-23 dB to 4 dB], [-9 dB to 10 dB] and [4 dB to 23 dB] as shown in figure 2.9. These ranges stand for relatively bad, moderate, and good channel conditions, respectively [25].

The reduction in total energy for a given throughput, as was our concerned *Energy Minimization* problematic in this chapter, is given in figure 2.10 and 2.11. Each energy reduction value was averaged from a total of 100 channel realizations of the considered scenario.

It was generally observed that as the number of subcarriers is increased in a system for a given channel scenario and target throughput (M), the percentage of reduction in total Energy becomes less. This is because of the fact, that when the subcarriers are randomly dispersed in the concerned CNR range, as in our case, an increase in the total

Figure 2.10 Reduction in Total Energy by means of Bit-Loading for different values of N (Total No. of subcarriers) and M (M-ary modulation scheme per subcarrier when no bit-loading)



number of subcarriers results in the availability of more relatively *good* subcarriers. As we have shown by means of 3-dB subgrouping in this chapter, the bit-allocation starts from the best subcarriers and gradually includes the decreasing CNR subcarriers, the participation of more *good* number of subcarriers in the bit-loading procedure leads to an overall reduction in the total energy reduction.

2.7.4 Complexity Comparison

2.7.4.1 Expected Algorithm Complexity

Figure 2.12 presents the expected algorithmic complexity of the different phases of the concerned algorithms. Since the number of bits to be allocated in a particular phase varies from case to case for a particular algorithm, a precise comparison of complexity can only be made by actual execution of algorithms on hardware for a given channel scenario and target bit-rate, as performed below. Figure 2.12 also depicts the well-known high complexity of the greedy bit-filling approach as it requires N operations to allocate a single bit, where N indicates the total number of subcarriers. For the algorithm of

Figure 2.11 Reduction in Total Energy by means of Bit-Loading for different values of N (Total No. of subcarriers) and M (M-ary modulation scheme per subcarrier when no bit-loading)

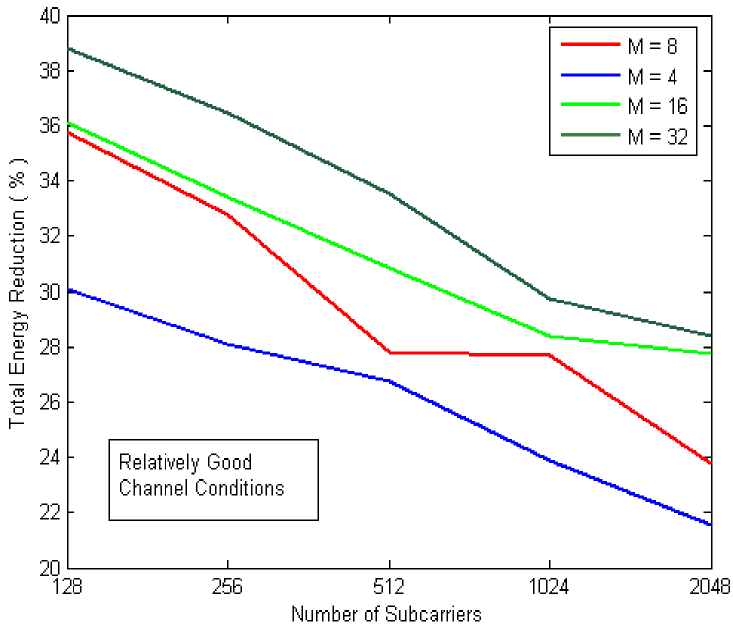


Figure 2.12 Expected Algorithmic Complexity for Different Algorithms

Algorithm	Complexity
Greedy	O(B.N)
Papendreou Initial Phase	O(N)
Papendreou Intermediate Phase	O(N)
Papendreou Final Phase	O(B_left1.N)
3-dB Subgroup Initial Phase	O(N)
3-dB Subgroup Final Phase-1	O(N)
3-dB Subgroup Final Phase-2	O(B_left2.N_interval)

Papandreou [15], although the initial and intermediate phases do not require extensive computations, the final phase depends on the factor B^{left}(the remaining bits that will be allocated using the greedy procedure), which when large can lead to high complexity because each bit allocation in this phase will require $O(N)$ order of operations. Even for the cases where B^{left} is not large but the total number of subcarriers is large, the final phase of Papandreou algorithm renders large complexity because all the subcarriers have to be compared before allocation of a single bit.

In our proposed 3dB-subgroup allocation algorithm, we find that the initial phase requires classification of sub-carriers with respect to 3-dB difference from the least attenuated subcarrier where this classification can be efficiently implemented in a number of operations which is linear with respect to N. Final allocation requires further classification in different *intervals* whose complexity is also linear with respect to the number of subcarriers N. Finally the B^{left} number of bits are greedily allocated to the last interval, which requires only comparing the subcarriers belonging to the last interval and hence an eventual reduction in complexity.

2.7.4.2 Exact Number of Execution Cycles on a Processor

In order to have a precise complexity comparison, along with performing the theoretical complexity analysis, we also compared the concerned algorithms in terms of actual number of execution cycles, when executed over a processor. Theoretical complexity analysis and a comparison of simulation times on any simulation tool, can only provide an approximate complexity comparison. Most simulation environments, like Matlab, are based on interperative languages and hence are very much dependent on the intermediate compiler. Therefore, it is preferable to use some other method/tool to evaluate the exact number of cycles taken by an algorirthm.

In this regard, Simplescalar [61] is a popular tool, that enables extracting the parameters (No. of execution cycles, energy consumed etc.) concerned with the execution of a given program, for a wide variety of processors. Therefore, we employed the Simplescalar tool, to extract the exact number of execution cycles taken by different bit-loading algorithms over a MIPS [80] processor architecture processor.MIPS (originally an acronym for Microprocessor without Interlocked Pipeline Stages) is a RISC (Reduced Instruction Set Computer) microprocessor architecture and is primarily used in embedded processors. The default parameters of the Simplescalar tool, with which the algorithms were run, are shown in figure 2.13.

2.7.4.3 SimpleScalar Tool For Execution-Based Algorithm Complexity Analysis

Simplescalar is an execution driven cycle accurate instruction set simulator (ISS) of a superscalar microprocessor [61]. A complete development chain (compiler, debugger, profiler) comes with the tool which allows the quick porting of any ANSI C application to Simplescalar. The Simplescalar toolset is composed of a gcc compiler ported for the

Figure 2.13 The Default values of the Superscalar Processor Architecture used by SimpleScalar

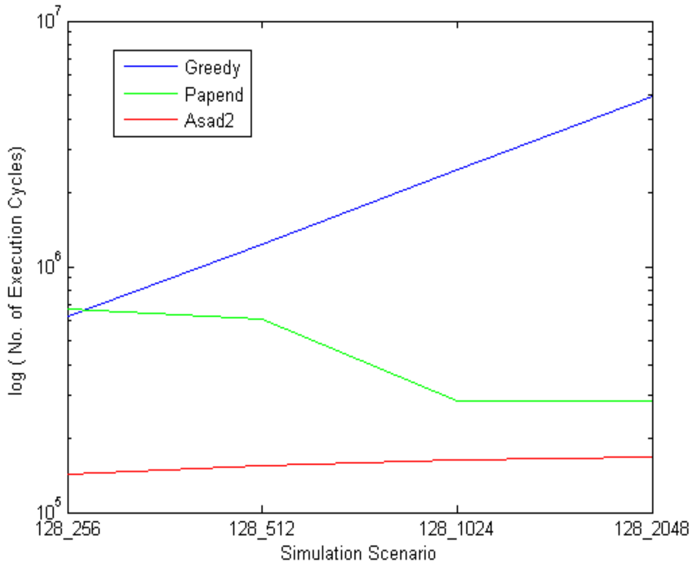
Processor Parameters	Default Values	Cache Parameters	Default value
Decode width	4	Basic cache parameters: Associativity / Block size / Nbr. of sets / replacement policy	Data: 128/32/4/l Inst: 512/32/1/l
Nbr. of int mult/dividers	1		
Nbr of L1 cache ports	2		
Nbr of floating point ALU	4		
Nbr of floating point mult/dividers	1		
Max issue width	4	Hit latency of L1 cache	1 cycle
Capacity of the RUU in instructions	16	Hit latency of L2 cache	6 cycles
Capacity of the load/store queue	8	Main memory access latency	18 cycles
Fetch width	4	Main memory width	8 bytes

Simplescalar microprocessor which generates simplescalar binary files. The simplescalar microprocessor models an out-of-order superscalar architecture based on a RUU (Register Update Unit) . The RUU exploits a reorder buffer to automatically rename registers and hold the results of pending instructions. However, completed instructions are retired in program order to the register file. The microarchitecture supports speculative execution. The memory system uses a load/store queue and a rich set of cache memories with tunable sizes is also available. The detailed description of the Simple-Scalar tool alongwith its Superscalar MIPS architecture will be given in chapter 6, where we will adapt different processor parameters as a mean to further reduce the running-time of the bit-loading algorithm. In this context we keep the processor parameters constant for all the three compared algorithms so as to perform an exact execution-cycles based complexity comparison of the concerned algorithms.

2.7.4.4 Comparison of Number of Execution-Cycles

The simplescalar architecture parameters were kept same for all of the simulations. Asad1 indicates our earlier proposed method [21] where *sorting* was used in the final allocation, whereas Asad2 represents our improvement in the algorithm where sorting is replaced with the *interval-classification* based final allocation, as presented in [22] as well. Figure 2.15 shows the number of execution cycles taken by different algorithms for the case where the number of subcarriers (N) is increased but the number of bits/subcarrier for the non-adaptive case is kept constant. The horizontal axis shows various simulation

Figure 2.14 No. of Execution cycles for different algorithms for different simulation scenarios



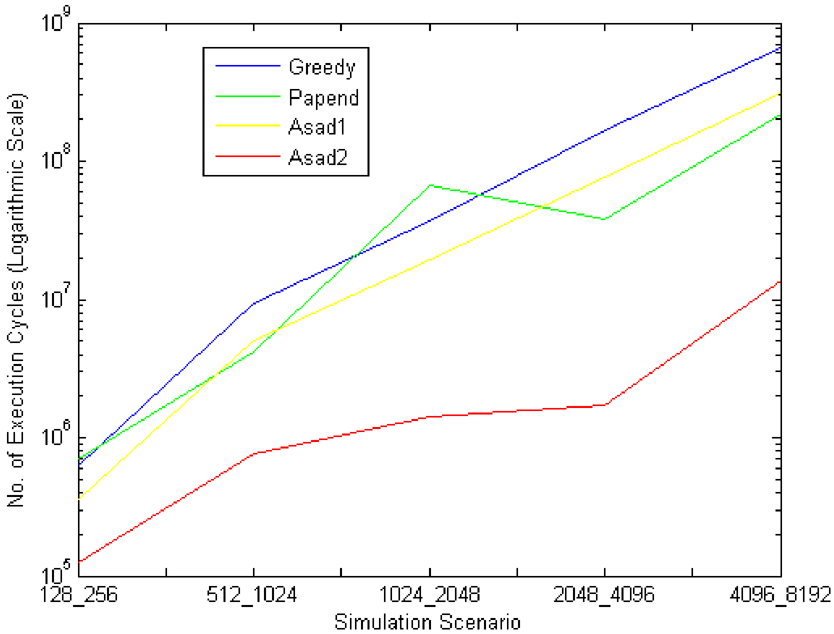
scenarios where the first number is for the number of subcarriers (N) and the next one indicates the total number of bits to be allocated assuming a QPSK modulation over all the subcarriers when there is no adaptation. It is evident from the figure that for all of the simulation scenarios, our improved algorithm takes a lot less number of execution cycles.

Figure 2.14 shows the case where the number of subcarriers is kept constant but the total number of bits to be allocated (B) is varied. As evident our proposed algorithm not only takes a lot less number of execution cycles than other algorithms but also that the number of execution cycles is almost independent of the number of bits to be allocated.

2.8 Conclusion

Adaptive resource allocation has established itself as a strong mean of enhancing a system's performance while making the best-possible use of system's resources at the same time for a given system and channel. Discrete Bit-Loading has played a major role in the revolutionary performance measures by the DSL and hence now making its way in the wireless applications of multicarrier systems (WiFi, WiMax, UWB etc.). However, for discrete bit-loading to be employed in the fast time-varying wireless scenario, its algorithm

Figure 2.15 No. of Execution cycles for different algorithms for increasing Number of Subcarriers but same Bits per Subcarrier ratio



not only has to be of reduced complexity, but also flexible enough to make the run-time performance-to-complexity trade-off decisions with respect to the evolving demands and constraints of the system.

In this direction, where a large number of optimal/sub-optimal algorithms have been proposed in the recent past, we have made the contribution of proposing an algorithm, which to our knowledge converges towards the optimal with the least complexity involved than any other existing optimal-converging algorithm. Like most of the existing algorithms, we have also made use of the 'Gap approximation' phenomenon, which has established itself as a close approximation to the capacity while providing adequate simplification to the capacity analysis for practical systems. By making use of the Gap Approximation, the algorithm we designed, is based on the classification of all the subcarriers into subgroups of 3-dB, with respect to their channel to noise ratios (SNR) or the corresponding bit-incremental powers. It was discovered that an inherent pattern of allocation is present underneath the classical optimal Hughes-Hartogs allocation procedure. This underlying pattern was brought into use for devising the 3-dB subgroup based allocation algorithm.

It was found that a pattern of allocation not only exists for the bit-allocations that take place across different 3-dB subgroups but also for the allocations within a single 3-dB sub-group based on which each 3-dB sub-group was further divided into a number of *intervals*. The algorithm consists of two parts of initial and final allocation, where the initial allocation makes use of the rythm/pattern of allocation that exists across different sub-groups while the final allocation makes use of the rythm/pattern of allocations that exist across different intervals of a subgroup.The resultant bit-profile is the very same as obtained through the optimal Hughes-Hartogs [10] algorithm.

The algorithm complexity was not only compared with the classical optimal Hughes-Hartogs solution, but also to that of a recently proposed discrete bit-loading algorithm by Papendreou et al. [15]. Our proposed algorithm was found distinctively less complex than the rest of the algorithms. To verify the theoretical complexity analysis of the three algorithms, we have compared the running-time of the three algorithms in terms of the actual number of execution-cycles over a Superscalar processor by means of SimpleScalar tool. The comparison of the execution cycles of the three algorithms, as presented in this chapter, verify the significant reduction in the complexity of our algorithm.

An interesting feature of the 3-dB subgroup based algorithm is the flexibility it offers for a possible performance to complexity tradeoff. Using the same methodology, the subgrouping can be done for larger chunks, i.e. 6-dB so as to use the same modulation-size across a wide-block of subcarriers that can further reduce complexity but at the cost of performance. Also, inherently the Rate-Maximization based allocation also follows the same rythm of bit-allocation, though the constraint in that case is the total available energy instead of total number of bits to be allocated. Hence the same sub-grouping methodology can be easily adopted for its use in the Rate-Maximization based allocation.We also extended our analysis to the peak-power constrained systems, where a condition of feasibility for a given peak-power constraint corresponding to a given number of bits to be allocated was developed.

Finally this work can be extended in a number of interesting directions. Although we developed the algorithm for a single-user case, the same approach can be extended for a multi-user scenario. The above-mentioned direction of complexity-to-performance tradeoff can be taken as well and the subgrouping methodology can be modified so as to take into account the run-time variation in the needs and constraints of the underlying system and channel. Although the performance of the Gap Approximation factor has been established in the literature for a given set of modulation and coding schemes, it can be extended to other sets as well, so as to quantify the difference when no approximation is performed. Finally a complete framework involving along-with adaptive modulation (bit-loading), other modes of adaptation like that of adaptive power-allocation and adaptive-coding and making use of this 3-dB subgrouping methodology can be developed as well. This can be based on the run-time sub-grouping of the channel SNRs with respect to the changing system characteristics, and hence this methodology can be used for multi-mode transceivers or cognitive-radios that are the future of transceiver-technology.

Chapter 3

Optimal Power Allocation Algorithm for Peak-Power Constrained Multicarrier Systems

3.1 Introduction

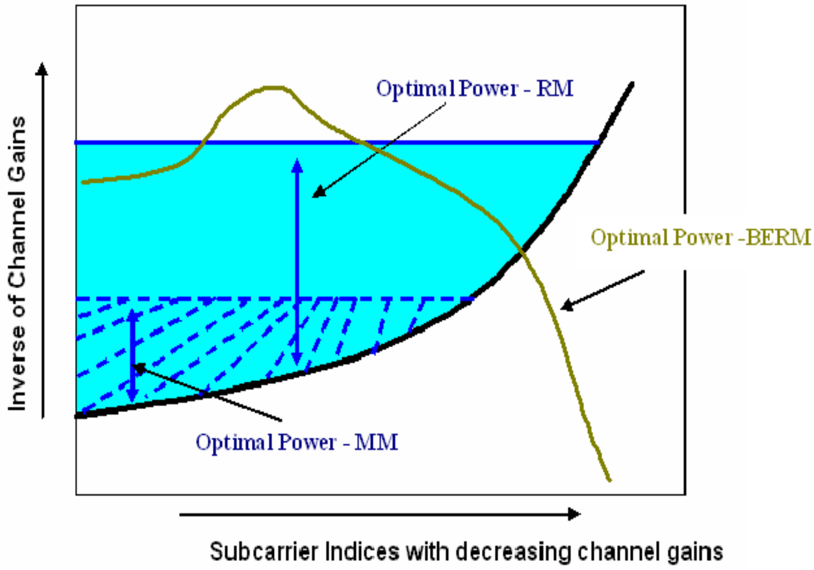
We established earlier that one of the biggest advantages linked with multicarrier communications is the ability to fine tune the system parameters at different frequencies, with respect to the corresponding channel gain, because of the division of system's bandwidth spectrum into a large number of subcarriers. This *tuning* of a subcarrier can be done by a number of means due to the presence of a number of parameters/characteristics associated with a subcarrier. The set of the basic parameters associated with a particular subcarrier consists of the no. of allocated bits b_n (=modulation size), the amount of allocated power (energy) e_n , the redundancy (=coding rate) and the resulting BER of a subcarrier. These parameters are interlinked depending on case-to-case via famous relations e.g. like that of the Shannon's capacity equation or the probability of error equation linked with a particular modulation and coding scheme. We showed in chapter 2 that the analytical expression for the probability of error for a modulation and coding scheme can be represented in the form of the classical Shannon's capacity expression by means of a Gap Approximation factor, which the application of of theoretical analysis for practical purposes.

Generally, the goal of adaptive algorithms is to find the best distribution of one or more of these parameters over all the subcarriers in order to optimize a particular attribute of the overall system e.g. total energy $E = \sum_{n=1}^{N} e_n$, total no. of allocated bits $B = \sum_{n=1}^{N} b_n$ or aggregate BER = BER_{avg} with constraints on any number of the rest of attributes. In chapter 2, we dealt with the best distribution/allocation of the parameter b_n so as to optimize(minimize) the overall system attribute of total energy, with the strict restriction of b_n taking up only discrete positive integer values. Although this discrete constraint is true for the optimization of the parameter b_n, optimizing the power/energy (e_n) distribution can be done free of this constraint and hence powerful optimization techniques that exist for continuous variables can be adopted in this context. The goal of this chapter is hence to employ one such classical continuous-variable optimization techniques to find the best distribution of the parameter e_n, taking into account our particular set of goals and constraints.

An important remark to make at this point is regarding the effect of the presence of a constraint on a parameter on the overall optimization problem. When individual constraints are not present on any one parameter (e.g. e_n or b_n), then the optimal allocation of any of these parameters dictates the resulting distribution of the other parameters as they are interlinked by means of famous relations as that of Shannon's capacity equation or the analytical expression defining the probability of error for a given system and channel specifications. An example of this behaviour is shown by the waterfilling solution where the discrete parameter b_n distribution follows the optimal distribution of the continuous variable e_n. However, as we will see in this chapter that a change in the goals and constraints of the optimization problem leads to a change in the resulting optimal distribution profile of a given parameter. Unlike waterfilling, where the goal is to maximize the system capacity/throughput, our objective in this chapter will be to optimize the power/energy distribution of a total energy amount of E amongst different subcarriers so as to minimize the overall system BER, if modulation-type is not varied across different sub-carriers . This possible difference in the resulting energy profile can be represented by means of the figure 3.1:

Figure 3.1 depicts varied behaviour of energy allocation, each for a unique set of constraints and goals where RM, MM and BERM denote Rate-Maximization, Margin-Maximization and BER-Minimization Problem respectively. In general, for each combination of the constraints on the parameters (i.e. fixed/variable e_n,b_n or BER_n) associated with a subcarrier, the corresponding set of RM, MM and BERM solutions will be of the

Figure 3.1 The Optimal Power Allocation Response with respect to different objectives and constraints



same nature. For example, generally the RM and MM solutions corresponding to the class of constraints where both the e_n and b_n are allowed to be varied take the form of the Classical Water-filling solution, though the *Water-Level* could be different as depicted in figure 3.1.

This same nature of the solutions for different optimization problems is commonly known in literature as the *Duality* that exists between different solutions and which refers to the fact that a solution which is optimal with regards to a particular optimization problem will also be optimal with respect to the other (dual) optimization problem. RM and MM are widely known in literature as the dual-problems [81]. Hence if RM solution given by the waterfilling approach results in a total allocation of B bits, then the MM solution for a target no. of bits equal to B will result in the same solution as that of RM, as we explained in the previous chapter that in fixed throughput optimal allocation, the water is poured only till B is achieved. The duality between RM and MM problems can be furthered to BERM problems as well and the concerned proofs were given by Piazzo et al. [12].

Different possible classes/combinations of constraints can be given by means of the table 3.1.

Some recent research, as will be cited in the next section, has been dedicated to find the best combination of a sub-carrier's parameters that should be varied so as to arrive at

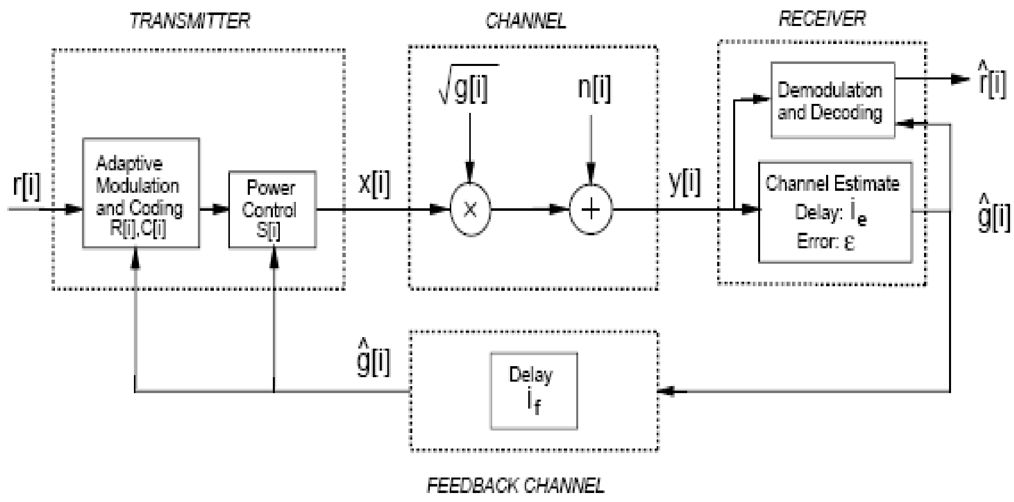
Table 3.1: Different combinations of constraints on Sub-carriers

	Channel Gain (h_n)	Energy Allocated (e_n)	Bits allocated (b_n)	BER (BER_n)
1	Variable	Variable	Variable	Constant
2	Variable	Variable	Constant	Constant
3	Variable	Constant	Variable	Constant
4	Variable	Constant	Constant	Constant
5	Variable	Variable	Variable	Variable
6	Variable	Variable	Constant	Variable
7	Variable	Constant	Variable	Variable
8	Variable	Constant	Constant	Variable

the optimal compromise between performance and complexity. Our aim in this chapter is to explore the optimal distribution/allocation of the parameter e_n if b_n is assumed to be constant over all the sub-carriers, so as to minimize the overall system BER. This will be different from the classical optimization problem approach where both the modulation scheme (b_n) as well as the allocated-energy (e_n) are allowed to be varied.

The BER-Optimal energy distribution solution will then be viewed in presence of the Peak-Power/Energy constraint, which due to the reasons of bandwidth scarcity and interference limitation is present in state-of-the-art wireless (UWB) and wireline (DSL) systems. This Peak-Energy per subcarrier constraint is defined in terms of the Power-Spectral-Density Mask, which dictates the maximum power to be transmitted at each frequency within the system bandwidth, and since these regulations are subject to changes from region to region, modifying existing resource allocation theory and algorithms so as to take into account this additional constraint of Peak-Power is of particular interest. Finally, an efficient algorithm for energy-allocation that respects the peak-energy limit will be presented and its simulation times compared with those of the classical solution of Iterative-Waterfilling, so as to establish the computational complexity advantage of our

Figure 3.2 A communication system with adaptive power allocation



algorithm.

Next section will give the state-of-the-art on different power/energy optimization problems and the corresponding proposed solutions. Works related to different combinations of hybrid rate-power allocations along with recent works and algorithms for power/energy allocation with the peak-power/energy constraint will also be mentioned. In the next section we will introduce the classical Lagrange Optimization technique which will be used in section 4 of this chapter for our particular BER-minimization problem. Extension of analysis for BER-Optimization to the Peak-Energy constraint will also be given in the same section. In section 5, we propose a computationally efficient energy allocation algorithm for peak-energy constrained multicarrier systems and its reduced complexity will be verified by means of comparison with the classical Iterative Waterfilling approach in the same section and finally conclusion and perspectives on the work will be given in the last section.

3.2 State of the Art on Power Allocation Schemes

The paradigm of adaptive transmission exists for a good number of years and hence a handsome amount of literature is present related to theoretical as well as implementation oriented aspects of power-allocation methodologies which have been proposed for different systems, taking into account their respective goals and constraints. In this section we will emphasize on the fundamental theoretical works related to power allocation on multicarrier systems as well as the works related to our particular set of goal (BER minimization) and constraints (e.g. Peak-Power constraint).

3.2.1 Works on Theoretical Foundations

Although, the *waterfilling* solution has been long known in the information theory domain, often attributed to R. Gallager [9] for his optimized power distribution proposal over a colored gaussian channel, it was Kalet [27] who first theoretically demonstrated that the optimal power distribution for capacity maximization over the multi-carrier channel takes the *waterfilling* form as well. A detailed work outlining the optimal distribution of different subcarrier parameters such as e_n and b_n, in both discrete and continuous form, with respect to different criteria and channels was given by Willink et al. [82] and it was shown that with optimal distribution of different parameters, a multicarrier system clearly outclass the equalized single-carrier system in different SNR ranges. Also, Ligdas et al. [83] provided the theoretical foundations for the power-allocation problem when practical constraints as that of the feedback delay and channel estimation error are taken into account. Notable theoretical foundations related to the adaptive optimization of variable system parameters for a generalized communication system and channel were outlined in the doctoral dissertation of Goldsmith [84] and [85].

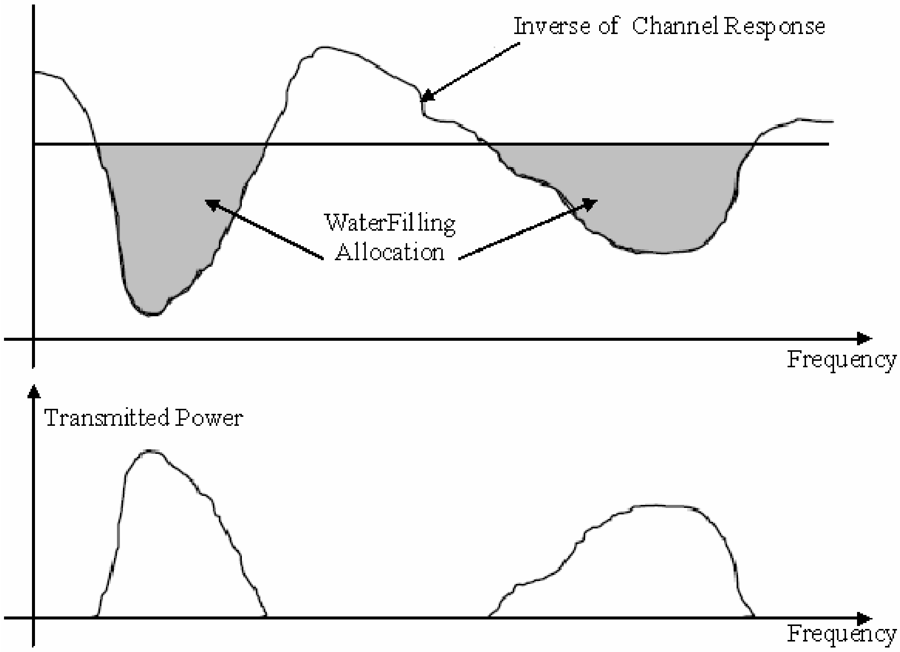
3.2.2 Works on BER-Optimal Power Allocation

The optimal power distribution on a multicarrier system for the Rate/Capacity Maximization problem results in the *Waterfilling* distribution as discussed earlier and as depicted in the figure 3.3

The intuition behind water-filling is to take advantage of good channel conditions: when channel conditions are good, more power and a higher data rate is sent over the channel. As channel quality degrades, less power and rate are sent over the channel. If the instantaneous channel gain falls below the cutoff value (defined by total power to be allocated and the channel conditions), the channel is not used.

These schemes of increasing capacity may be suitable for variable-rate services, such as e-mail and web browsing. On the contrary, delay-sensitive services, such as voice or video, are usually provided at a fixed rate [18]. In these applications, it is desirable to design a transmit power allocation scheme that improves error-rate performance for a given

Figure 3.3 Optimal power allocation (Waterfilling Distribution) for Rate-Maximization problem



fixed rate. A number of studies have been performed to investigate the characteristics of the power distribution, when optimized with respect to the criterion of BER.

First concise work on the BER-optimal power allocation over a fading channel was performed by Ligdas et al. [83], who investigated the impact of continuous as well as finite-state power control over a slow fading channel using the dynamic programming approach. In the multicarrier context, Goldfeld et al. [24] employed the *Lagrange Optimization* to quantify the optimal power allocation over different subcarriers by minimizing the aggregate BER of a multicarrier system under the total power constraint and under the assumption that same constellation is used in each subcarrier. The optimal solution was finally simplified into a quasi-optimal solution, where the performance of the quasi optimal solution was found to be practically same as that of the optimal solution but with much less complexity involved. The Quasi-Optimal allocation was found to be similar in performance to the Equal-SNR based power allocation at high SNR regions.

A good analysis on the characteristics of the BER optimized power allocation for the multicarrier systems was presented by Chang et al. [25], again by making use of the Lagrangian multipliers. Chang observed that for an M-ary phase-shift keying (PSK) or M-ary quadrature amplitude modulation (QAM), the exact or approximate BER function may be expressed as a Q-function [29]. However, since solving Lagrange requires differentiation of the objective function, it may not be possible to find a closed-form solution, using the Q function to describe the error probability function. In this case, he proposed utilization of an adaptive method, such as the steepest-descent algorithm [?] to find a solution in an iterative manner. However, since such algorithms are high in complexity, Chang employed the exponential upper bound [26] to the error probability function and optimized it under the total power constraint. It was observed that the characteristics of the BER optimized power allocation scheme differ from those of the waterfilling [27] scheme that maximizes the capacity, as at high SNR range, the BER optimized scheme was found to allocate more power to the more attenuated subchannel, which is contrary to the behavior of the waterfilling. Finally, recently BER-Optimal power allocation in multicarrier systems was investigated in the context of MIMO multiplexing [86], where the power-optimized transmissions were shown to have performance gains with respect to the classical MIMO receivers. Another important work relating the dual relationship that exists between optimization solutions with respect to different criteria (BER, power, throughput etc.) was done by Piazzo et al. [12], focusing on the commonly existing systems with constraints such as that of equal power constraint on all the subcarriers or that of the equal BER constraint on all the subcarriers. Allocations optimal with one criterion were proven to be optimal with another criterion establishing the duality relationship between them and hence a class of *Global Optimal* solutions was finally derived for a particular set of goals and constraints.

3.2.3 Works Related to Power Allocation with Peak-Peak Constraint

A practical constraint on any transmitter is that the peak power cannot exceed a certain level, which may vary from system to system. This type of constraint, referred to as peak-power constraint, may be due to hardware limitations, human safety concerns or dictated by regulatory authorities to avoid interference amongst systems sharing the same chunk of bandwidth. This constraint may impose severe limitations to the maximum gain (in terms of reduced BER) achievable by power control. Since many new wired and wireless systems are being subjected to peak-emission power related regulatory measures in-order to incorporate such large number of systems in scarce bandwidth, it is very important to consider this peak-power constraint in the overall optimization problem.

One of the first works exploring the impact of peak-power constraint on the use of adaptive power allocation in fading channels was done by Choi [75] by making use of Lagrange multipliers. Both, the discrete as well as the continuous constellation size case were evaluated and an iterative algorithm for peak-power constrained adaptive modulation was proposed. Regarding the works with respect to the peak-power constrained optimization problem over a multicarrier system, Baccarelli et al. [20] analytically solved the Rate-Maximization problem employing the lagrange method and also proposed the *Iterative Waterfilling* method to optimally allocate the power over a set of subcarriers such as the peak-power constraint over a group of subcarriers is respected as well. Recently, Papendreou et al. [74] analyzed both the Rate-Maximization and the Margin Maximization problem with respect to the additional peak-power constraint and a number of propositions were made so as to simplify the peak-power constrained power-allocation procedures. Palomar et al. [87] gave the unified view of different class of waterfilling algorithms including the proposal of efficient algorithms for a range of waterfilling problems.

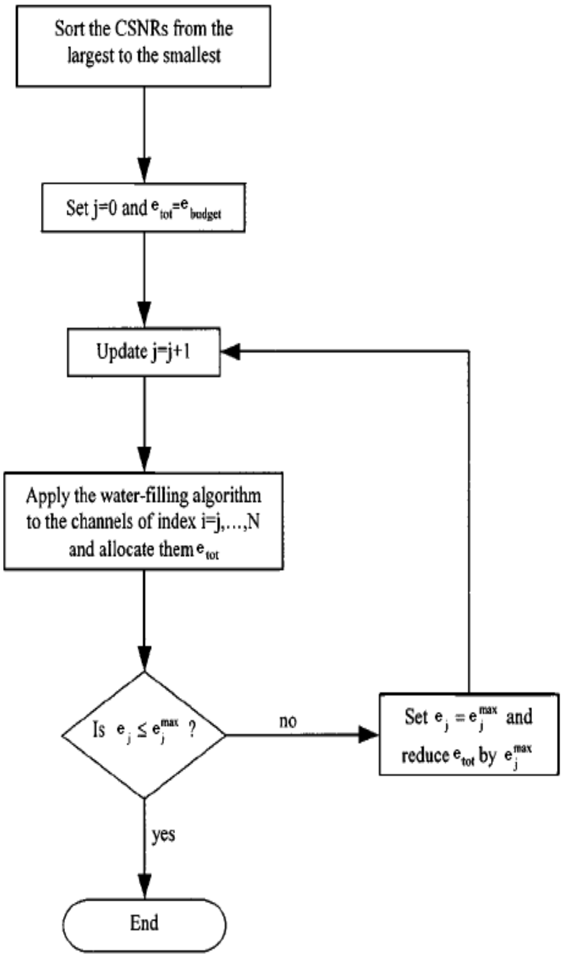
3.2.4 Iterative Waterfilling Algorithm

As mentioned in the previous sub-section that for the capacity-optimal peak-power constrained power allocation, Baccarelli et al. proposed an *Iterative Waterfilling* approach. Since we tend to reduce the computational complexity of this algorithm in this chapter, which might eventually be used for peak-power constraint conforming power distribution optimal with respect to any given criterion (capacity, BER etc.), it is important that we outline the basic principle of the Iterative Waterfilling algorithm as given by Baccarelli et al.

We denote the energy allocated to subcarrier j as e_j and e_j^{max},e_{budget} and e_{tot} as the maximum permissible energy to be allocated to subcarrier j, the total amount of energy to be allocated and the total energy allocated at a particular step, respectively. It was observed that though the eventual energy allocation is not water filling in its true meaning, as some subcarriers' energies will be clipped to e_j^{max}, nevertheless, the algorithm was named as Iterative waterfilling because of the iterated applications of the standard water-filling algorithm to decreasing-size subsets of the overall channels, as indicated by the flowgraph in the figure below.

Using the case where the peak-energy constraint is same over all the subcarriers i.e.

Figure 3.4 Flowgraph of Iterative Waterfilling Algorithm



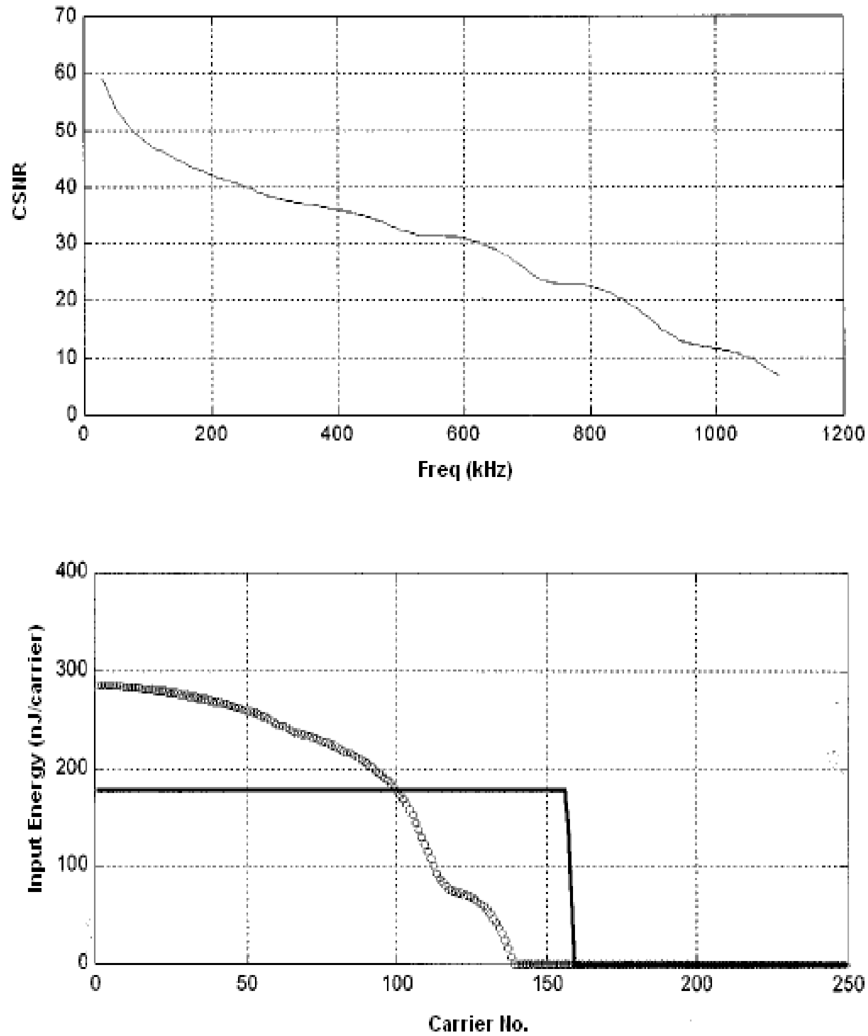
$e_j^{max} = e^{max}$, Baccarelli made use of the observations that when the power allocation is optimized with respect to capacity maximization, the channels candidates to receive the maximum energy level are those with the largest Channel to Signal Noise Ratios (CSNRs), whereas the links that should be turned off possess the lowest CSNRs. At the same time, channels whose energies do not cross the e_j^{max} limit are allocated energy according to a standard water-filling type algorithm.

Thus, the basic form of Iterative Waterfilling algorithm as given by Baccarelli [20] starts with sorting of the CSNR from largest to smallest followed by the application of the water-filling procedure to the set of N subchannels, so to allocate overall available energy e_{budget}. After execution of the water-filling algorithm, the energy so obtained for the first subchannel i.e. e_1 is compared to e^{max}. If e^1 does not exceed e^{max}, then the overall procedure ends, and the resulting distribution for the input energies coincides with that achievable via the standard water-filling procedure. On the contrary, when e^1 is greater than e^{max}, the energy of subcarrier 1 is clipped to e^{max} , the available total energy is reduced by e^{max} and the water-filling procedure is performed again, this time operating only on the remaining subchannels of index i ranging from 2 to N and attempts to allocate to them a total energy equal to e_{budget}. The above procedure is repeated till no subcarrier's energy as obtained by the waterfilling procedure exceeds the permissible limit of e^{max}. The finally allocated energies over the subcarriers for the DSL channel alongwith the peak-energy constraint is shown in the figure 3.5 [20]. The upper figure represent the CNR for different subcarriers, while the lower figure shows the energy allocation profile at respective subcarriers with (dark-line) and without (broken-line) taking ino account the peak-power constraint.

3.3 Using Lagrange Multipliers for Convex Optimization

A large number of optimization techniques/methodologies and tools is available in the literature, where a particular technique is employed for finding the optimal solution to a problem depending upon the characteristics of the objective function, constraints involved and the nature (e.g. discrete/continuous) of the variables involved in the given problem. *Lagrange Multipliers* based optimization is commonly employed for the class

Figure 3.5 Channel Response Alongwith the Peak Power Constraint and the Allocated Energies over a DSL Channel



of *constrained* optimization problems and the optimality of the solution can be verified *Convex* by making use of the Karush-Kuhn-Tucker (KKT) conditions, which will be introduced below.

Most the objective functions to be optimized (Throughput, Error Probability etc.) in the physical layer resource allocation problems of a communication system can be expressed in the form of a convex function. Since the objective of this chapter is the optimization/minimization of the error probability function, where we will demonstrate later that the error probability function can be conveniently approximated in terms of a convex function, the power optimization problem gets converted into a constrained convex optimization problem, which can be conveniently solved by means of the Lagrangian method. Hence, we will outline the basic characteristics and formulation of the Lagrange optimization technique in this section and for the details we refer the reader to [28]

3.3.1 Lagrangian Problem Formulation

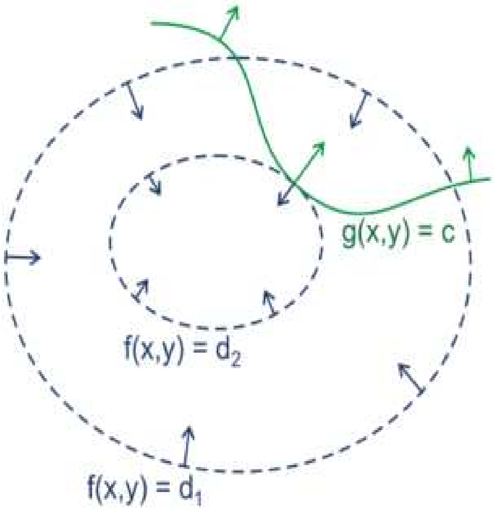
The method of Lagrange multipliers is used for finding the extrema of a function of several variables subject to one or more constraints; it is the basic tool in nonlinear constrained optimization. Simply put, the technique is able to determine where on a particular set of points (such as a circle, sphere, or plane) a particular function is the smallest (or largest).

More formally, Lagrange multipliers compute the stationary points of the constrained function. It reduces finding stationary points of a constrained function in n variables with k constraints to finding stationary points of an unconstrained function in $n+k$ variables. The method introduces a new unknown scalar variable (called the Lagrange multiplier) for each constraint, and defines a new function (called the Lagrangian) in terms of the original function, the constraints, and the Lagrange multipliers.

Consider a two-dimensional example case as depicted by figure 3.6 [28]. Suppose we have a function $f(x, y)$ we wish to maximize or minimize subject to the constraint $g\left(x, y\right) = c$, where c is a constant. We can visualize contours of f given by $f\left(x, y\right) = d_n$ for various values of d_n, and the contour of g given by $g(x, y) = c$.

Suppose we walk along the contour line with $g = c$. In general the contour lines of f and g may be distinct, so traversing the contour line for $g = c$ could intersect with or cross the contour lines of f. This is equivalent to saying that while moving along the contour line for $g = c$ the value of f can vary. Only when the contour line for $g = c$ touches contour lines of f tangentially, we do not increase or decrease the value of f :

Figure 3.6 Graphical representation of the optimization problem of objective function $f(x, y)$ under the constraint function of $g(x, y) = c$



that is, when the contour lines touch but do not cross.

This occurs exactly when the tangential component of the total derivative vanishes: $df_{\|} = 0$, which is at the constrained stationary points of f (which include the constrained local extrema, assuming f is differentiable). Computationally, this is when the gradient of f is normal to the constraint(s): when $\nabla f = \lambda \nabla g$ for some scalar λ. Note that the constant λ is required because, even though the directions of both gradient vectors are equal, the magnitudes of the gradient vectors are most likely not equal.

Geometrically we translate the tangency condition to saying that the gradients of f and g are parallel vectors at the maximum, since the gradients are always normal to the contour lines. Thus we want points (x, y) where $\nabla_{x,y} f = \lambda \nabla_{x,y} g$ and further $g(x, y) = c$. To incorporate both these conditions into one equation, we introduce an unknown scalar λ, and solve

$$\nabla_{x,y,\lambda} F\left(x, y, \lambda\right) = 0 \tag{3.1}$$

with

$$F\left(x, y, \lambda\right) = f\left(x, y\right) + \lambda\left(g\left(x, y\right) - c\right) \tag{3.2}$$

and

$$\nabla_{x,y,\lambda} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial \lambda}\right) \tag{3.3}$$

where the function $F(x, y, \lambda)$ is termed as the *Lagrangian* function and is solved according to equation 3.1 so as to find the optimal solution.

3.3.2 Duality and Karush-Kuhn-Tucker Conditions

Lagrangian function is mostly solved by means of its alternate representation in the form of a *dual* function. Given a convex optimization problem in standard form

$$min \quad f_0(x) \tag{3.4}$$

$$\begin{aligned} f_i(x) &\leq 0,\ i \in \{1, \ldots, m\} \\ h_i(x) &= 0,\ i \in \{1, \ldots, p\} \end{aligned} \tag{3.5}$$

the Lagrangian function L for this problematic is defined as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{3.6}$$

The vectors λ and ν are called the dual variables or Lagrange multiplier vectors associated with the problem. The Lagrange dual function $g(\lambda, \nu)$ is then defined as the minimum value of the Lagrangian over x:

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \tag{3.7}$$

It is easy to show [28] that the dual function yields lower bounds on the optimal value p^* of the problem in equation 3.5: For any $\lambda \geq 0$ and any ν we have

$$g(\lambda, \nu) \leq p^* \tag{3.8}$$

Thus we have a lower bound that depends on some parameters λ, ν. In order to find the best lower bound that can be obtained from the Lagrange dual function, the original optimization problem in 3.5 is transformed as follows into its *dual* representation

The novel optimization problem states:

$$\begin{aligned} maximize \quad & g(\lambda, \nu) \\ subjectto \quad & \lambda > 0 \end{aligned} \tag{3.9}$$

This problem is called the ***Lagrange dual problem*** associated with the problem in 3.5 . In this context the original problem is sometimes called the primal problem.

The optimal value of the Lagrange dual problem, which if denoted by d^*, is, by definition, the best lower bound on p^* that can be obtained from the Lagrange dual function. In particular, this can be expressed by means of the inequality $d^* \leq p^*$, which holds even if the original problem is not convex. This property is called ***weak duality***. The difference $p^* - d^*$ is termed as the optimal *duality gap* of the original problem, since it gives the gap between the optimal value of the primal problem and the best (i.e., greatest) lower bound on it that can be obtained from the Lagrange dual function. The optimal duality gap is always nonnegative. If the optimal duality gap is zero, then we say that strong duality holds i.e. $d^* = p^*$. This means that the best bound that can be obtained from the Lagrange dual function is tight.

Strong duality does not, in general, hold. But if the primal problem is convex, strong duality is usually present and as we will show below that the presence of the strong duality leads to the development of Karush-Kuhn-Tucker (KKT) conditions of optimality, which helps solve the *dual* problem and eventually the concerned *primal* optimization problem.Let x^* and (λ^*,ν) be any primal and dual optimal points with zero duality gap. Since x^* minimizes $L(x, \lambda^*, \nu^*)$ over x, it follows that its gradient must vanish at x^*, i.e.,

$$
\begin{aligned}
\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i \nabla h_i(x^*) = 0 \qquad (3.10)\\
\lambda_i \geq 0 \ (i = 1, \ldots, m)\\
f_i(x^*) \leq 0, \ \text{for all } i = 1, \ldots, m\\
h_i(x^*) = 0, \ \text{for all } i = 1, \ldots, p\\
\lambda_i f_i(x^*) = 0 \text{ for all } i = 1, \ldots, m. \qquad (3.11)
\end{aligned}
$$

which are called the Karush-Kuhn-Tucker (KKT) conditions. To summarize, for any optimization problem with differentiable objective and constraint functions for which strong duality obtains, any pair of primal and dual optimal points must satisfy the KKT conditions [28]. Also when the primal problem is convex, the KKT conditions are also sufficient for the points to be primal and dual optimal, which eventually allows us to compute a primal optimal solution from a dual optimal solution, which becomes especially interesting for the cases where the dual problem is easier to solve than the primal problem, as will be demonstrated in the next section to solve our concerned optimization problem of BER minimization.

3.4 BER-Optimized Power Allocation

3.4.1 BER minimization problem

Based upon the notations describing the various parameters of a multicarrier system in a frequency-selective environment, as given in chapter 2, the BER minimizing optimization problem under constraints of fixed constellation size and fixed total power, can be formally expressed as below.

$$min \quad p^{err}_{avg} = \frac{1}{N} \cdot \sum_{n=1}^{N} p^{err}_{n} = \frac{1}{N} \cdot \sum_{n=1}^{N} f_{BER}(e_n.h_n) \quad \forall\, 0 \;\geq\; p^{err}_{n} \;< 1; \tag{3.12}$$

such that

$$\begin{aligned} E_{total} &= \textstyle\sum_{n=1}^{N} e_n \quad e_n \in \Re^{+} \\ b_n &= b_{cnst} \quad \forall\, n \\ e_n &\leq e^{max} \quad \forall\, n \end{aligned}$$

where p^{err}_{n},e_n,h_n and b_n represent the probability of error, allocated energy, channel gain and allocated number of bits on subcarrier n respectively and e^{max} denotes the Peak-energy constraint and is also represented by $\overline{e}$. $f(e_n.h_n)$ is the function that defines the error probability on subcarrier n and is dependent on e_n and h_n. The problem is to find how much energy/power (e_n/p_n) must be allocated to each of the subcarriers such that the resulting BER is minimized, the sum of the energies in the subcarriers is equal to the total available energy with the constraint that modulation remains the same over all the subcarriers.

The solution to the above problem gives the optimal distribution of the total power amongst all of the subcarriers such that the resultant aggregate BER is minimized and hence the objective function is the average BER of the system. The probability of error for M-ary QAM or M-ary PSK system may be represented exactly or approximately by the Q-function [29] which is convex in nature. Also, a subchannel's BER is a function only of the power allocated to it and independent of the characteristics of other channels. The total BER of the system can then be represented as a sum of separable functions as used in the equation above. This, combined with convexity, classifies the problem as a continuous separable convex resource allocation problem which allows the use of Lagrange multipliers and the KKT conditions to characterize the optimal solution. The constrained optimization problem given above can thus be converted into the unconstrained optimization problem by means of Lagrange Multipliers, and readily solved using the KKT optimality conditions, as exhibited below.

3.4.2 BER-Optimized Power Allocation with Peak-Power Constraint

As stated earlier, lagrange optimization is the most natural method to find the optimal distribution of e_n which minimizes the convex objective BER function with the constraints as given in equation 3.12. Transforming the optimization problem in 3.12 into the standard Lagrange optimization problem form

$$\begin{aligned} minimize \;\; p_{avg}^{err} &= \frac{1}{N}\sum_{n=1}^{N} f\left(h_n \cdot e_n\right) \\ subject \;\; to \quad \sum_{n=1}^{N} e_n - E_{total} &= 0 \\ e_n - \overline{e} &\leq 0 \end{aligned} \tag{3.13}$$

Introducing λ as the lagrange multiplier for the equality constraint and ν_n as the multipliers for the inequality constraints, the lagrangian function may be expressed as:

$$\begin{aligned} L\left(e_n, \lambda, \nu_n\right) \;\; = & \\ & \frac{1}{N}\sum_{n=1}^{N} f_{BER}\left(h_n \cdot e_n\right) + \lambda\left(\sum_{n=1}^{N} e_n - E_{total}\right) + \nu_n\left(e_n - \overline{e}\right) \end{aligned} \tag{3.14}$$

The above optimization problem involving both equality and inequality constraints can be solved using the KKT conditions [28], whose solution are those of the above optimization problem. Solving KKT conditions involves differentiating the objective function, which is the BER function $f_{BER}\left(h_n \cdot e_n\right)$ in our case. As we mentioned that for M-ary PSK or M-ary QAM, the exact BER function is generally expressed as a Q-function [29].

It would be extremely difficult to find a closed form solution, if the Q-function for exact BER is used in the Lagrangian function. Therefore it is more appropriate to use a simple approximation of the BER rather than the exact BER expression. For M-ary QAM, the exact BER function may be approximated using an exponential upper bound [26]:

$$f_{BER}\left(h_n e_n\right) \cong aQ\left(\sqrt{b h_n e_n}\right) \leq \frac{a}{2}\exp\left(\frac{-b}{2}h_n e_n\right) \tag{3.15}$$

where $Q\left(x\right) = 1/\sqrt{2\pi}\int_x^\infty \exp\left(-t^2/2\right)dt$ represents the Q-function, while variables $a = 2(\sqrt{M}-1)/\sqrt{M}\log_2\sqrt{M}$, and $b = 3/(M-1)$ depend on the modulation size(M). Chang [25] used these BER bounds to develop a BER optimized energy allocation strategy. A good analysis of the characteristics of such energy allocation can thus be found in [25]. But we want to use these BER bounds to develop a simplistic BER optimized energy allocation algorithm which would respect the additional peak-power constraint as well.

The KKT conditions [28] for the Lagrangian given in eq. 3.15 can be presented as

$$\begin{aligned} e_n^* - \overline{e} &\leq 0\,; \qquad (3.16)\\ \sum_{n=1}^{N} e_n^* - E_{total} &= 0\,;\\ \nu_n^* &\geq 0\,;\\ \nu_n^*\left(e_n^* - \overline{e}\right) &= 0\,; \qquad (3.17) \end{aligned}$$

$$\frac{d}{de_n^*}\left[\frac{1}{N}\sum_{n=1}^{N} f_{BER}\left(h_n e_n^*\right) + \lambda^*\left(\sum_{n=1}^{N} e_n^* - E_{total}\right) + \sum_{n=1}^{N}\nu_n^*\left(e_n^* - \overline{e}\right)\right] = 0 \tag{3.18}$$

Solving equation 3.18 gives

$$e_n^* = -\frac{2b}{h_n}\ln\left[\frac{4N(\lambda^* + \nu_n^*)}{ab\alpha_n}\right] \tag{3.19}$$

If *strong duality* holds and e_n^* is a primal optimal solution and (λ_n, ν_n) any dual optimal point. Then it can be shown that the following condition must hold [28]:

$$\lambda_i f_i(x^*) = 0, i = 1, ..., m \tag{3.20}$$

This condition is known as complementary slackness; it holds for any primal and dual optimal points, when strong duality holds. The *complementary slackness* condition for the dual function in equation 3.18 provides us with two cases;

$$\begin{aligned} either \qquad \nu_n^* = 0 \\ or \qquad e_n^* = \overline{e} \qquad if \quad \nu_n^* > 0 \end{aligned} \tag{3.21}$$

Using eq. 3.22 in eq. 3.20 provides us with the equation for optimal power allocation as given in equation 3.22

$$e_n^* = \frac{\lambda_0^*}{h_n} - \left(\frac{2}{b}\right)\left(\frac{1}{h_n}\right)\ln\left(\frac{1}{h_n}\right) \tag{3.22}$$

with the value of λ_0^* as given in equation 3.24. To obtain the conditions for subcarriers that are not respecting the $\overline{P}$ constraint, we equate eq. 3.22 with eq. 3.22 thus giving

$$h_n\overline{e} + \left(\frac{2}{b}\right) ln\left(\frac{1}{h_n}\right) \leq \lambda_0^* \tag{3.23}$$

After solving the Lagrangian in eq. 3.15 by means of the relevant KKT conditions and the *complementary slackness* conditions, the closed-form solution for optimal power allocation for BER minimization and respecting the peak-energy constraint is represented by:

$$e_n = \begin{cases} 0 & h_n < \exp\left(\frac{-b\lambda_0}{2}\right) \\ \overline{e} & h_n \geq \exp\left(\frac{-b\lambda_0}{2}\right),\ h_n\overline{e} + \left(\frac{2}{b}\right) ln\left(\frac{1}{h_n}\right) \leq \lambda_0 \\ \frac{\lambda_0}{h_n} - \left(\frac{2}{b}\right)\left(\frac{1}{h_n}\right)\ln\left(\frac{1}{h_n}\right) & h_n \geq \exp\left(\frac{-b\lambda_0}{2}\right),\ h_n\overline{e} + \left(\frac{2}{b}\right) ln\left(\frac{1}{h_n}\right) > \lambda_0 \end{cases}$$

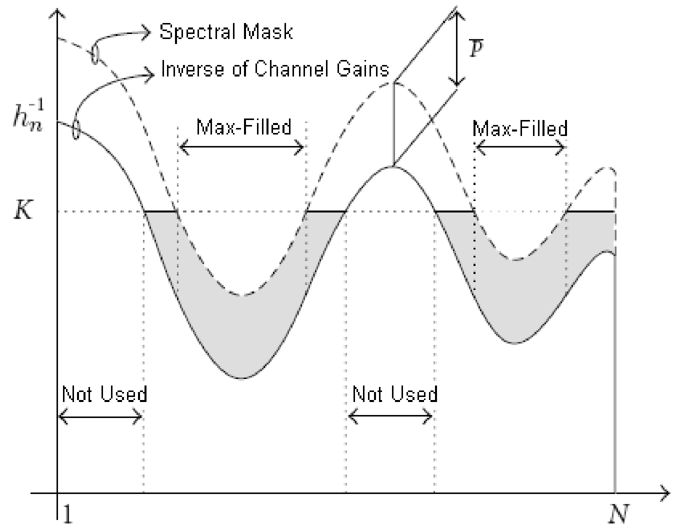
where $\lambda_0 = -(2/b)\ln(4N\lambda/ab)$, and which can be calculated as

$$\lambda_0 = \frac{E_{total} + \left(\frac{2}{b}\right)\sum^{n\in N}\left(\frac{1}{h_n}\right)\ln\left(\frac{1}{h_n}\right)}{\sum^{n\in N}\left(\frac{1}{h_n}\right)} \tag{3.24}$$

3.4.3 Computationally Efficient Algorithm for Peak-Energy Constrained Energy Allocation

As mentioned earlier, many practical systems require a peak-power constraint, where the maximum power/energy transmitted over a frequency/subcarrier is not allowed to exceed

Figure 3.7 Peak Power Constraint for a given Channel Response



a pre-defined limit. This constraint, if uniform over all the subcarriers, can be represented by means of the figure 3.7

The optimal energy allocation, optimized with respect to a particular goal (throughput, BER etc.), is thus often required to be clipped at certain subcarriers in order to respect this constraint. *Iterative Waterfilling* is proposed [20] in the literature as a method to optimally allocate energy to all the subcarriers in the presence of a peak-energy constraint, as explained earlier in this chapter. The basic structure of the iterative waterfilling algorithm is given in form of the following pseudocode

Algorithm 1 PseudoCode of Iterative-Waterfilling (IW) Algorithm

procedure IW(h_n,$\overline{e}$)
    sort h_n in descending order
    set n=1
    while $e_n > \overline{e}$ **do**
        apply WATER-FILLING over subchannels n,...,N
        set $e_n = \overline{e}$
        reduce the total available power by $\overline{e} \Rightarrow E_{total} = E_{total} - \overline{e}$
        update n=n+1
    end while
end procedure

It is important, however, to mention here that the above given IW algorithm structure, as proposed by Baccarelli et al. [20], is for the capacity maximization problem and hence it is assumed that if subcarriers sorted from highest to lowest channel gain, the first subcarrier will be allocated the maximum energy. But, in order for this algorithm to be implemented for our above mentioned BER minimization problem, it has to be a little modified, where in each iteration the subcarrier with the maximum allocated energy has to be searched for, before clipping its power to $\overline{e}$ level.

As can be observed from the pseudo-code, the *iterative* routine of Water-filling is re-run for a number of times in order to arrive at an allocation where no subcarrier is violating the peak-power constraint. This iterative application of the Waterfilling routine, which is termed as the *Iterated Waterfilling* routine, renders high complexity and thus necessitates its simplification, as proposed in the following.

A simplification for the above mentioned *Iterative Waterfilling* procedure in what is termed as the Simplified Iterative Waterfilling (SIW) algorithm is presented below. It is known that the optimal solution uses all the available power budget, that is, the total power constraint (E_{total}) in 3.13 is met with equality.

Remark It is observed that as more power budget is available, the constant λ_0 in 3.24 becomes higher and as a consequence not only allocated energies to the previously allocated subchannels are greater than before but also more subchannels may be turned on.

At the same time a proposition was made in [74] as follows:

Proposition

Given the sorted water-filling energy allocation vector e_N*, if one removes subchannels* $1, ..., L$ *and reduces* E_{total} *by* $\sum_{n=1}^{L} e_n$*, then the new optimal water-filling solution is the (N-L)- point energy vector* $e_{N-L} = e_N - e_1, ..., e_L = [e_{L+1}, ..., e_N]$

The above proposition implies that if we remove any number of subcarriers from the total set of N subcarriers, form a new $E^{'}_{total}$ by removing from the original E_{total} the sum of the allocated energies of the removed subcarriers, and re-apply waterfilling based upon the new set of subcarriers and the new total available energy $E^{'}_{total}$, then the amount of energy allocated to these subcarriers will be same as before.

At the same time we observe that at each iteration of *Iterative Waterfilling* algorithm, the energy of the subcarrier with maximum energy,if exceeding the maximum permissible limit $\overline{e}$, is clipped to $\overline{e}$ and the amount of energy clipped is added to the sum of the energies of the rest of the subcarriers, which is to be re-allocated again in the next iteration to all but the clipped (maximum energy) subcarrier of the current iteration. The implies that the total amount of energy that is to be allocated to $N-1$ subcarriers in the next iteration is more than the total energy allocated to them in the current iteration, which according to the above given remark and proposition infers that the amount of energy allocated to any subcarrier in the next iteration will be more than its allocated energy in the previous iteration. From these arguments, it can be inferred, that a subcarrier exceeding the $\overline{e}$ limit will exceed this limit in any future iteration of the *Iterative Waterfilling* algorithm. Based on this, the iterative waterfilling algorithm can be simplified by clipping ***all*** the subcarriers exceeding the $\overline{e}$ limit to $\overline{e}$ and then removing all of these clipped subcarriers from the allocation process of the next iteration, as shown by means of a pseudo-algorithm below:

Algorithm 2 PseudoCode of Simplified Iterative-Waterfilling (SIW) Algorithm

procedure SIW($\{h_N\}$,$\overline{e}$)
    sort $\{h_N\}$ in descending order
    apply WATER-FILLING over subchannels vector $\{h\}$
    find n_{max} such that $\forall n \in \{h\}$, $e_{n_{max}} \geq e_n$
    while $e_{n_{max}} > \overline{e}$ **do**
        set $N_{max} = 0$ and $\{h_{max}\} = \{\}$
        while $e_{n_{max}} > \overline{e}$ **do**
            update $N_{max} = N_{max} + 1$ and $\{h_{max}\} = \{h_{max}\} \cup \{h_{n_{max}}\}$
            update $\{h\} = \{h\} - \{h_{n_{max}}\}$
            find n_{max} such that $\forall\, n \in \{h\} \Rightarrow e_{n_{max}} \geq e_n$
        end while
        set $e_n = \overline{e}$ for $n \in \{h_{max}\}$
        reduce the total available power $E_{total} = E_{total} - (|\{h_{max}\}|\,.\overline{e})$
        apply WATER-FILLING over subchannels vector $\{h\}$
        find n_{max} such that $\forall n \in \{h\}$, $e_{n_{max}} \geq e_n$
    end while
end procedure

This collective clipping of all the subcarriers crossing the $\overline{e}$ in an iteration of *Iterative Waterfilling* procedure largely reduces the complexity of the original algorithm. The improvement in complexity depends on the number of iterations of the original IW routine, the greater they are, the larger the complexity improvement. We will demonstrate that further substantial complexity reduction of the IW routine is possible by means of our proposed *Iterative Surplus Re-distribution* (ISR) method.

The ISR algorithm is based upon the idea, that once optimal allocation has been performed over all the sub-carriers and all the subcarriers crossing the $\overline{e}$ limit have been crossed, instead of re-allocating all of the new E_{Total}(as done in SIW approach), only the surplus energy (E_{surp} = Sum of the amount of energy greater than $\overline{e}$ of all the clipped subcarriers) needs to be re-allocated or re-distributed over the rest of the sub-carriers in such a way that none of them exceeds the $\overline{e}$ limit.Thus E_{surp} is repeatedly distributed such that with each iteration, its amount decreases till it is completely re-allocated with no violation of $\overline{e}$. This avoid the re-run of the WATERFILL routine in each iteration, thereby further reducing the involved complexity. At each iteration, the goal is to find how much change in the initial λ_0 (λ_0^{ini}), which is equal to ($\epsilon = \lambda_0^{new} - \lambda_0^{ini}$) will be needed to distribute the E_{surp} amongst the suitable subcarriers. This is based upon the assumption that the value of λ_0^{ini} is going to increase with each iteration of Iterative Waterfilling, as established earlier based upon the above given remark and proposal.

An increase ϵ in the λ_0^{ini} value results in a total amount of distributed power given by

$$\begin{aligned}
E_{total}^{new} &= \sum^{n\in\{h\}} e_n(\lambda_0^{new}) \qquad (3.25)\\
&= \sum^{n\in\{h_{old}\}} e_n(\lambda_0^{new}) + \sum^{n\in\{h_{new}\}} e_n(\lambda_0^{new})\\
&= \sum^{n\in\{h_{old}\}} e_n(\lambda_0^{old}+\epsilon) + \sum^{n\in\{h_{new}\}} e_n(\lambda_0^{old}+\epsilon)\\
&= \sum^{n\in\{h_{old}\}} e_n(\lambda_0^{old}) + \sum^{n\in\{h_{old}\}} \frac{\epsilon}{h_n} + \sum^{n\in\{h_{new}\}} e_n(\lambda_0^{old}) + \sum^{n\in\{h_{new}\}} \frac{\epsilon}{h_n}\\
&= \underbrace{\sum^{n\in\{h_{old}\}} e_n(\lambda_0^{old})}_{E_{total}^{old}} + \underbrace{\sum^{n\in\{h_{old}\}} \frac{\epsilon}{h_n} + \sum^{n\in\{h_{new}\}} e_n(\lambda_0^{old}) + \sum^{n\in\{h_{new}\}} \frac{\epsilon}{h_n}}_{E_{re-distributed}}
\end{aligned}$$

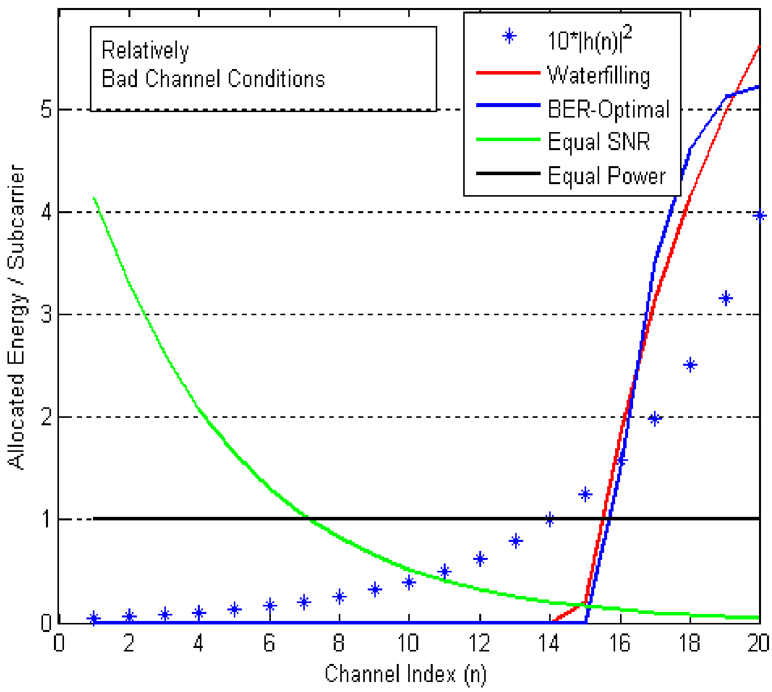
where $\{h_{new}\}$ and $\{h_{old}\}$ represent the set of new subcarriers that were not allocated previously and the set of subcarriers that were allocated energy in the previous iteration and whose energy levels did not exceed $\overline{e}$ in the previous iteration. The new subcarriers gets included in the allocation process because of an increase in the λ_0 value, as more the value of λ_0, more will be the channels with positive power e_k and hence used in the allocation process.

The appropriate value of increase in λ_0 is calculated by searching for the *increase* in its value which is good enough to re-distribute the additional surplus energy. This can be done by tracing the number of subcarriers that are additionally added in the allocation process, as a result of the increase in λ_0. Thus, a new subcarrier from amongst the un-allocated subcarriers is selected and added to the allocation procedure one by one, and the amount of additional energy re-distributed is calculated to see whether it suffices to allocate the additional *surplus energy*. If yes, the procedure is moved on, if not, then an additional previously un-allocated subcarrier is included in the allocation procedure, so as to increase the value of λ_0, which eventually increases the amount of additional energy allocated, which is represented in equation 3.26. Once the total set of subcarriers required to allocate the additional amount of surplus energy is found, the value of ϵ which exactly re-distributes the surplus energy is calculated and based upon that the energy is re-distributed to the concerned subcarriers. Finally, it is checked whether subcarrier still violates the $\overline{e}$ limit; if yes, then the new *Surplus Energy* is calculated and the above procedure is repeated. This is done, till we arrive at an allocation where no subcarrier is violating the $\overline{e}$ constraint. The ISR algorithm is presented in the form of its pseudocode and flowchart below:

Algorithm 3 PseudoCode of Iterative Surplus Re-distribution (ISR) Algorithm

```
procedure ISR({h},ē)
    sort {h} in descending order
    apply WATER-FILLING over subchannels vector {h}
    find {h_max} such that ∀n ∈{h_max}, e_n ≥ ē
    set e_n = ē for n ∈ {h_max}
    Calc. E_surp = ∑^{n∈{h_max}} (e_n − ē)
    while E_surp > 0 do
        while E_{re−distributed} > E_surp do
            Include a new prev. un-alloc. subcarrier {h} = {h} ∪ {h_new}
            Calc. E_{re−distributed} based upon eq. 3.26
        end while
        Calc. λ_new for exact E_surp re-distribution
        Allocate Energy ∀n ∈{h} ⇒ e_n = e(λ_new) as per eq. 3.24
        find {h_max} such that ∀n ∈{h_max}, e_n ≥ ē
        set e_n = ē for n ∈ {h_max}
        Calc. E_surp = ∑^{n∈{h_max}} e_n − ē
        update {h} = {h} − {h_max}
    end while
end procedure
```

Figure 3.8 Energy distribution profile for different allocation strategies for relatively bad channel conditions



As mentioned earlier, the BER-Optimal energy allocation scheme need not resemble the classical Waterfilling distribution, optimized for the throughput. In order to exhibit this difference, energy allocation profiles for different energy distribution methods over different channel scenarios is presented. Channel scenarios consisting of relatively *bad*, relatively *moderate* and relatively *good* channel conditions [25] are taken from in-between the range of [-23 dB to 4 dB], [-9 dB to 10 dB] and [4 dB to 23 dB], respectively as shown in chapter 2.

In 3.8, where relatively bad channel conditions are cosidered, the response of BER-Optimal energy distribution is approximately the same as that of the Waterfilling distribution, as more energy is allocated to the subcarriers with higher SNR, though this response is exactly opposite to that of the Equal-SNR based energy allocation. However, this behaviour is completely changed for good channel conditions, as shown in figure 3.10, where the allocated-energy profile of BER-Optimal scheme becomes approximately same as that of Equal-SNR scheme and opposite to that of Waterfilling, as proven by Chang et al. [25] that the BER-Optimal power allocation scheme becomes the Equal-SNR scheme asymptotically, when the channel conditions are made good.

Figure 3.9 Energy distribution profile for different allocation strategies for relatively moderate channel conditions

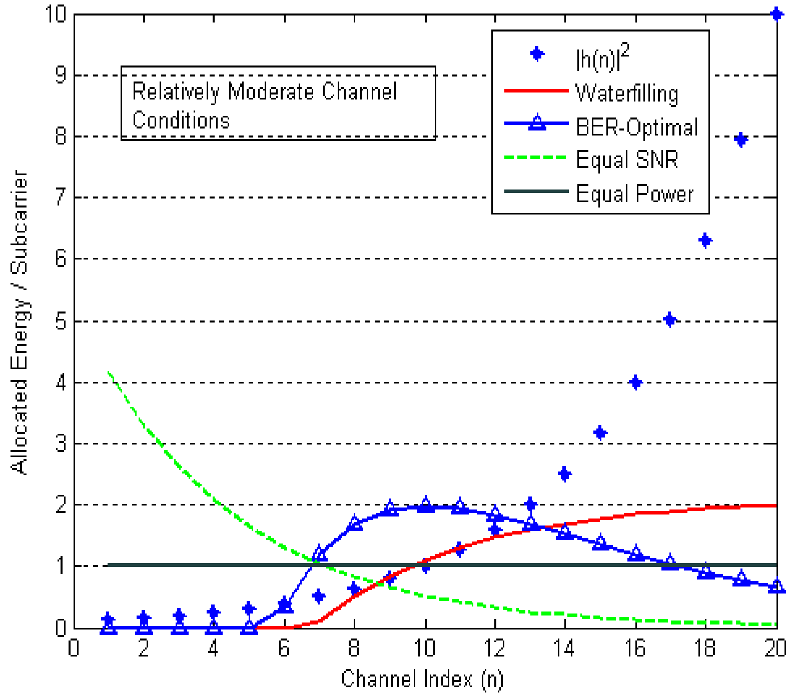


Figure 3.10 Energy distribution profile for different allocation strategies for relatively good channel conditions

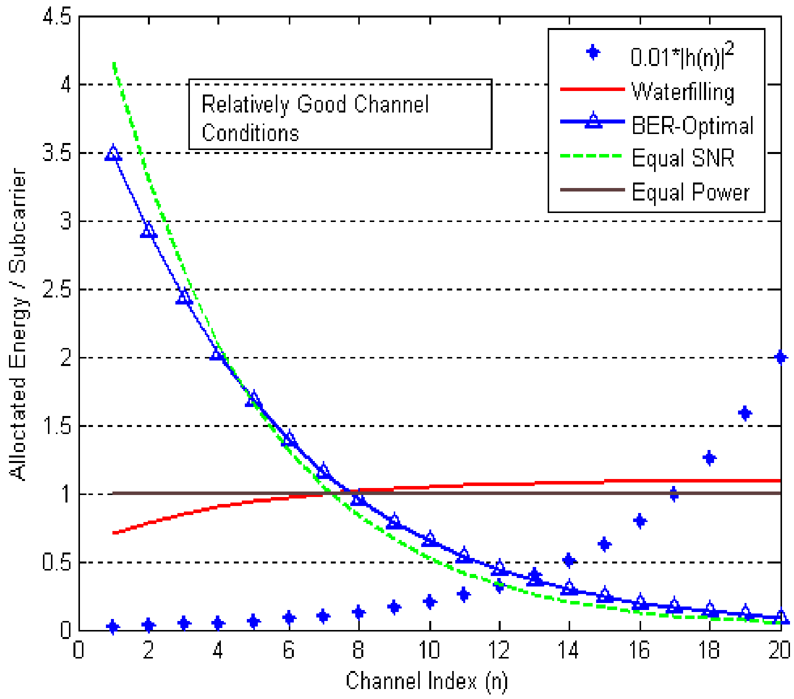
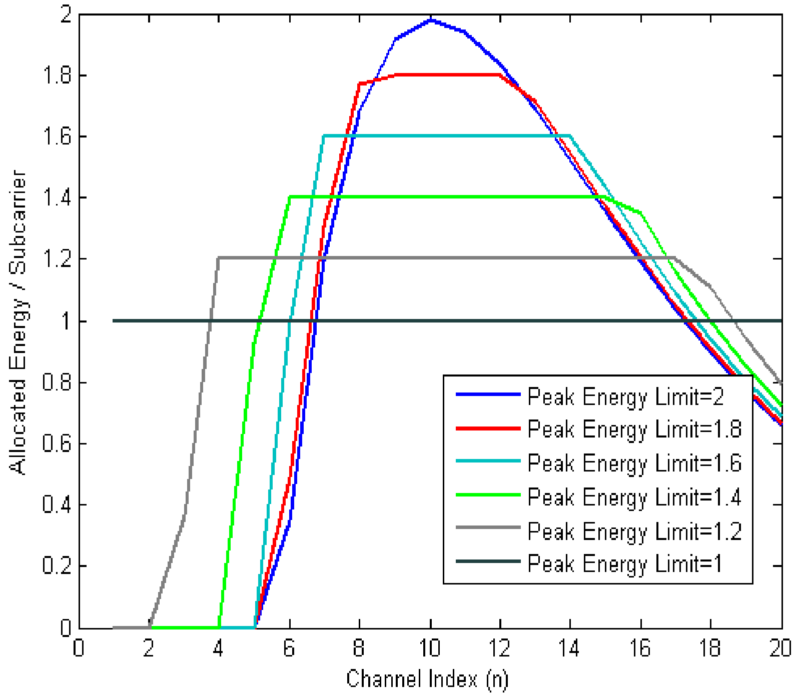


Figure 3.11 Constrained BER-Optimal Energy distribution for different Peak-Energy Constraints.

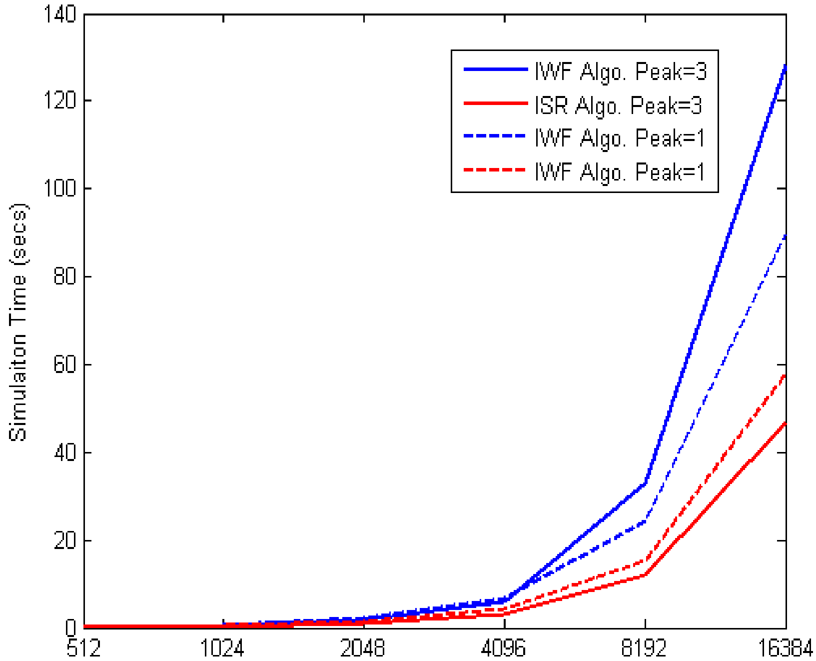


Any optimized energy-allocation, is not bound to respect a given peak-energy constraint. Based upon our earlier developments of an Iterative Surplus Re-distribution (ISR) algorithm, the peak-energy compliant energy distribution, for different peak-energy constraints is shown in figure 3.11. It can be easily observed from the figure that as the peak-energy constraint is lowered and hence more energy is available for re-distribution, those subcarriers that were previously *turned off*, are *turned on* and involved in the allocation procedure, so as to make the total distributed energy equal to the total amount of Energy budget present.

Although, the same peak-compliant optimal energy distribution can also be obtained by means of the classical Iterative Waterfilling (IWF) algorithm, the ISR algorithm helps achieve that with less computational complexity. A comparison in the complexity of the two algorithms, in terms of the simulation time taken, for different parameters and over the same host processor is given in figure 3.12.

As is obvious from the figure 3.12, our algorithm takes less time to converge towards the optimal than the IWF routine. This is mainly because of the fact that it avoids calculation of the complete Waterfilling procedure in each iteration. The simulation

Figure 3.12 Constrained BER-Optimal Energy distribution for different Peak-Energy Constraints.



times of different algorithms, for each peak-energy value, were obtained by averaging over a sufficiently large number of random channel values.

3.5 Conclusion

As established earlier, adaptive resource allocation techniques have emerged as a mean to enhance the system performance in a time and frequency varying channel. In the previous chapter, we tackled the problem of optimizing the allocation of discrete modulation-type (constellation-size) (=bit-loading) over different subcarriers for a given channel response, so as to minimize the total energy consumed for a given throughput and QoS. Just like bit-loading, power/energy allocation techniques, with or without rate-adaptation (=bit-loading), have also been proposed in the literature as means of performance enhancement. Unlike bit-loading, the optimization problem of energy variable does not involve the 'discrete' quantity constraint and hence powerful optimization techniques, like that of Lagrange optimization can be applied for a given set of goals and constraints.

In this respect, we have tackled the problem of optimal energy distribution, over a multicarrier system for a selective channel, such that the aggregate BER is minimized with constraints on total available energy, the peak-energy per subcarrier and the constellation size. The novelty in our contribution lies in extending the theoretical developments by including the peak-energy constraint as well and in the form of proposing an algorithm for peak-energy constrained energy-allocation, that claims to involve significantly less complexity than that of earlier contributions.

While *Iterative Waterfilling* has been proposed in the literature as a method for peak-energy constrained energy allocation, it was observed that it requires the execution of the waterfilling routine in each of its iteration. We propose to avoid the execution of the waterfilling routine in each iteration, in what we name as an Iterative Surplus Re-distribution (ISR) routine, which proposes to allocate only the *Surplus Energy* that is left in an iteration instead of the total energy left, where the Surplus Energy is defined as the sum of the amount of energies that exceed the peak-energy limit. This results in reducing the overall algorithm complexity, because of the avoidance of the Waterfilling routine in each iteration.

The algorithm complexity was compared with the Iterative Waterfilling solution. Our proposed algorithm was found distinctively less complex than the other algorithms, by means of the theoretical complexity analysis of the two algorithms. To verify the theoretical complexity analysis of the three algorithms, we compared the simulation-time of the two algorithms for different parameters and the theoretical complexity analysis was verfied.

Since, the bandwidth scarcity is pushing towards more and more constraints on power-spectral masks, dictating the amount of maximum energy/power allocated at a particular frequency, the application of the ISR algorithm becomes more and more interesting. Although, we assumed the same peak-energy constraint over all the sub-carriers, but it can be easily extended to the case where the peak-energy varies on subcarrier-per-subcarrier basis. In the continuation of the work, the constellation type may be allowed to be varied at the same time, and the best combination of energy and bit allocation can be explored. Also, to quantify the complexity advantage, the two algorithms can be evaluated on a real processor in terms of exact number of execution cycles, which might enable us to quantify the complexity gain of the algorithm with respect to different hard time constraints put forward by different channel scenarios.

Chapter 4

Simplistic Algorithm for Irregular LDPC Codes Optimization Based on Wave Quantification

4.1 Background and Problem Definition

As discussed in the introductory chapter, in the adaptation mode, parameters such as transmission power [6], symbol rate [3], constellation size [5, 30], coding rate/scheme [31] or any combination of them [32, 33] are changed in response to time and/or frequency-varying channel conditions. Since the previous two chapters dealt with simplifying the algorithms for adaptive modulation and adaptive power allocation respectively, in this chapter we will explore the adaptation possibilities from a channel coding perspective, where the amount of redundancy can be varied with respect to the channel condition.

Claude E. Shannon [66] was the first to characterize the optimal performance theoretically reachable for coded transmission over a noisy channel. A basic scheme for communicating over noisy channels is by adding redundancy (in other words channel coding) to the original message to be transmitted. The message to be sent is encoded with a channel code before it is transmitted on the channel. At the receiving end, the output from the channel is decoded back to a message, hopefully the same as the original one. A fundamental property of such systems is Shannons channel coding theorem, which states that reliable communication can be achieved as long as the information rate does not exceed the *capacity* of the channel, provided that the encoder and decoder are allowed to operate on long enough sequences of data (extensive treatments can be found in many textbooks, e.g. [29]).

In traditional channel coding schemes [29], fixed codes are used which are not optimized with respect to the nature of the channel. To keep the performance at a desirable level, they are designed for the average or worst case situation which results in an inefficient usage of system resources. To more efficiently employ the channel coding, techniques have been proposed in the past to optimize the code structure with respect to the underlying channel. This may involve varying the code-rate in a time-selective channel [31,88] or optimizing the structure of the code-word with respect to the frequency selective nature of the underlying channel [89].

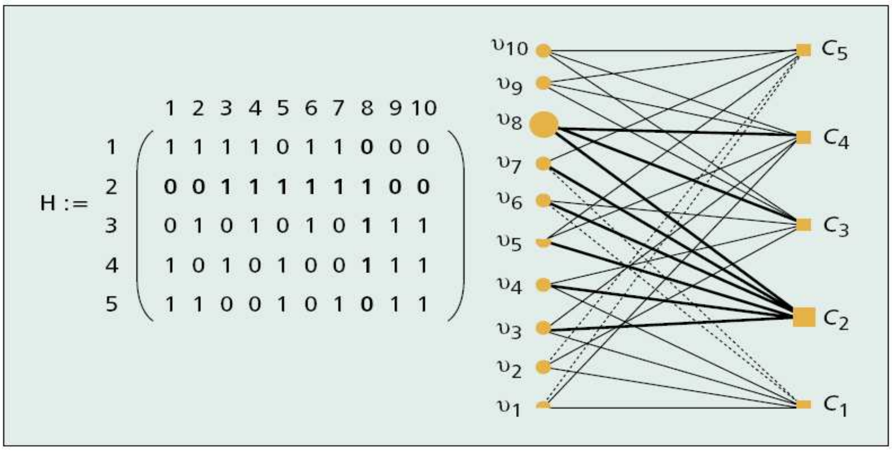
Since the introduction of Shannon's Coding theorem, the construction of capacity approaching codes has been the main challenge of coding research. A huge library of error correcting codes have been designed, however, until the arrival of turbo codes in 1993 [35], practical coding schemes for most channels fell short of the Shannon limit. Turbo codes marked the beginning of near Shannon limit performance for the AWGN channel, largely improving on the previous schemes. Immediately afterward, MacKay and Neal [36] rediscovered Gallager's [47] long-neglected Low-Density Parity-Check codes (LDPC) codes as they are explained in the next section.

4.1.1 LDPC History, Representation and Types

Low-density parity-check (LDPC) codes were invented by R. G. Gallager [34] in 1962. He discovered an iterative decoding algorithm which he applied to a new class of codes. He named these codes low-density parity-check (LDPC) codes since the parity-check matrices had to be sparse to perform well. Yet, LDPC codes have been ignored for a long time due mainly to the requirement of high complexity computation, if very long codes are considered. In 1993, C. Berrou et. al. invented the turbo codes [35] and their associated iterative decoding algorithm. The remarkable performance observed with the turbo codes raised many questions and much interest toward iterative techniques. In 1995, D. J. C. MacKay and R. M. Neal [36] rediscovered the LDPC codes, and set up a link between their iterative algorithm to the Pearl's belief algorithm [37], from the artificial intelligence community. At the same time, M. Sipser and D. A. Spielman [48] used the first decoding algorithm of R. G. Gallager (algorithm A) to decode expander codes.

LDPC codes are a type of linear block codes. Linear codes are defined in terms

Figure 4.1 Parity Check Matrix of a (3,6) Regular LDPC Code and the corresponding Bi-Partite Graph



of generator and parity-check matrices. Generator matrix **G** maps information **s** to transmitted blocks **t** called codewords by **t** = **s.G**. For a generator matrix **G**, there is a parity-check matrix **H** which is related as **G.**H^T = **0**. All codewords must satisfy **s.**H^T = **0** in terms of the parity-check matrix **H**. LDPC codes can be represented in a simple bipartite graph representation [90], which consists of two types of nodes: variable nodes and check nodes. Each variable(check) node corresponds to the column (row) of the parity-check matrix H. The edges in the graph indicate the variable nodes participating in the corresponding check node. Thus, a one located at position (i, j) of H corresponds to an edge between variable node i and the check node j. As an example of a Tanner graph [91], a regular LDPC code of length n = 10 and k = 5 is shown in the following figure. The equivalent bipartite graph representing this code is also shown in the same figure. In this code, every variable node has degree three and each check node has degree six. Thus, this code is called a (3,6) regular LDPC code.

If the parity-check matrix H has the same weight per row and the same weight per column, the resulting LDPC codes is called regular. We use a two tuple (d_v, d_c) to represent a regular LDPC code whose column weight is d_v and row weight is d_c. When the weight in every column is not the same in the parity-check matrix, the code is known as an irregular LDPC code. Irregular LDPC codes have a better asymptotic performance and can practically reach channel capacity as shown in [38]. We will discuss the issues related to the construction of finite-length irregular codes later which is the main topic of this chapter but first we give a brief introduction to LDPC codes construction/encoding and decoding phenomena.

4.1.2 LDPC Codes Construction and Encoding

Usage of LDPC codes to transmit information bits in a system requires two distinct steps:

1. **LDPC Code Design or Code Construction**

2. **Encoding methodology**

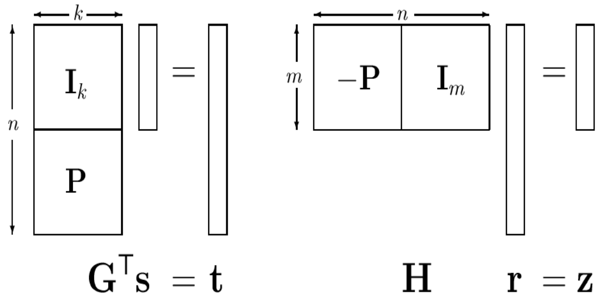
The design or construction process refers to the process for the construction of a particular LDPC parity check matrix H. The structure refers to the description of the Tanner graph: the degrees of the various nodes, restrictions on their interconnections, whether variable nodes are punctured etc. and as how these restrictions (degree distribution, rate etc.) can meet the practical constraints (finite dimension of the code-block, girths [1]). Conventionally this design process is again divided into two steps of 1. Optimization of Degree Distribution and 2. Placement of Edges between different variable-nodes for a given Degree Distribution. Generally these steps are performed seperately independent of each other where the first step of Degree Distribution Optimization is classically performed for asymptotic lengths of block-codes while Combinatorial techniques have been employed for the latter step of Edge Placement to define the parity check matrix of finite length codes. Various algorithms/methodologies have been proposed for both of the steps which will be referred to in the next section of state-of-the-art.

The encoding of LDPC codes like that of other Linear Block Codes takes place by means of a Generator Matrix $\mathbf{G}$, which is related to the Parity-Check Matrix $\mathbf{H}$ as $\mathbf{G}.H^T = \mathbf{0}$ and which is used for the generation of the code-blocks to be transmitted $\mathbf{t}$ from the message bits $\mathbf{s}$ by $\mathbf{t} = \mathbf{s}.\mathbf{G}$. It is common to consider $\mathbf{G}$ in systematic form with $G \equiv [I_k|P]$ so that the first k transmitted symbols are the source symbols. The notation $[A|B]$ indicates the concatenation of matrix $\mathbf{A}$ with matrix $\mathbf{B}$; I_k represents the $k \times k$ identity matrix while the remaining m=n-k symbols are *parity-checks.* The corresponding parity-check matrix will have the form $[-P|I_m]$. Each row of the parity-check matrix describes a linear constraint satisfied by all codewords i.e. $\mathbf{G}.H^T = \mathbf{0}$ and hence the parity-check matrix can be used to detect errors in the received vector $\mathbf{r}$=$\mathbf{t}$+$\mathbf{n}$ where $\mathbf{n}$ represents the noise added by the channel. Therefore

$$\mathbf{Hr} = H(t+n) = HG^T s + Hn = Hn := z \tag{4.1}$$

[1]In a bi-partite graph, a closed path with edges starting from and ending at the very same bit-node is called a 'cycle' of 'e' edges. 'Girth' refers to the length of the shortest cycle in a graph.

Figure 4.2 Systematic Linear Block Codes Encoding and Decoding Process



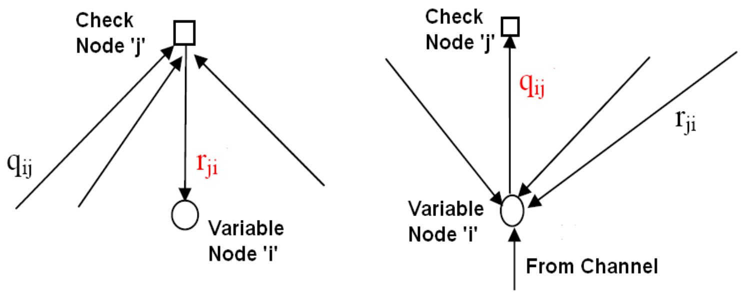
where **z** is the syndrome vector. If the syndrome vector is null, we assume there have been no errors. Otherwise, the decoding problem is to find the most likely noise vector n^i that explains the observed syndrome given the assumed properties of the channel. For conventional linear-block codes this decoding is done by means of a given well-known estimator like that of Maximum-Likelihood (ML), however the decoding of LDPC block-codes is conventionally one by means of an iterative-decoding algorithms whose performance is a close approximation of the ML decoding [92] and which will be explained in the next section. Pictorically the above mentioned operation is given by figure 4.2

Thus, given a Tanner graph, the problem of designing an encoder generally boils down to selecting a Generator Matrix **G** that can efficiently encode the message bits. The encoding algorithm is generally less complex than the decoding algorithm, however, LDPC codes have a weak point at their encoding process because the sparse parity-check matrix does not have necessarily a sparse generator matrix. Encoding process using a dense generator matrix G yields to an N^2 computational complexity that is quadratic with respect to the block length. Thus, this method is not suited for encoding LDPC codes with long block lengths. Another encoding scheme which makes use of a lower triangular shape Parity-Check matrix [36] instead of a Generator matrix is very commonly employed because of reduced complexity as instead of computing the product **t=s·G**, the equation $H\Delta s^T = 0$ is solved. Similarly other methods such as those of iterative encoding [93], cyclic parity-check matrices [94] and Sparse Generator matrices [95] have been proposed in the literature for reducing LDPC encoding complexity.

4.1.3 LDPC Codes Decoding

Linear block codes conventionally employ syndrome-based decoding methods as explained above, that make use of techniques like MAP (Maximum a Posteriori) for finding the

Figure 4.3 Message Passing Phenomenon between Variable and Check Nodes



best code-word for a given received vector of bits **r**. LDPC on the other hand employ a class of iterative decoding techniques known as the 'Message-Passing' algorithms. The reason for their name is that at each round of the algorithms messages are passed from variable nodes to check nodes, and from check nodes back to variable nodes.

The messages from variable nodes to check nodes are computed based on the observed value of the variable node and all but one of the messages passed from the neighboring check nodes to that variable node. The one neighboring message not used for computing the message to be sent from the variable node v to a check node c, must not take into account the message sent in the previous round from the very same c to v. The same is true for messages passed from check nodes to message nodes. This phenomenon can be pictorically respresened as shown in figure

4.1.3.1 Belief Propagation

One important subclass of message passing algorithms is the belief propagation algorithm. This algorithm is present in Gallager's work [47], and it is also used in the Artificial Intelligence community [90]. The messages passed along the edges in this algorithm are probabilities, or beliefs. More precisely, the message passed from a message node v to a check node c is the probability that v has a certain value given the observed value of that message node v and all the values communicated to v in the prior round from check nodes incident to v other than c. On the other hand, the message passed from c to v is the probability that v has a certain value given all the messages passed to c in the previous round from message nodes other than v.

It is sometimes advantageous (computationally) to work with likelihoods, or sometimes even log-likelihoods instead of probabilities. For a binary random variable x let L(x) = Pr[x=0]/Pr[x=1] be the likelihood of x. Given another random variable y, the conditional likelihood of x denoted L($x|y$) is defined as Pr[$x = 0|y$]/Pr[$x = 1|y$]. Similarly, the log-likelihood of x is ln(L(x)), and the conditional log-likelihood of x given y is ln((L($x|y$)). A commonly implemented form of message-passing algorithm is known as the Sum-Product Algorithm because the computation of messages at nodes which are to be exchanged between check and variable nodes, is in the form of a Summation of many Product terms. Generally a decoding algorithm consists of the following four steps:

1. **Initialization** : For all variable nodes i, initializing q_{ij} (messages to be sent to the check-nodes) based upon the channel response at each variable node.

2. **Check to Variable Node Message Passing** : Updating r_{ji} messages at all the check nodes to be transmitted to the corresponding variable nodes in the form of ln(L(r_{ji})). ln(L(r_{ji})=log($r_{ji}(0)/r_{ji}(1)$) where $r_{ji}(0)$ ($r_{ji}(1)$) is the probability of the check-node j being satisfied such that the variable node i is 0 (1).

3. **Variable to Check Node Message Passing** : Updating q_{ij} messages at all the check nodes to be transmitted to the corresponding variable nodes in the form of ln(L(q_{ij})=log($q_{ij}(0)/q_{ij}(1)$) where $q_{ij}(0)$ ($q_{ij}(1)$) is the probability of the variable-node i having the value of 0 (1) based upon the information of all the associated check-nodes except check-node j.

4. **Decision** : Update L(Q_i) where Q_i is the likelihood of the variable-node i based upon the incoming information from channel and that from all the associated check-nodes.

At the end of each iteration it is checked whether the resultant codeword is satisfying $\mathbf{H}{\cdot}v^T$=0 where **v** indicates the values of variable-nodes after each iteration. The decision is given by $\mathbf{v} = v_i$ such that $v_i = 1$ if L(Qi)<0; otherwise, $v_i = 0$. If **v** is a valid codeword satisfying $\mathbf{H}{\cdot}v^T$=0, the algorithm halts; otherwise, the routines from step 2 to step 4 are repeated until some maximal number of iterations is reached without a valid decoding.

Another important note about belief propagation is that the algorithm itself is entirely independent of the channel used, though the messages passed during the algorithm are completely dependent on the channel. With regards to the relationship of belief propagation and maximum likelihood decoding, the answer is that belief propagation is in general less powerful than maximum likelihood decoding and hence converges iteratively to a sub-optimal solution that may not be the maximum likelihood solution.

There is a second distinct class of decoding algorithms that is often of interest for very-high speed applications, such as optical networking. This class of algorithms is known as hard-decoding algorithms; as the messages exchanged between the nodes in each iteration consist of the hard values of 0 and 1. They generally have even lower complexity than belief-propagation algorithms, albeit at the cost of somewhat worse performance. A popular class of such decoders known as the Majority-Based Hard Decoders will be discussed in detail later in this chapter.

4.1.4 Irregular LDPC Codes

4.1.4.1 Irregularity

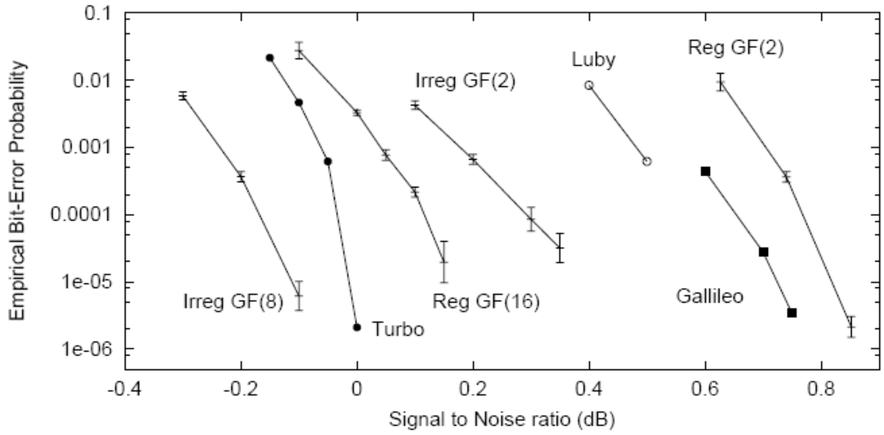
In the Gallagers original LDPC code design, there is a fixed number of ones in both the rows (k) and the columns (j) of the parity check matrix: it means that each bit is implied in j parity check constraints and that each parity check constraint is the exclusive-OR (XOR) of k bits. This class of codes is referred to as regular LDPC codes. On the contrary, irregular LDPC codes do not have a constant number of non-zero entries (representing edges in the bi-partite graph) in the rows or in the columns of H. They are specified by the distribution degree of the bit $\lambda(x)$ and of the parity check constraints $\rho(x)$, using the notations of (Luby et al. 1997) as specified below

$$\lambda(x) = \sum_{i=2}^{dv} \lambda_i x^{i-1} \qquad \rho(x) = \sum_{i=2}^{dc} \rho_i x^{i-1} \tag{4.2}$$

where λ_i (resp. ρ_i) denotes the proportion of non-zero entries of H which belongs to the columns (resp. rows) of H of weight i. If *degree* is the number of edges connected to a node then λ_i is the fraction of the edges which are connected to the degree i data nodes and ρ_i is the fraction of the edges which are connected to the degree k check nodes with dv and dc representing the maximum degree for data and check nodes respectively.

If the irregularity distribution of an Irregular LDPC Code is properly chosen, it was observed that Irregular codes can show superior performance to their regular couterparts. For large block lengths, they have shown to achieve the aysmptotic performance closest to capacity than other known codes (at times even better than the best know turbo-codes) as depicted in the figure 4.4 [90]

Figure 4.4 Improved performance of Irregular Codes

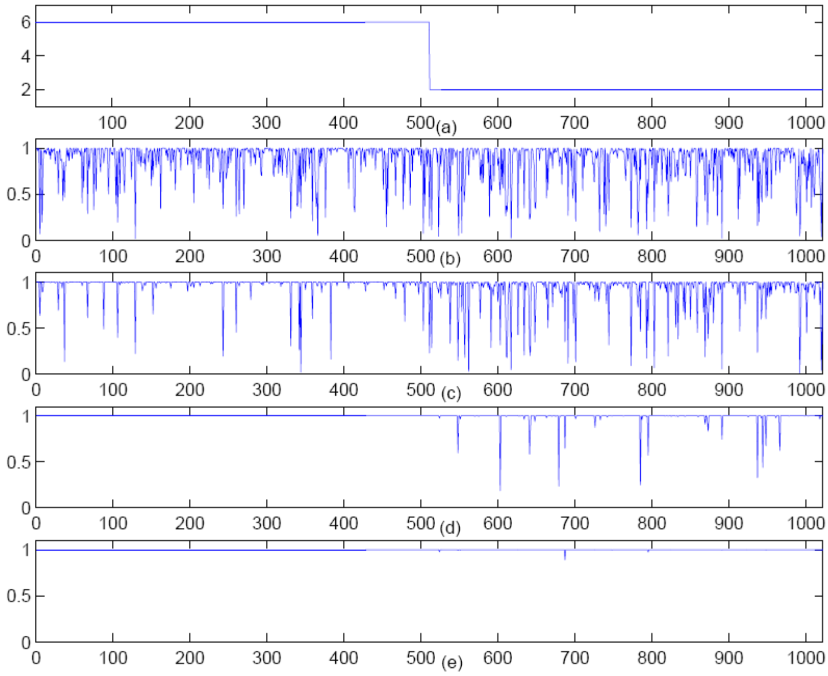


4.1.4.2 Wave-Effect

In his seminal paper on Irregular LDPC [38], Luby gave the intuitive analysis of the better performance of Irregular LDPC codes, of what he termed as the ***Wave Effect***. He argued that from the point of view of a message node, it is best to have high degree, since the more information it gets from its check nodes the more accurately it can judge what its correct value should be. In contrast, from the point of view of a check node, it is best to have low degree, since the lower the degree of a check node, the more valuable the information it can transmit back to its neighbors. He inferred that there is a reason to believe that a wide spread of degrees, at least for message nodes, could be useful. Message nodes with high degree tend to correct their value quickly. These nodes then provide good information to the check nodes, which subsequently provide better information to lower degree message nodes. Irregular graph constructions thus have the potential to lead to a wave effect, where high degree message nodes tend to get corrected first, and then message nodes with slightly smaller degree, and so on down the line. This wave effect is represented in figure 4.5 [96], where the A Posterior Probability variable nodes with higher degree reaches the maximum value 1 in much less number of iterations than the nodes with lesser degree.

It has been shown that irregular LDPC codes are especially interesting because one can optimize the parameters that characterize their irregularity in order to find the codes that are the closest to the capacity for various types of channels. The flexible structure of the Irregular LDPC codes, which can be represented by means of a Degree distribution can be used to curtail Unequal Error Protection (UEP) as per channel's requirements

Figure 4.5 Wave Effect of Irregular Codes. (a) : Degree distribution for different Variable Nodes. A Posteriori Probability after (b) 1 iteration. (c) 5 iterations (d) 10 iterations (e) 15 iterations



e.g. Frequency Selective channel where the protection requirements of different variable nodes is different based on the different corresponding channel gains.

4.1.4.3 Design of Irregular LDPC Codes

As stated earlier design process of Irregular LDPC codes is divided into two steps of 1. Optimization of Degree Distribution and 2. Placement of Edges between different variable-nodes for a given Degree Distribution. Usually, either the channel threshold or the code rate are optimized so as to find the best degree distributions. The threshold of the channel is the value of the channel parameter above which the probability tends towards zero if the iterations are infinite (and the code length also). Optimization seeks the irregularity profile which minimizes the threshold or maximizes the rate. Generally the irregularity is introduced only in the degrees of variable nodes with constant check-node degrees. Some common methods for the irregularity profile optimization like that of Density Evolution will be discussed in the next section along with the pros and the cons attached with such methods and some open problems that we have tried to address in this chapter.

4.2 State of the Art on Irregular LDPC Codes Optimization and Construction

In this section important works related to both the ***optimization*** methodologies for irregularity profile optimization as well as the techniques for ***construction*** of good finite-length codes for a given degree distribution will be given, as they both combine to define the method of developing irregular LDPC codes. Also, some recent developments related to hard-decoding will be cited because one of our major contributions in this chapter relates to the complexity reduction of the hard-decoding error-probability analysis which was later used for the 'Wave Quantization' methodology, as they will be discussed later in this chapter.

4.2.1 Works Related to Irregularity Profile Optimization

The Irregularity profile of Irregular LDPC codes has conventionally been optimized based upon the asymptotic assumption i.e. considering an infinite length of the code-word and infinite decoding iterations. This results in giving a degree-profile to which corresponds a ***family*** of codes. Generally performance bounds are often derived with respect to the parameter set pertaining to a ***family*** of codes based upon the assumption that no ***cycles*** exist in a particular code-word, which might not be true for finite length code-words as will be discussed in the next section.

Two algorithms have generally been employed to design a class of irregular LDPC codes under some channel constraints: 1. The ***Density Evolution*** algorithm [39] and the ***Extrinsic Information Transfer (EXIT)*** charts [40].

Richardson et al. [39] designed capacity approaching irregular codes with the Density Evolution (DE) algorithm. This algorithm tracks the probability density function (pdf) of the messages through the graph nodes, under the assumption that the cycle free hypothesis is verified. It is a kind of belief propagation algorithm with pdf messages instead of log likelihood ratios messages. Usually the *Irregularity Profile* is optimized such that either the channel noise threshold [2] or the code-rate is maximized. Optimization based on DE algorithm are often processed by means of differential evolution algorithm

[2]It is defined as the noise limit below which all the codes from the code ensemble reach a very small probability of error if the code-block length is taken to infinity.

when optimizations are non-linear, as for example in [97] where the authors optimize an irregular LDPC code for uncorrelated flat Rayleigh fading channels.

The technique of Differential Evolution [41] is essentially an efficient modified version of Genetic Algorithm. The main properties of Differential Evolution are 1) 'Initialization' in which a random first generation of vectors is created with changes over time, 2) 'Mutation and Recombination' which defines how to modify the population in each generation, 3) 'Selection' of the survivors and finally 5) the 'Stopping Criterion'. Its name comes from the differential nature of the mutation step, in which at each round random pair-wise differences of two pairs of population vectors are added to population members – the details of it can be consulted from [41]. Richardson [39] was the first to use this optimization approach for the problem of irregularity profile optimization employing the channel threshold criterion. DE has also been used for optimizing irregular LDPC profile in BEC channels [98] with respect to channel threshold and in frequency selective multicarrier channels [96] with respect to BER.

Another technique named Gaussian approximation [42] can also be used instead which renders simplified implementation to the DE algorithm. The probability density function of the messages in this case are assumed to be Gaussian and the only parameters that has to be tracked in the nodes is the mean of the message pdf.

Optimization for LDPC codes have been made for various types of channels e.g. Partial Response channels [99] , ISI Channel (Code Length 10^6, Linear Programming, DE) [100], Frequency Selective Channel [96], Multiple Access Channel (2 users, Gaussian Approximation) [101]; a joint AWGN and Rayleigh fading channel approach [97], Fading Broadcast Channels [102] minimum shift keying modulations [103]; a Multiple Input Multiple Output (MIMO) channel with an OFDM modulation (DE with Gaussian Approx. for both Short and Large Block-Length) [104].

4.2.2 Works Related to Finite Length Codes Construction

A number of methods exist for the **construction**(as defined above) of individual codewords for a given degree distribution profile corresponding to a **family/ensemble** of codes. Initial LDPC Codes construction methods consisted of *Pseudorandom* [34] or *Random* [43] construction techniques. Random techniques create the H matrix by randomly selecting the positions of '1s' to be allocated to the parity-check matrix H, constrained by the degree (maximum number of 1s to be allocated per column/row of H matrix) distribution profile. A number of slightly varying random-construction techniques are given in [43] along with their performance comparison. Normally such ‘Completely Random’ constructions render bad sparsity and girth characteristics to the resultant TG, however in [34], the parity check matrix is constructed by the concatenation and/or superposition of sub-matrices; these sub-matrices are created by processing some permutations on a particularly (random or not) sub-matrix which usually has a column weight of 1. This helps in rendering good sparse properties to the resultant H matrix.

In spite of their algorithm simplicity, there are certain disadvantages involved when randomly constructed LDPC codes are to be used in actual communication scenarios. Random constructions need to be stored explicitly in memory in order to be used for encoding or decoding. Long block length means very large memory usage just to store the m*N Parity Check matrix. This also affects the computational efficiency of the code which, in real life, might be even more crucial than the BER performance. To counter this, usage of some form of structure to achieve a deterministic construction algorithm. The main advantages in using structure can be summarized as an increase in flexibility/adaptability, and a reduction in cost; in terms of complexity, memory usage, and transmission latency. The latter is due to the possibility of specifically adapting decoders to the structural pattern of the code.One approach to the deterministic construction of LDPC codes is based on *Array Codes* which belong to the Algebraic Codes Family. Array codes are two dimensional codes that have been proposed for detecting and correcting burst errors [105]. When array codes are viewed as binary codes, their parity-check matrices exhibit sparseness which can be exploited for decoding them as LDPC codes using the SPA or low-complexity derivatives thereof. Improving on the algebraic construction, two branches in combinatorial mathematics have also been exploited for structured LDPC Code designs: Finite Geometry [106] and Balanced Incomplete Block Designs [107].

It is imperative to search for good LDPC codes with improved girth characteristics due to the bad influence of short-length cycles on the performance . Many algorithms have been proposed in this regard which improve the girth characteristics of the underlying TG for a given degree distribution profile. A ‘Bit Filling’ algorithm was proposed in [44] which proposes to maximize the code-rate (k/n) given an irregularity profile and using the criterion of code-matrix homogeneity for the allocation of edges. Basic idea is to incremently add the bit-nodes (corresponding to the columns of parity-check matrix) , each time picking the check-nodes that result in least perturbation of the code-matrix homogeneity and also respecting the pre-defined girth constraint. This method was later employed to design codes with 'largest possible' girth [108]. Similarly a 'Progressive Edge Growth' (PEG) algorithm [45] has also been recently proposed to find the optimal irregular LDPC code for a given CSI and 'irregularity' profile. It performs an edge-by-edge allocation each time selecting the case which has least effect on the girth thereby optimizing irregularity profile for girth maximization. Both for regular and Irregular finite length codes, PEG algorithm significantly improves the performance with respect to random constructions with no girth conditioning. Furthermore, in [109], the authors study the histogram of cycles length of randomly generated LDPC codes, based on MacKays construction 2A and they select the best cycle-length histogram shape and show that increasing the mean girth of the TG results in giving better performance than maximizing the local girth. Similarly it was shown in [46], that not all short-length cycles have equally deteriorative impact on the performance and hence the selective avoidance of cycles was proposed.

4.2.3 Works Related to Hard Decoding for Irregular LDPC

The decoding of LDPC codes, regular or irregular, is generally done with the classical Message-Passing or Belief-Propagation (BP) algorithm where information related to a bit's probability/likelihood is exchanged. However, a mathematical analysis of probabilistic decoding for a number of iterations is difficult. Gallager [47] developed a weak-bound based on Hard-Decoding/ Bit-Flipping [48] Decoding Algorithms (Later known as the Gallager A and Gallager B decoding algorithms), which was later extended for the irregular [38] case and which will be employed later in this chapter for our analysis. Bit flipping usually operates on hard decisions: the information exchanged between neighboring nodes in each iteration is a single bit. The basic idea of flipping is that each bit, corresponding to a variable node assumes a value, either 0 or 1, and, at certain times, decides whether to flip itself (i.e., change its value from a 1 to a 0 or vice versa).

Exact Thresholds and Optimal Codes for the Binary-Symmetric Channel (BSC) were calculated analytically using Gallagers Decoding Algorithm A in [110]. A special form of the bit-flipping algorithms known as the Majority-Based (MB) decoding [Zarrinkhat04] algorithms, which decide the message bit to be sent in each iteration based upon the values of majority of the incoming messages, has also been explored in [49]. The convergence properties of MB-based algorithm showed that they converge to zero super-exponentially to the number of iterations, as compared to the exponentional convergence of Gallager-A algorithm to zero. Also Hybrid Decoding based on the MB-based algorithms has been recently explored in [111], which was shown to perform better than classical non-hybrid decoding schemes.

Renewed research interest in Hard-Decoding has resulted in some improved proposals such as that of Weighted Bit-Flipping algorithms [112] which increase in performance without significant increase in the algorithm complexity.

4.2.4 Open Problem in Finite Length Irregular LDPC Code Design

Although LDPC codes exhibit impressive capacity approaching performance for very long code lengths, such long code lengths are not appropriate for many practical bandwidth efficient communication applications. Analysis and construction of short-to moderate-length Irregular LDPC codes is of particular interest in delay-sensitive wireless communications. The asymptotic performance of LDPC codes (as the code length goes to infinity) can be obtained from density evolution [39]. Less is known about the performance of finite-length LDPC codes.

Another important finding with regards to the optimization of finite-length irregular LDPC codes, is that the optimal degree distribution which is found by rigorous analytical analysis for asymptotic case, gives poor performance than its regular counterpart when applied to finite-length LDPC codes. This is primarily because of the fact that aymptotic analysis does not take into account the presence of cycles in the graph, whose presence strongly affects in finite length codes. It was confirmed by the simulation results in [113] that irregular LDPC codes with degree distribution optimized for infinite-length codes, is not optimal for finite-length LDPC codes.

What amount of irregularity (i.e. degree distribution) should be introduced in the Finite-Length Irregular Codes? How the interconnections between nodes with different degrees should be made, i.e. should the high-degree *elite* nodes be connected to other *elite* nodes to generate the best possible wave based on Luby's intuitional analysis or nodes with higher degree be connected to those with a low degree (less no. of attached parity-checks) ? How the different connections be made in perspective of the local and mean girth of the underlying graph? These are some open questions in the field of Finite

Length Irregular LDPC Codes Design and the answer to some of which we will try to find in this chapter. Our approach would be to somehow quantify the well-known 'wave-effect' and use this as a parameter for determining the irregularity structure of the LDPC code of a given length and code-rate.

4.3 Majority-Based (MB) Hard-Decoding Algorithm for Irregular LDPC Codes

For practical applications on channels other than the BEC the belief propagation algorithm is rather complicated, and often leads to a decrease in the speed of the decoder. Therefore, often a discretized version of the belief propagation algorithm is used. The lowest level of discretization is achieved when the messages passed are binary. In this case one often speaks of a hard decision decoder or the bit-flipping decoded, as opposed to a soft decision decoder as introduced in the last section. Gallager himself described two hard decision decoding algorithms on the BSC alongwith the soft-decoder, when he initially proposed LDPC codes. Rather than the decoder complexity, he was interested in the analytical analysis of the decoder and conjectured that though the mathematical analysis of probabilisic decoding is difficult, a very weak bound on the probability of error (p_e) of probabilistic decoding can be found by the much easier analytical analysis of these hard-decoders. Below is the basic structure of the commonly cited Gallager A and Gallager B hard-decoders alongwith their p_e analysis.

4.3.1 Gallager A and Gallager B Hard-Decoding Algorithms

Gallager A Algorithm :

- Messages are from the set $0, 1$ and represent the current estimate of the decoder of a particular bit.
- **At Check Nodes** : The outgoing message from a check-node results from the computation of the XOR sum of the incoming (extrinsic) messages
- **At Variable Nodes** : The outgoing message equals the originally received message, except if ***all other*** incoming messages agree, in which case this common value is

sent

To derive the p_e analysis for the above algorithm, we assume a Binary Symmetric Channel (BSC) with crossover probability p_e^0 and a regular (n,j,k) code with n as the code-length, j as the variable-degree and k as the check-degree. If a variable node is received in error (an event of probability p_0) and p_i is the probability of error of a variable node after iteration i, then it was showed [34] that p_{i+1} is given by

$$p_{i+1} = p_0 - p^{err-corr}(p_i) + p^{err-add}(p_i) \tag{4.3}$$

where $p^{err-corr}(p_i)$ is the probability of correcting an error and $p_{err-add}(p_i)$ is the probability of adding an error during iteration $i+1$. In the case of Gallager A algorithm $p^{err-corr}(p_i)$ is given by

$$p_0 \cdot \left[\frac{1+(1-2.p_i)^{k-1}}{2}\right]^{j-1} \tag{4.4}$$

and $p^{err-add}(p_i)$ is given by

$$(1-p_0) \cdot \left[\frac{1-(1-2.p_i)^{k-1}}{2}\right]^{j-1} \tag{4.5}$$

Gallager proposed that stronger results for decoding will be achieved if for some integer b, a variable-node is changed whenever b or more of the parity-check constraints rising from the variable-node are violated, unlike the case of the Gallager A algorithm where all the parity-check constraints of the incoming messages need to be violated, for a variable node to change its value . This modified decoding is commonly termed as Gallager B decoding algorithm. The value of $p^{err-corr}(p_i)$ in equation 4.4 this case is given by

$$p_0 \cdot \sum_{l=b}^{j-1} C_l^{j-1} \cdot \left[\frac{1+(1-2.p_i)^{k-1}}{2}\right]^{l} \cdot \left[\frac{1-(1-2.p_i)^{k-1}}{2}\right]^{j-1-l} \tag{4.6}$$

and $p^{err-add}(p_i)$ as:

$$(1-p_0) \cdot \sum_{l=b}^{j-1} C_l^{j-1} \cdot \left[\frac{1-(1-2.p_i)^{k-1}}{2}\right]^{l} \cdot \left[\frac{1+(1-2.p_i)^{k-1}}{2}\right]^{j-1-l} \tag{4.7}$$

where C_l^{j-1} indicates the binomial coefficient.

4.3.2 Concept and Algorithm for MB Hard Decoding

The basic underlying concept of MB-decoding is evident from its name i.e. the decision on whether the flipping of a variable-node's value at each iteration is taken based upon the consensus over the **majority** of incoming messages. First detailed works on the MB-based algorithms were performed by Zarrinkhat et al. [49] where he defined MB hard-decoding algorithms of varying ***order*** (w). The order w defines the *strength* of the majority i.e. how many more than the at least half the incoming messages are in agreement with each other where $0 \leq w \leq \lfloor (j-1)/2 \rfloor$. The basic per iteration computations for an order w MB hard-decoder can be summarised as:

- Messages are from the set $0, 1$ and represent the current estimate of the decoder of a particular bit.
- **At Check Nodes** : The outgoing message from a check-node results from the computation of the XOR sum of the incoming (extrinsic) messages
- **At Variable Nodes** : The outgoing message equals the originally received message, except if a $\lceil (j-1)/2 \rceil + w$ or more of the incoming messages disagree, in which case this common value is sent

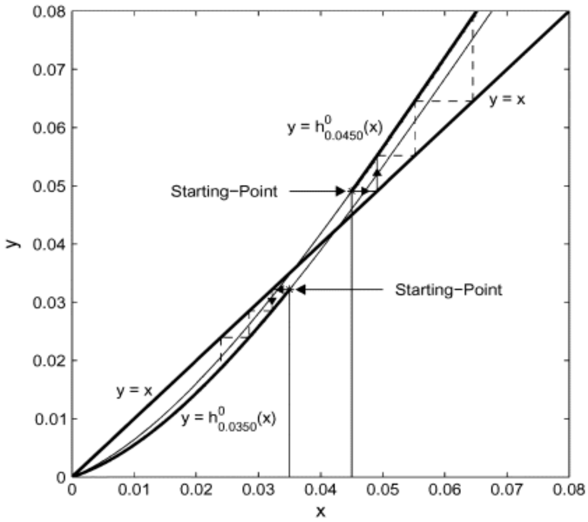
The value of w determines the order and hence the strength of the disagreement required for the change. Extreme cases are standard majority-decoding algorithm (order 0) i.e. $w = 0$, and the case where all the incoming messages unanimously agree on a value (maximum order and same as Gallager A algorithm) i.e.$w = \lfloor (j-1)/2 \rfloor$. The p_i evolution as represented by 4.4 in this case is given by the modification in the value of $p^{err-corr}(p_i)$ as

$$p_0 \cdot \sum_{l=b}^{\lceil (j-1)/2 \rceil + w} C_l^{j-1} \cdot \left[\frac{1 + (1-2.p_i)^{k-1}}{2} \right]^l \cdot \left[\frac{1 - (1-2.p_i)^{k-1}}{2} \right]^{j-1-l} \tag{4.8}$$

and that of $p^{err-add}(p_i)$ as:

$$(1-p_0) \cdot \sum_{l=b}^{\lceil (j-1)/2 \rceil + w} C_l^{j-1} \cdot \left[\frac{1 - (1-2.p_i)^{k-1}}{2} \right]^l \cdot \left[\frac{1 + (1-2.p_i)^{k-1}}{2} \right]^{j-1-l} \tag{4.9}$$

Figure 4.6 Convergence of Probability of Error for Ensemble (3,6) for $p_0 = 0.0350$ and $p_0 = 0.0450$ Regular LDPC Code decoded by MB algorithm of order 0



4.3.2.1 Important Characteristics of MB Hard-Decoding Algorithms

Majority-based algorithms are especially attractive for their remarkably simple implementation (per iteration) because of the exchange of only hard (0,1) messages alongwith their superieur performance to the classical Gallager A algorithm. Many important characteristics of the Majority-Based algorithms were studied in [49] that has led to renewed research interest in them in the recent years.

Classical graphical approach is generally employed to visualize the behavior of these iterative algorithms. Important characteristics like that of threshold values and speed of convergence can be easily depicted by means of such figures. The graphical approach, introduced originally by Gallager, has been used in similar contexts by others, including ten Brink [40]. It involves plotting down the values of p_i as a function of p_{i-1}. A use of this graphical interpretation of the convergence of p_i (denoted by h(x) function) to 0 for the ensemble of (3,6) regular LDPC codes, decoded by MB algorithm of order 0 ($w = 0$) and for two varying initial probabilities of error ($p_0 = 0.0350$) and ($p_{0.0450} = 0$) is depicted in the figure 4.6 [49]. It can be observed from the figure that depending on the channel parameter, there is either a decreasing ($p_0 = 0.0350$) or an increasing ($p_{0.0450} = 0$) trend of average error probability with iterations. For a starting point on the line $x = y$, the algorithm is stuck right from the beginning, and there will be no decreasing or increasing trend.

Threshold Value of MB Algorithms :

As clearly observed from the above figure that p_i can follow both an increasing or a decreasing trend for a p_0 and w. An important theorem with respect to the threshold value states [49] that for a given order w and a given $p_0 \in [0, 0.5]$, p_i is a decreasing sequence which converges to zero, if and only if the curve $y = p_i$ is below the line $y = x$, $\forall x \in (0, p_0)$. Threshold is conventionally defined as (p_0^*), such that for all values $p_0 < p_0^*$, $lim_{l \to \infty} p_i = 0$. Since it can be shown that p_i is an increasing function of p_0 such that $p_0 \in [0, 0.5]$ and $p_0 \in [0, 0.5]$, then using the above mentioned theorem as well, it can be seen that if for a given p_0 and an MB algorithm of order w $(= MB^w)$ p_i is decreasing and converges to zero, then for every $p_i \in [0, p_0]$, p_i is decreasing and converges to zero. Hence, an alternate definition for the threshold p_0^* of an MB^w algorithm can be defined as the supremum of all $p_0 \in [0, 0.5]$, such that the curve p_i is below the line $y = x$ for every $p_{i-1} \in (0, p_0]$.

Convergence Speed of MB Algorithms :

Using the same graphical interpretation of the evolution of p_i with respect to iterations as mentioned above, it was observed [49] that for a given p_0, the convergence speed of an MB^w at any iteration depends on how far the curve p_i is from the line $y = x$ at that iteration. The farther the curve, the faster the trend. This explains the slow convergence of Gallager A algorithm in decoding the ensemble (d_v, d_c) regular LDPC codes over a channel with $p_0 < p_0^*$ and close to p_0^* as inn this case, there is only a narrow tunnel between the curve and the line in the vicinity of $x = 0$ taking it longer to converge. Similarly it was concluded that between two majority-based algorithms used to decode the same ensemble of regular LDPC codes over the same channel, the one with smaller order eventually converges faster at the later stages of decoding. Another important result regarding the convergence of the MB algorithms was that in comparison with the Gallager A algorithm, whose convergence to zero is exponential with respect to the number of iterations, the convergence of MB algorithms is super-exponential.

4.4 Simplifying the MB Hard-Decoding Analysis for Irregular LDPC Codes

The main idea behind our work on LDPC codes was to make use of the **Wave Effect** phenomenon that exists in the irregular LDPC codes and which has been explained earlier in this chapter. Although the wave-effect concept was first introduced by Luby et al. only to serve as the intuitive reasoning behind the better performances of irregular codes, our purpose is to go further ahead and quantify this wave-effect so that it may be used eventually as a mean for the design and construction of finite-length Irregular LDPC codes. This involves quantifying the **change** in the probability of error (p_i) of a variable-node because of the addition of a new parity-check node in its neighborhood. The idea behind quantifying the wave-effect is that it would enable us to calculate the effect generated by the allocation of an extra parity-check to a variable node. By this we can construct the irregularity from a regular-graph, by allocating additional edges one-by-one, each time allocating to the variable node which results in the maximum overall decrease in the probability of error. The purpose of this section is to show how this effect can be quantified and hence measured. It was observed that the conventional analysis methods are computationally prohibitive for calculation of exact 'elite' effect quantization. Thus, we have introduced a method based on the Sum of Products of Combinations (SPC), by the help of which, the calculation of wave-effect quantization for the Irregular LDPC codes becomes feasible.

4.4.1 Hard-Decoding Analysis for Irregular LDPC Codes

Luby et al. modified the regular LDPC hard-decoding analysis for the irregular case. Based upon the earlier definitions of the functions of $\lambda(x)$ and $\rho(x)$ defining the probability distribution on the degrees of variable and check nodes respectively, the probability that a check node c receives an even number of errors, when the probability of error on variable nodes is p_i is given by

$$\frac{1+\rho(1-2p_i)}{2} \tag{4.10}$$

which is the generalization of the equation used in the regular case, taking into account the probability distribution on the degree of c. Using the similar philosophy for modifying the computations done at the variable node considering that a variable node is of degree j with probability λ_j, the recursive relationship for p_i can be given modified for an irregular code as [38] :

$$\begin{aligned} p_{i+1} &= p_0 - \sum_{j=1}^{d_v} \lambda_j \\ &\cdot \Bigg[p_0 \sum_{l=b_j}^{\lceil (j-1)/2 \rceil + w} C_l^{j-1} \cdot \left[\frac{1+\rho(1-2.p_i)}{2}\right]^l \cdot \left[\frac{1-\rho(1-2.p_i)}{2}\right]^{j-1-l} \\ &-(1-p_0) \sum_{l=b_j}^{\lceil (j-1)/2 \rceil + w} C_l^{j-1} \cdot \left[\frac{1-\rho(1-2.p_i)}{2}\right]^l \cdot \left[\frac{1+\rho(1-2.p_i)}{2}\right]^{j-1-l} \Bigg] \end{aligned} \tag{4.11}$$

However the equation 4.12 represents the generalized overall behavior of the irregular code with different variable-nodes of varying degrees. For the case where one has to quantify the difference in the p_i of different variable-nodes with varying degrees, one has to modify the original p_i recursive equation for a particular variable-node with a specific value of 'j' and 'k', instead of taking into account all possible values. Development of such a relation will describe the exact evolution of p_i of a variable-node based upon its exact number and type of neighbors instead of a generalized evolution of p_i of the whole code as given by equation 4.12. Consider a case where all the check-nodes have the same and fixed degree (d_c), while the variable node degree may be allowed to vary i.e. $d_v^n \in [3, d_v^{max}]$, where d_v^n is the degree of node n. Considering the fact that p_i on nodes with different degrees will be different as the decoding starts, we denote p_i^n as the probability of error of node n at iteration i. Thus the $p_i^{err-corr}$ in the original Gallager equation 4.4 can be modified for this irregular case as:

$$
\begin{aligned}
p_{i+1}^{n} \;=\;& p_0 - p_0 \cdot \\
& \sum_{l=\lceil (j_n-1/2)\rceil}^{j_n-1} \; \sum_{l'=1}^{C_{l'}^{j_n-1}} \cdot \prod^{v=\phi_l^{j_n-1}\{l'\}} \left[\frac{1+\prod^{k\in\psi_v}(1-2.p_i^{vk})}{2} \right] \cdot \\
& \prod_{v'\notin v}^{v'\in\{j_n-1\}} \left[\frac{1-\prod^{k\in\psi_{v'}}(1-2.p_i^{vk})}{2} \right] + \\
& (1-p_0) \cdot \sum_{l=\lceil (j_n-1/2)\rceil}^{j_n-1} \; \sum_{l'=1}^{C_{l'}^{j_n-1}} \cdot \prod^{v=\phi_l^{j_n-1}\{l'\}} \left[\frac{1-\prod^{k\in\psi_v}(1-2.p_i^{vk})}{2} \right] \cdot \\
& \prod_{v'\notin v}^{v'\in\{j_n-1\}} \left[\frac{1+\prod^{k\in\psi_{v'}}(1-2.p_i^{vk})}{2} \right]
\end{aligned}
\tag{4.12}
$$

There is a difference of the equation 4.13 with respect to its regular counterpart, as mentioned previously (Gallager's recursive equation for MB Hard-Decoding). In the regular version, for a given set **dv** representing the check-nodes associated with a variable node, the possibilities of selecting l check-nodes out of the total $j_n - 1$ are represented by means of multiplication with the factor $C_l^{j_n-1}$. This is because the probability of error p_i associated with each of the neighbors of any given check-node in an iteration is the same because the graph is regular. However, in an irregular graph the p_i associated with each variable-node at the end of an iteration is different, depending upon the number and quality of its neighbors and is represented by p_i^n. Hence the multiplication with the factor $C_l^{j_n-1}$ cannnot be anymore valid, and each combination of node in the total combinations of $C_l^{j_n-1}$ has to be treated individually, finally summing all the individual results thereby giving the true representation of probability of distribution of p_i^n in the Gallager's equation, as represented by equation 4.13

4.4.2 Modified Representation of Gallager's Equation in the form of Deltas (∆s)

As discussed in the start of this section, achieving our primary goal of wave-effect quantization involves quantizing the increasing/decreasing effect in the probability of error (p_i) of a variable-node. This stems from the addition/subtraction of its associated check-nodes, or from a change in quality (p_i^n of one or more variable-nodes in its *neighborhood*, where we define *neighborhood* of a variable-node v as the set of all variable-nodes that are associated to the node v via one or more of v's check-nodes. To facilitate this quantization analysis, our first step is the representation of the basic building blocks of the equation 4.13 in terms of different Δ, which define the change in the p_i^n of a variable-node with respect to the initial p_0 of the variable-node (which is uniform for all the nodes in case of a Gaussian channel). Hence if we define two building blocks x_i and y_i as follows

$$x_i = \left[\frac{1+\prod_{k'=1}^{k}(1-2.p_{k'})}{2}\right] \qquad y_i = \left[\frac{1-\prod_{k'=1}^{k}(1-2.p_{k'})}{2}\right] \tag{4.13}$$

we can define x_i and y_i alternatively as

$$x_i = \left[\frac{1+\prod_{k'=1}^{k}(1-2.(p_0+\Delta_{k'}))}{2}\right] \qquad y_i = \left[\frac{1-\prod_{k'=1}^{k}(1-2.(p_0+\Delta_{k'}))}{2}\right] \tag{4.14}$$

which is equivalent to

$$x_i = \underbrace{\left[\frac{1+(1-2.p_0)^k}{2}\right]}_{\mathbf{x}} + \underbrace{\sum_{k'=1}^{k}(1-2.p_0)^{k-k'}.\prod_{l'=1}^{k'}(-2\Delta_{l'})}_{\Delta_{neigh}} \tag{4.15}$$

and

$$y_i = \underbrace{\left[\frac{1-(1-2.p_0)^k}{2}\right]}_{\mathbf{y}} + \underbrace{\sum_{k'=1}^{k}(1-2.p_0)^{k-k'}.\prod_{l'=1}^{k'}(2\Delta_{l'})}_{\Delta_{neigh}} \tag{4.16}$$

where Δ_{neigh} represents that combined increasing/decreasing effect in the value of $x_i(y_i)$ because of the change in the probabilities of error of the k neighboring nodes of a check-node.

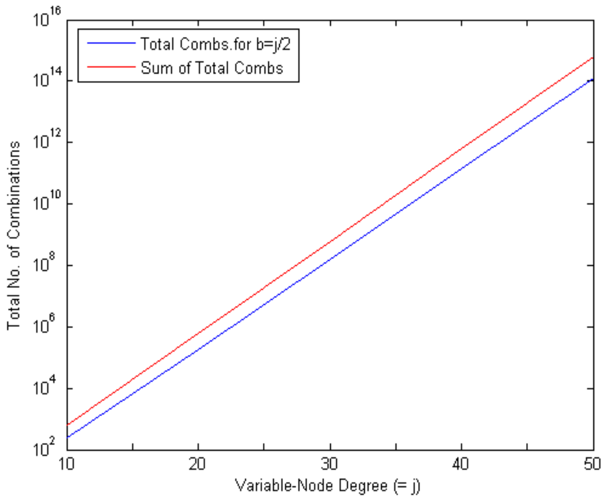
4.4.3 Classical Calculation Method of Updating p_i^n in Irregular Graphs

The simplest way to calculate the p_i^n update equation, where different neighbors may have different p_i^n, can be observed from the equation 4.13. For each value of l, the total number of possible *combinations* such that l check-nodes are selected from the vector **dv** of length $j_n - 1$, have to be found, where the elements of the vector **dv** represent the neighboring check-nodes of the variable-node v. From the combinatorial theory, it can be easily found that the total number of such combinations/subsets is

Table 4.1: Number of Combinations for Different Variable-Node Degrees

Variable Node Degree (=j)	$C^j_{b=j/2}$	$\sum_{l=j/2}^{j} C^j_l$
10	252	638
20	1.84×10^5	6.16×10^5
30	1.55×10^8	6.14×10^8
40	1.37×10^{11}	6.18×10^{11}
50	1.26×10^{14}	6.26×10^{14}

Figure 4.7 Number of Combinations required for calculation of p_i with increasing variable-node degree



given $C_l^{j_n-1} = (j_n-1)!/(j_n-1-l)!l!$ whose value can become notoriously large in the range of commonly used values for the variable-node degree as shown in table 4.1 and figure 4.7.

As evident from the data presented, attempting to calculate p_i via this simplistic method soon becomes computationally prohibitive because of the high number of combinations involved as the variable-node degrees gets increased, where each combination represents a subset vector of the complete set of a variable-node's associated check-nodes. This calculation process, being the very basic computation step involved in our calculation of Δ_{total}, can be largely simplified by means of our Sum of Products of Combinations

(SPC) approach as explained below:

4.4.4 Sum of Products of Combinations (SPC) based Calculation Method of Updating p_i^n in Irregular Graphs

The basic idea behind our proposed methodology for the calculation of p_i^n in irregular graphs, is to represent the probabililty of error of each variable node in terms of a standard probability of error p_0 and a difference Δ. This results in seperating the deltas from the original equation original and finally the problem is reduced to that of finding the sum of products of different terms, as will be demonstrated below. This can be best presented in the form of an example. We will look into the calculations involved in either of $p_i^{err-corr}$ or $p_i^{err-add}$, in Gallager's recursive equation, and the same sort of operations can be applied to the other as well. Extracting the portion of calculations pertaining to a given value of l for $p_i^{err-corr}$ in the MB Hard-Decoding equation for Irregular Graphs given by equation 4.13 can be represented as:

$$p_i^{err-corr} = \sum_{l'=1}^{C_{l'}^{j_n-1}} \cdot \prod^{v=\phi_l^{j_n-1}\{l'\}} \left[\frac{1+\prod^{k\in\psi_v}(1-2.p_i^{vk})}{2}\right] \cdot \prod_{v'\notin v}^{v'\in\{j_n-1\}} \left[\frac{1-\prod^{k\in\psi_{v'}}(1-2.p_i^{vk})}{2}\right] \tag{4.17}$$

Considering a particular case with a variable-node having a degree $j=5$ and when $l=3$, by making use of the previously introduced concept of Δs, the above equation can be written as:

$$= \sum_{l'=1}^{C_{l'}^{j_n-1}} \cdot [(x+\Delta_i)(x+\Delta_j)(x+\Delta_k)] \cdot [(y+\Delta_l)(y+\Delta_m)] \tag{4.18}$$

where i, j, k, l, m are the index value of the check-nodes and can have any value from 1 to 5, with total number of check-nodes being 5. Equation 4.18 is a sum of C_3^5 terms, each term representing an instance of the combinations resulting from the selection of 3 terms from a total of 3 terms. A one particular instance can be represented as:

$$\begin{aligned} &[(x+\Delta_1)(x+\Delta_2)(x+\Delta_3)] \cdot [(y+\Delta_4)(y+\Delta_5)] = \\ &\left[x^3 + x^2.(\Delta_1+\Delta_2+\Delta_3) + x.(\Delta_1\Delta_2+\Delta_2\Delta_3+\Delta_3\Delta_1)\right] \cdot \\ &\left[(y^2 + y.(\Delta_4+\Delta_5) + \Delta_4\Delta_5\right] \end{aligned} \tag{4.19}$$

This can be re-written in the form

$$
\begin{aligned}
= \quad & x^3y^2 + x^3y(\Delta_4+\Delta_5) + x^3(\Delta_4\Delta_5) + x^2y^2(\underbrace{\Delta_1+\Delta_2+\Delta_3}_{spc_1^5}) + \\
& x^2y(\underbrace{\Delta_1\Delta_4+\Delta_2\Delta_4+\Delta_3\Delta_4+\Delta_1\Delta_5+\Delta_2\Delta_5+\Delta_3\Delta_5}_{spc_2^5}) + \\
& x^2(\Delta_1\Delta_4\Delta_5+\Delta_2\Delta_4\Delta_5+\Delta_3\Delta_4\Delta_5) + xy^2(\Delta_1\Delta_2+\Delta_2\Delta_3+\Delta_3\Delta_1) + \\
& xy(\underbrace{\Delta_1\Delta_2\Delta_4+\Delta_2\Delta_3\Delta_4+\Delta_3\Delta_1\Delta_4+\Delta_1\Delta_2\Delta_5+\Delta_2\Delta_3\Delta_5+\Delta_3\Delta_1\Delta_5}_{spc_3^5}) + \\
& x(\Delta_1\Delta_2\Delta_4\Delta_5+\Delta 2\Delta_3\Delta_4\Delta_5+\Delta_3\Delta_1\Delta_4\Delta_5) + y^2(\Delta_1\Delta_2\Delta_3) + \\
& y(\underbrace{\Delta_1\Delta_2\Delta_3\Delta_4+\Delta_1\Delta_2\Delta_3\Delta_5}_{spc_4^5}) + \underbrace{\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5}_{spc_5^5}
\end{aligned}
\tag{4.20}
$$

Expression 4.21 indicates the presence of multiple subsets (spc_l^j) of a complete SPC_l^j set with its constituent terms of x^iy^j.A Sum of Product of Combinations given by SPC_l^j represents the sum of C_l^j product terms, where each product term results from the product of the members of a particular combination instance from the C_l^j combinations that result from the selection of l terms from a total of j terms. For example a partial subset i.e. spc_1^5 ($=\Delta_1+\Delta_2+\Delta_3$) of the complete set SPC_1^5 ($=\Delta_1+\Delta_2+\Delta_3+\Delta_4+\Delta_5$) exists with the term x^2y^2; another subset of the same order (spc_1^5) exists with the term x^3y^1 as well.All of these partial sets of Sums of Products of Combinations exist in a single instance of the sum term in equation 4.18. A total of C_l^j such instances exist in the complete summation and the partial sum of products of combinations with the same x^iy^j terms in different instances gets added to result in all the partial spc_l^j adding up to become the complete SPC_l^j set of a given order. Finally the equation 4.21 can be summarized as:

$$\sum_{x_{pow}=0}^{3}\sum_{y_{pow}=0}^{2} x^{x_{pow}}y^{y_{pow}}SPC^{j-x_{pow}-y_{pow}} \tag{4.21}$$

where the term $SPC^j_{j-x_{pow}-y_{pow}}$ indicates the Sum of Products of Combinations of order $j-x_{pow}-y_{pow}$, the order representing the l in C_l^j.

Equation 4.21 demonstrates how the original problem of $\sum_{l'=1}^{C_l^j} X_i.Y_i$ has been transformed into an addition of $(x_{pow}+1)*(y_{pow}+1)$ terms, where each of the term alone contains an $SPC^j_{j-x_{pow}-y_{pow}}$, which implies that the potential complexity of the original problem involving a large number of combinations can be potentially reduced if a computationally efficient method of calculating a SPC is derived, which we explain below how

Figure 4.8 The triangular structure of the S Matrix used for calculating SPC

$$S = \begin{vmatrix} s11 & s12 & s13 & s14 & s15 \\ s21 & s22 & s23 & s24 & 0 \\ s31 & s32 & s33 & 0 & 0 \\ s41 & s42 & 0 & 0 & 0 \\ s51 & 0 & 0 & 0 & 0 \end{vmatrix}$$

to achieve.

Problem Definition

Given a vector of deltas of length j denoted by $\delta_j = \{\Delta_1, \Delta_2, ..., \Delta_j\}$, for $1 \leq l \leq j$, what is the most efficient way to calculate SPC_l^j?

Our simplistic recursive calculation procedure can be best explained by means of visualizing the calculation steps in terms of a matrix **S**, whose dimensions for the calculation of SPC_l^j are $(SPC_1^j \times SPC_l^j)$. Each column contains different terms which add up to define SPC of a particular order l. Hence the first column is for SPC_1^j, the second for SPC_2^j and so on till SPC_l^j. The first column representing SPC_1^j will have a unique $\Delta_{j'}$ such that $1 \leq j' \leq j$ with no repetitions, hence a total of SPC_1^j terms in the first column.

Proposition

If (m,n) represents the row and column number of the above mentioned **S** matrix, then based upon the above mentioned configuration $S(m,n) = \sum_{m'=1}^{m'_{max}(n)} S(m'+1, n-1)$,

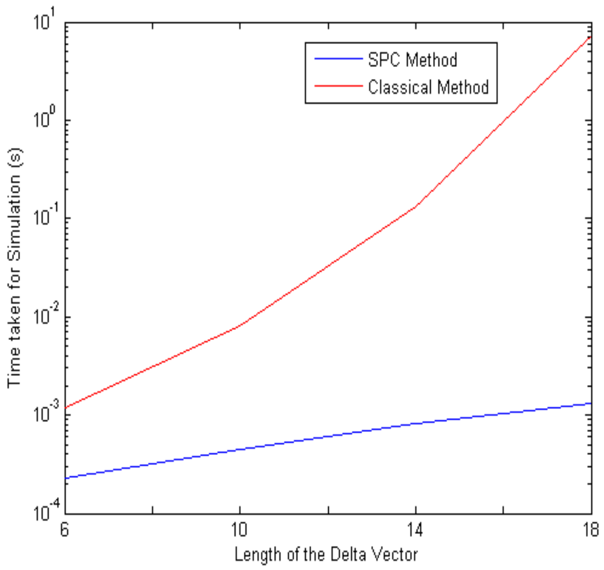
where $m'_{max}(n)$ represents the maximum index of the row of column n containing a non-zero value. Hence the matrix S will be triangular in shape as shown in figure 4.8.

A simulation-time comparison was made between the two methods for calculating the sum of products of the combinations, for varying lengths of the delta vector. Time taken for simulations by the classical method as well as by the SPC method is given in table 4.2, and the corresponding graph in figure 4.9

Table 4.2: Time taken for simulations of Sum of Products of Combinations by Two Different Methods

Length of Delta Vector	Time taken for Simulations (Classical Method)	Time taken for Simulations (SPC Method)
6	0.0012	2.25×10^{-4}
10	0.0080	4.50×10^{-4}
14	0.1311	8.16×10^{-4}
18	7.38	13×10^{-4}

Figure 4.9 Simulation Time taken for calculations of the Sum of Products of Combinations



4.5 Wave-Effect Quantization Methodology for Design/Construction of Irregular LDPC Codes

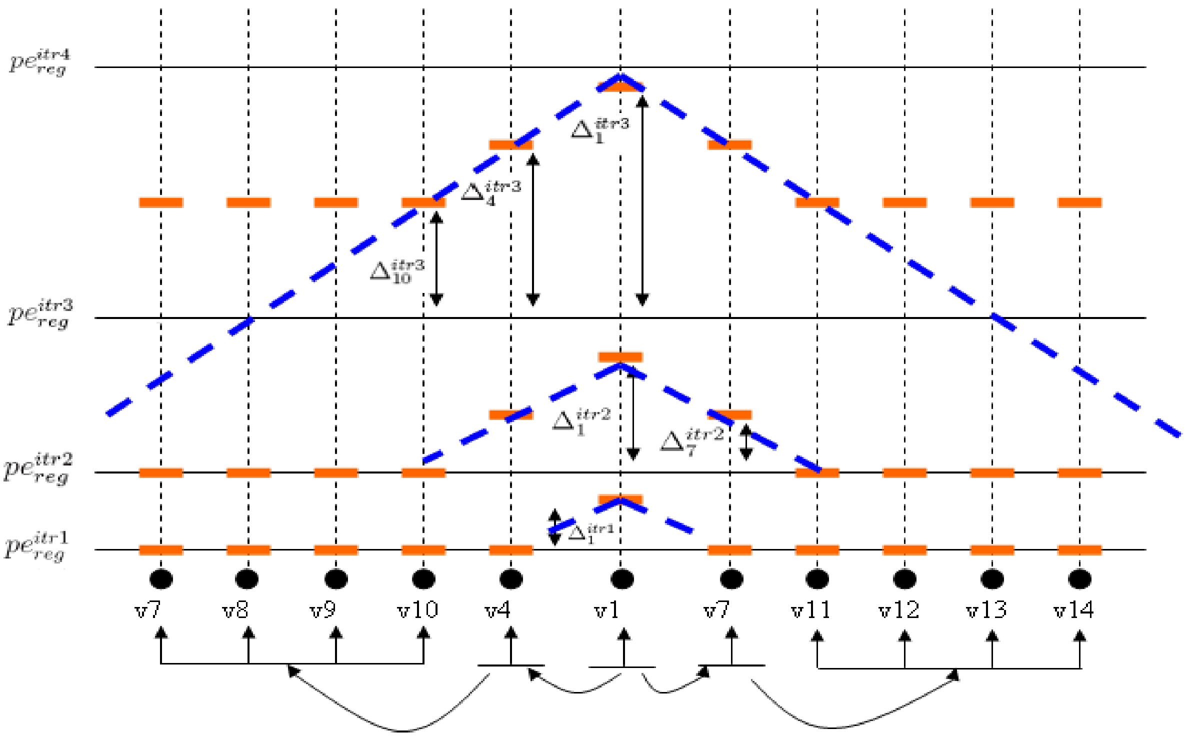
4.5.1 The Pyramid Effect

Our basic purpose is to quantify the very-popular *wave-effect* in Irregular LDPC codes, which was defined earlier in this chapter. This effect provides us with the intuitive reasoning (supported by the experimental evidence [38]) of why certain irregular codes would perform better than their regular counterparts. This intuition expresses that the high degree nodes get corrected very quickly followed by the variable nodes with slightly smaller degree, and so on down the line. Hence the name 'wave' associated with it, which suggests that the 'improvements' (= reduction in probability) caused by variable degrees with higher degree are greater than those by the nodes with comparatively lesser degree. This leads to a non-unform (like a wave with different peaks) reduction in the probability of error (p_i) of different nodes, unlike the uniform change in p_i of all the nodes in regular codes.

Once we are able to quantify these 'improvements' associated with the increase in a variable-node's degree e.g the total improvements that result from increasing d_v^n from 3 to 4 or from 4 to 5. However, to be able to calculate these 'improvements', we must understand how the p_i^n of different nodes is being updated at different iterations. In LDPC Codes, the probability of error of node n after iteration i (denoted by p_i^n) is calculated based upon the probability of error of its neighbors at the end of $i-1$ iterations. We denote by Δ_n^{itr} the improvement (reduction) in node n's p_i^n at the end of itr iterations because of the increase in its degree from d_v^{reg} to some other value d_v^n where d_v^{reg} represents the initial degree of a regular graph, which is same for all the nodes.

A depiction of this *wave* of improvements, which can be more appropriately called as the *pyramid effect* (because of its shape) is depicted in the figure 4.10. The solid horizontal lines indicate the update of p_i^n when the underlying graph is a regular one. If one of the variable-nodes (node $v1$) is made **elite** by increasing its degree from the rest of the nodes, it results in the update of p_i^n of nodes in a different manner as indicated by the short red lines. It can be seen from the figure that due to the presence of an elite node, after iteration 1 only the p_i^n of the elite node gets improved by a factor Δ_1^{itr1} while the p_i^n of the rest of the nodes remain the same as in the regular case. At the end of two

Figure 4.10 The Pyramid Effect showing the effect of an 'elite' node in different iterations



iterations, p_i^n of the elite node($v1$) as well as that of its immediate neighbors (e.g. $v7$) gets improved. It is important however to mention here that the improvement in $v2$ at the end of iteration 2 with respect to other nodes is because of the presence of increased degree but the improvement on its neighbors like that of $v7$ is because of the presence of an elite node as one of its neighbors. Similarly at the end of iteration 3, along with the elite node $v1$ and its neighbors (e.g. $v7$), the neighbors of the neighbors of $v1$ (e.g. $v10$) will get an improvement as well and the improvement in the node $v1$ in this case will not only be because of increased degree but also because of the improvement in its neighbors at the end of previous iteration. This infers that as the iterations are passed, a high-degree variable node will have its p_i^n not only because of its high degree but also because of the improvements that it has caused in its neighbors in the preceding iterations.

4.5.2 Neighborhood

Also, it is important to define the **neighborhood** of a node v_n at the end of i iterations. It is simply defined as the set of all the nodes that will be affected i.e. get an improvement Δ_n^{itr}, if the node v_n is made **elite** i.e. if its degree is increased. A tree representation of the variable-nodes corresponding to the ones cited in the pyramid effect figure is given in

Figure 4.11

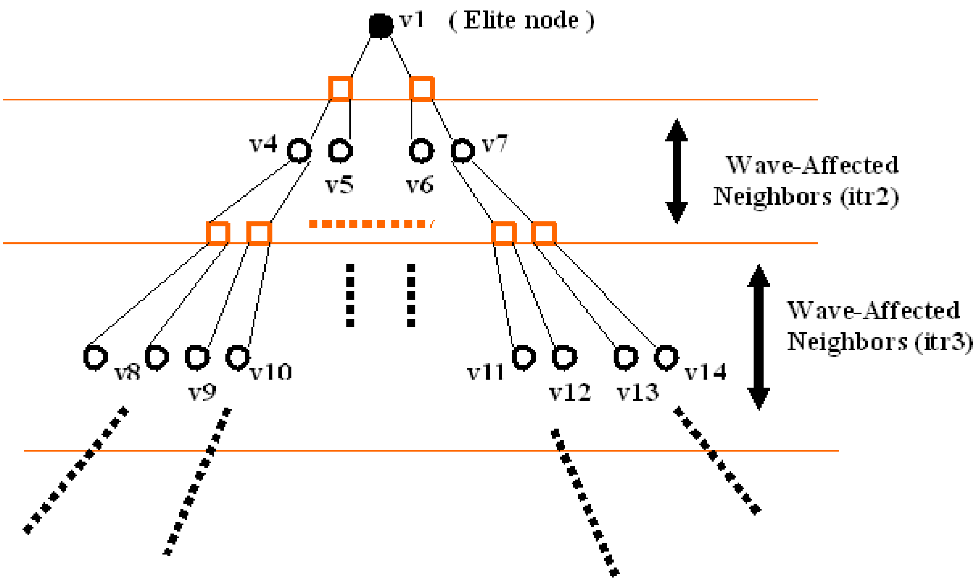


figure 4.11. Figure 4.11 helps us understand the neighborhood corresponding to different iterations. The basic aim is to identify those variable nodes that will get affected by the *wave* generated by the elite-node corresponding to a given number of iterations, which will eventually help us quantify Δ_{total}^{itr} (The total effect (improvements) caused by the elite node in all the affected variable-nodes for itr iterations.

Improvements in the p_i^n of a node takes place depending upon the improvements in the error probability, that have taken place in its immediate neighbors (variable-nodes connected to the check-nodes of the concerned node), at the end of previous iteration. Based upon this philosophy and referring to the figure 4.11, it can be observed that if the node $v1$ is made elite by an increase in its degree (=check-nodes), at the end of the first iteration only the elite-node will get improved and hence the neighborhood corresponding to iteration 1 will only consist of the elite-node i.e. $v1$ and the total improvements caused till the end of iteration 1 will be given by:

$$Neighborhood_{itr1} = \{v1\} \qquad \Delta_{total}^{itr1} = \Delta_1^{itr1} \tag{4.22}$$

Once a neighbor variable-node gets affected by the *wave* of improvements generated by the elite-node in iteration i, based on the pyramid-effect principle, it will continue to get affected by the elite-node in any number of iterations greater than i. Hence if N_{itr} denotes the set of all the variable-nodes belonging to the neighborhood of the elite-node at iteration itr, then

$$N_{itr2} = \{v1, v4, v5, v6, v7\} \qquad \Delta_{total}^{itr2} = \sum^{i \in N_{itr2}} \Delta_i^{itr2} \tag{4.23}$$

$$N_{itr3} = \{v1, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, ...\} \tag{4.24}$$
$$\Delta_{total}^{itr2} = \sum^{i \in N_{itr3}} \Delta_i^{itr3}$$

4.5.3 Greedy Irregularity Construction (GIC) Algorithm

Our research focus was the optimization of the irregularity profile for the *finite-length* LDPC codes. As mentioned earlier in this chapter, existing works on the finite-length irregular codes make use of the irregularity-profile that has been optimized based on assumptions, such as the asymptotic length of the code. Since this assumption is no longer true in the finite-length case, some works (refer to state-of-the-art section) have demonstrated that such asymptotically optimizated irregularities do not work well with the finite length codes and many-a-times because of this, at short-block lengths, the regular codes surpass the irregular ones in performance.

Hence in order to explore the behavior of irregularity at finite-lengths, we propose an algorithm, where *irregularity* is progressively added to the graph. At each step, the *response* of the code is measured/quantified for different available options of the irregularity profile and that option is selected, which results in the maximum reduction in the overall probability of error. The reduction in probability of error is calculated employing the Majority-Based hard-decoding analysis, as explained in detail earlier. We have indeed shown that the SPC (Sum of Products of Combinations) based simplified calculation methodology largely reduces the complexity involved, since this process is required to be repeated a large number of times because of *progressive* construction.

The basic idea is to start with a regular graph, void of small cycles, for a given code-length and code-rate. Then new connections are made between the check-nodes and the variable-nodes which are not previously connected to each-other. This is done by increasing check-degree of the graph by a fixed pre-determined amount. As a result of this degree increase, each check-node can now be allocated to new variable nodes. Thus, at each step, allocation of a certain check-node to different variable-nodes is compared and finally it is allocated to that variable-node which results in the maximum overall reduction in probability of error= Δ_{total}. In this manner, irregularity is automatically added in the graph. For a new parity-check allocation at each step, it is determined whether an allocation to a higher-degree node or to a lower-degree node would result in the maximum decrease in the overall probability of error.

By performing the *irregularity profile optimization* with the *construction* of the code in a single progressive allocation procedure, answers to many open questions as mentioned in section 2 can be explored. Unknowns which have generally been decided-upon based on a hit-and-trial method, e.g. how much irregularity to be added for a given code-length, how connections are to be made between nodes with varying degree, how the local and global cycle length is to be selected with respect to varying node degrees etc. can be easily explored by means of this approach. Also issues as those of how many edges to be added, maximum degree of a variable-node, are dependent on the **stopping criterion** of the algorithm which may vary as well. The stopping criterion may be based upon the fact that no cycles shorter than a particular length will be acceptable and as long as no more allocations are possible without violating this law, the procedure may be stopped. Since we have based our analysis on the Majority-Based hard-decoding method, allocation of 2 check-nodes are done at a time so that the degree of the variable-node may be kept even at all times, to keep the equality in comparison. A basic structure of the GIC algorithm can be seen from pseudo-algorithm 4.

The basic structure of the algorithm is very simple because of its greedy nature as can be observed from pseudo-algo 4. Check nodes are allocated greedily in each step to the node where their allocation results in maximum reduction in probability of error. It is obvious from the pseudo-algo that each step (allocation of 2 check-nodes) is repeated over the complete set of variable nodes so as to calculate the resulting Δ_{total}, and finally they are allocated to the variable node with maximum Δ_{total}. An interesting characteristic of this progressive allocation is the fact that it renders the possibility of using any of the Girth-optimizing algorithms, as mentioned earlier e.g. PEG [45], to be integrated in the above algorithm as a stopping condition for maximizing the performance, though we have used only the avoidance of length-4 cycles as a stopping condition.

Based upon the above mentioned algorithm irregularity was added to a regular (336,504) code for cycle4-free case and for different number of extra edges. As obvious from the figures below, the irregularity addition resulted in improving the convergence speed of the same length code.

Figure 4.12 Graphical interpretation of density evolution for ensembles regular LDPC codes, decoded by MB hard decoding, with different check-node degree (k)

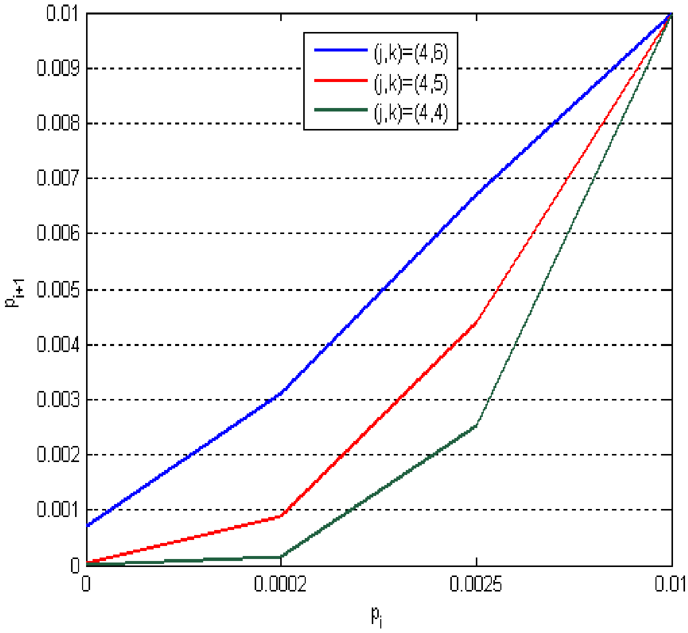
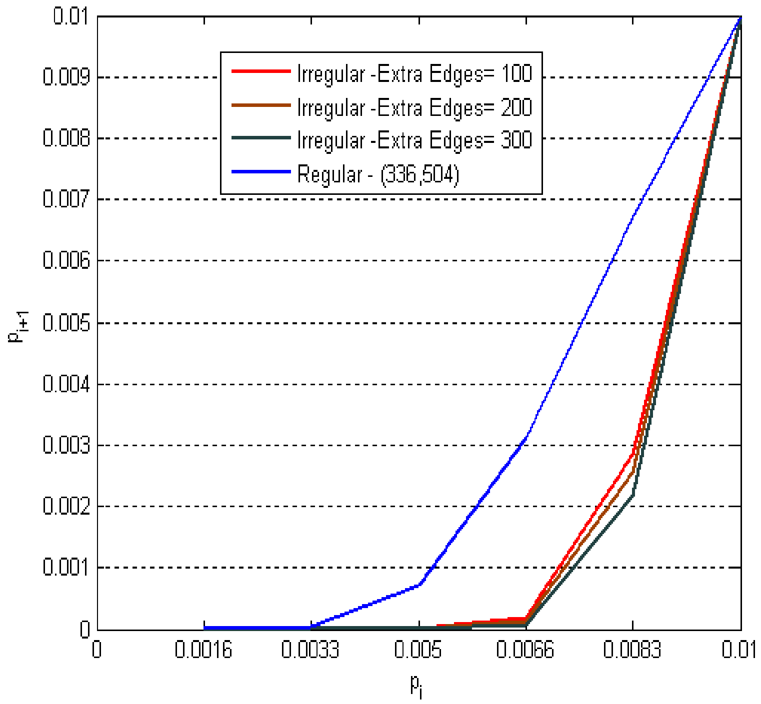


Figure 4.13 Graphical interpretation of density evolution for different regular and Irregular LDPC codes of size (M,N)=(336,504) decoded by MB hard decoding



Algorithm 4 The Greedy Irregularity Construction (GIC) Algorithm

procedure GIC(k,N,StopCond,p0,itr)
    H=makeLDPC(k,N,dc,dv)
    $p_0^{reg} = p0$
    for $i = 1 : itr$ **do**
        p_i^{reg}=FuncRegProba(p_{i-1}^{reg})
    end for
    CheckNo=1
    $H^{test} = H$
    loopNo=1
    for all $v(n) \in V$ **do**
        $pIrr_0^{loopNo}(n) = p0$
    end for
    while StopCond $\neq$ True **do** // StopCond e.g. no4cycles
        CheckIndices=[CheckNo,CheckNo+1] // 2 check allocs per itr
        for all $v(n) \in V$ **do**
            if $H^{test}(CheckIndices, n) == 1$ **then**
                goto next Variable node
            else
                $H^{test}(CheckIndices, n) = 1$
                $pIrr_{itr}^{loopNo}(n)$=FuncIrrProba($pIrr_{itr}^{loopNo-1}, Pe_{itr}^{reg}, dc, H^{test}, n, itr$)
                $\Delta_{total}^{n} = pIrr_{itr}^{loopNo}(n) - p_{itr}^{reg}$
            end if
        end for
        $n^{max} = max(\Delta_{total}^{n})$
        $H(CheckIndices, n^{max}) = 1$
    end while
end procedure

Algorithm 5 FuncIrrProba Routine for Calc. of Error Improvement

```
function FuncIrrProba(p_itr^irr, Pe_itr^reg, dc, H, EliteNode, iterations)
    for itr=1:iterations do
        Neighbors(itr)=NeighborhoodFunc(H,EliteNode,itr)
        for all v(n) ∈ Neighbors(itr) do
            for c = 1 : dv(n) do
                Δ_total^n(c) = Δ^neigh(c) + Δ^check(c))
            end for
            p_itr^irr(n) =FuncPeIrrCalc(Δ_total^n)
        end for
    end for
end function
```

4.6 Conclusion

LDPC codes because of their superior performances and simple structure are becoming academia's as well as industry's choice for the future generation of communication systems. Though irregular LDPC structures have been shown to outperform their regular counterparts, the optimization of irregular structures has been limited to asymptotic constraints, making it inappropriate for the design of finite-length irregular Codes. At the same time complex optimization techniques conventionally employed for the optimization of the irregularity profile for a given channel renders difficulty to optimize the irregularity-profile for the cases where it needs to be re-optimized a number fo times e.g. where the channel is time-varying. This chapter tries to tackle these problems, where a Greedy Irregularity Construction (GIC) has been proposed as a progressive manner to construct the irregularity profile for finite-length codes. This approach is based on the Quantization of the well-known 'Wave-Effect' phenomenon. The quantization methodology relies on the Majority-Based (MB) hard-decoding probability of error analysis for irregular codes. Where the classical calculation methods for the irregular MB hard-decoding analysis are computationally prohibitive to be used, we have proposed a Sum-of-Product-of-Combinations (SPC) based analysis methodology which has shown to reduce the simulation times by a large factor, thereby making the GIC algorithm feasible for implementation. Once a feasible calculation methodology has been devised for the quantifization of the wave-effect, a method for irregular codes optimization is devised based on it. Since, to our knowledge this is the first attempt to perform the 'quantification' of the wave-effect, we believe that a continuation of this work for multiple decoding methods and multiple channels can result in producing yet more significant results.

Chapter 5

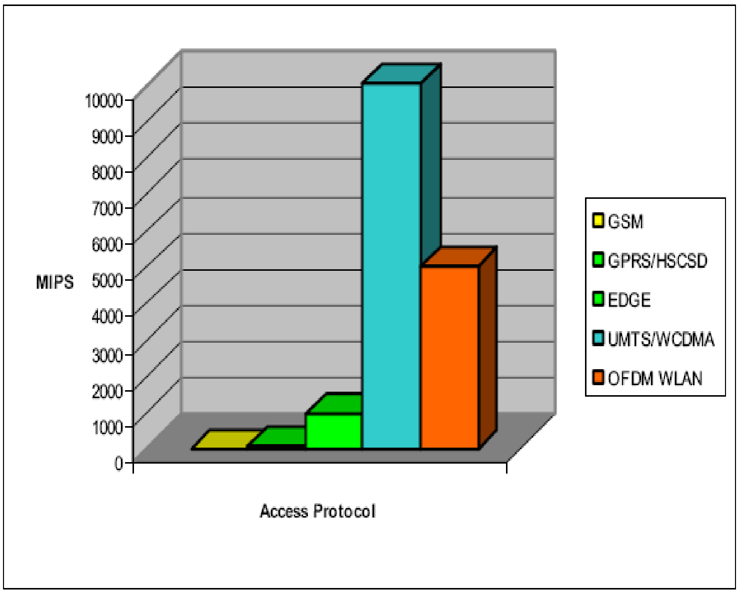
Algorithm-Architecture Co-Optimization for Delay-Constrained Link-Adaptation Algorithms

5.1 Introduction

In an adaptive communication system, the parameters of the system have to be re-evaluated each time the channel changes significantly, where the time it takes to move from one channel condition to another can be defined as the *adaptation interval.* The time required for the evaluation of the parameters for a particular channel state is therefore required to be much smaller than the adaptation interval, for the system to benefit from the adaptive process. In other words, the time for the re-evaluation of adaptive algorithms in each adaptation interval is strictly required to be under a particular time constraint, where this constraint is defined by the system specifications and the channel conditions at that particular instant. For a given maximum computation-time constraint, the smaller the computation-time of the adaptive algorithm in an adaptation interval, the greater the gain or the time available to make use of these adaptive parameters.

Due to this particular interest in the computation time of adaptive algorithms, the basic aim of our thesis till now has been to expose to the reader the most popular forms of adaptation (bit-loading, power-allocation etc.) in an adaptive multi-carrier system and to

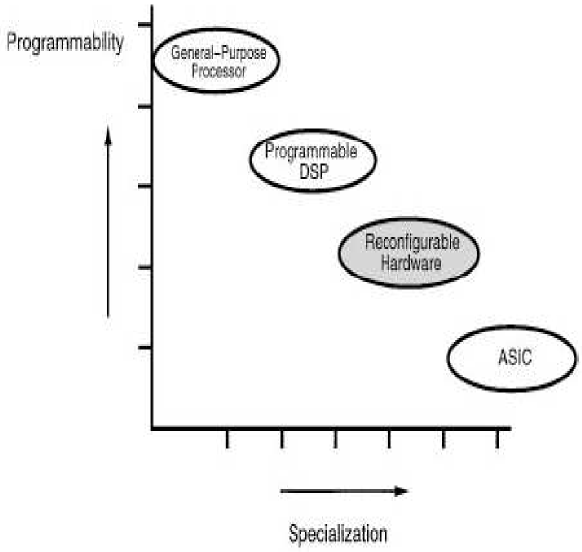
Figure 5.1 Computation Requirements of Different Protocols/Standards



the complexity reduction methods of the corresponding algorithms that we have proposed. These improvements or the reductions in complexity have been achieved via fundamental algorithmic changes in the algorithms which eventually leads to shorter computation time with no loss in performance as shown in chapter 2-4.

As much the running-time of an algorithm is dependent on its algorithmic complexity, it equally depends on the underlying implementation platform on which the algorithm is being run. The use of the best(fastest) implementation platform is not the answer for the cases where the constraint on the running-time of the algorithm is varying with respect to time, as in our concerned case of adaptive communication systems.However, the advent of the flexible or re-configurable computing paradigm seems to answer well the need for varying computational needs. Recent years have seen a grown interest in the use of re-configurable implementation platforms for the signal processing purposes [50–52]. Its motivation comes from multiple directions, where the popularity of Software/Cognitive radios [53, 54] is one of the biggest reasons for the interest in re-configurable platforms for signal processing purposes. Amongst other desired characteristics of software radios, its multi-mode functionality renders the possibility to different modes of communications from a set of multiple standards/protocols from the same transceiver at different times. Since computational requirements of different standards/modes vary significantly, as shown in the figure 5.1, the use of reconfigurable computing in this context leads to a significant economy of energy and resources.

Figure 5.2 Computation Requirements of Different Protocols/Standards

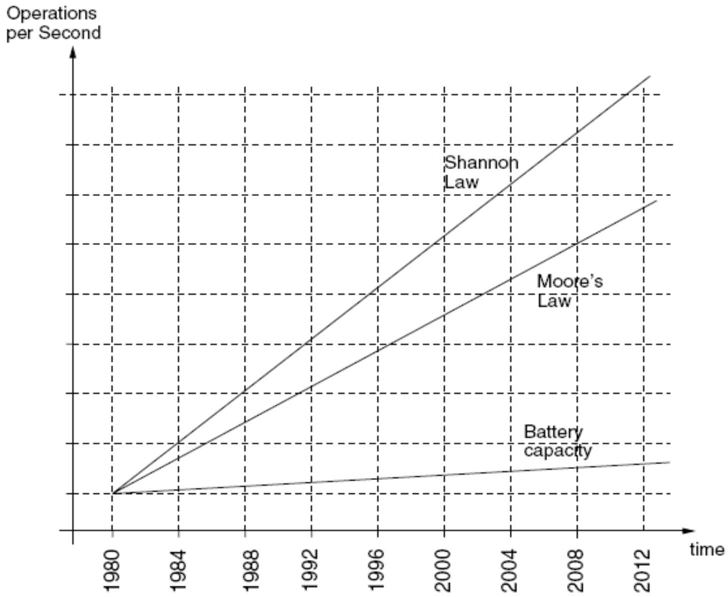


The trade-off between computational power and flexibility that exists in re-configurable hardwares in comparison with other commonly employed implementation platforms is represented by figure 5.2

The future of reconfigurable computing for DSP systems will be determined by the same trends that affects the development of these systems today: system integration, dynamic reconfiguration, and high-level compilation. DSP applications are increasingly demanding in terms of computational load, memory requirements, and flexibility. Traditionally, DSP has not involved significant run-time adaptivity, although this characteristic is rapidly changing. The recent emergence of new applications that require sophisticated, adaptive, statistical algorithms to extract optimum performance has drawn renewed attention to run-time reconfigurability. Major applications driving the move toward adaptive computation include wireless communications with DSP in hand-sets, base-stations and satellites, multimedia signal processing [114].

The primary trend impacting the implementation of many contemporary DSP systems is Moore's Law, resulting in consistent exponential improvement in integrated circuit device capacity and circuit speeds. According to the National Technology Roadmap for Semiconductors, growth rates based on Moore's Law are expected to continue until at least the year 2015 [115]. At the same time, the bandwidth/throughput of the communication systems in market, commonly defined by Shannon's Law, is increasing at an equally high rate. The qualitative graphs in figure 5.3, depict the increase of algorithmic

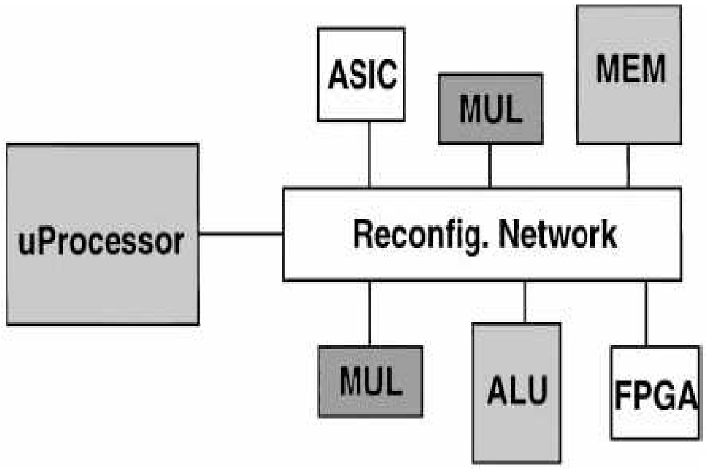
Figure 5.3 The Evolution of Computational Requirements (Shannon's Law) with Advances in Implementation Technology (Moore's Law)



demand compared with the increase of computational capability offered by technology improvement.

It can be observed that algorithm complexity, driven by Shannon's law, can not be tackled by technology alone, bound to Moore's law. For this reason, standard programmable architectures appear increasingly insufficient to handle computational demands. As a result, some of the corollaries of Moores Law will require new architectural approaches to deal with the speed of global interconnect, increased power consumption and power density, and system and chip-level defect tolerance. Several architectural approaches have been suggested to allow reconfigurable DSP systems to make the best use of large amounts of VLSI resources. All of these architectures are characterized by heterogeneous resources and novel approaches to interconnection. The term system-on-chip is now being used to describe the level of complexity and heterogeneity available with future VLSI technologies. Figure 5.4 illustrate various characteristics of future reconfigurable DSP systems. These are not mutually exclusive and some combination of these features is predicted to emerge as the suitable implementation platform in future, based on driving application domains such as wireless handsets, wireless base-stations, and multimedia platforms. Figure 5.4, taken from [116], shows research efforts to condense DSPs, FPGA logic, and memory on a single substrate.This work focuses on selecting the correct collection of functional units to perform an operation and then interconnecting them for

Figure 5.4 A model of the architectural platform structure for future Signal Processing Applications



low power. This same paradigm of the selection of a correct combination of the functional units so as to respond to the run-time computational needs of adaptive algorithms is investigated in this chapter.

Whereas a large number of works exist in the use of re-configurable/tunable platforms for various communication systems and signal processing algorithms [56, 117, 118], very few works exist for the use of these re-configurable(adaptive) hardware platforms for the case of adaptive resource-allocation algorithms [58, 59, 119, 120]. As our thesis mainly deals with the computational complexity of the adaptive resource allocation algorithms in multicarrier systems, we strongly feel that the paradigm of re-configurable computing can be exploited in this context for further reduction in the running-time of the adaptive algorithms, thereby leading to performance gains or even enabling the use of adaptive algorithms to some cases (stringent maximum computation-time constraints on adaptive algorithms) where their use would not have been possible otherwise. This chapter, therefore explores the context where the architecture/processor resources are run-time tunable and a link between the theoretical performance parameters (coherence time etc.) and the hardware resources (ALUs etc.) is established to increase the overall system performance. More precisely, we have explored a classical Superscalar processor architecture by means of SimpleScalar [61] tool, the details of which will be provided in section 3 of this chapter, to extract the performances of a given algorithm with different processor architecture parameters. The choice to select amongst a range of different architecture/processor resources (functional units, caches, pipeline characteristics etc.) leads to the research-area of *Design Space Exploration* [121]. In order to avoid prohibitive simulation times with exhaustive design space exploration, we have employed Genetic Algorithms, which greatly reduce the time to arrive at the optimal architecture configuration for a given algorithm, and whose functionality and characteristics will be given in detail later in this chapter.

Th next section relates state-of-the-art on the use of re-configurable computing for wireless and especially for the link adaptation algorithms. Section IIIdr will give a detailed introduction of the tools (SimpleScalar) and techniques (Genetic Algorithms) employed in our methodology. Section IV will describe the framework of our Algorithm-Architecture run-time co-optimization methodology along-with simulation and results. Finally this framework will be applied to three different case-studies in section V along with the conclusion and perspectives in section VI.

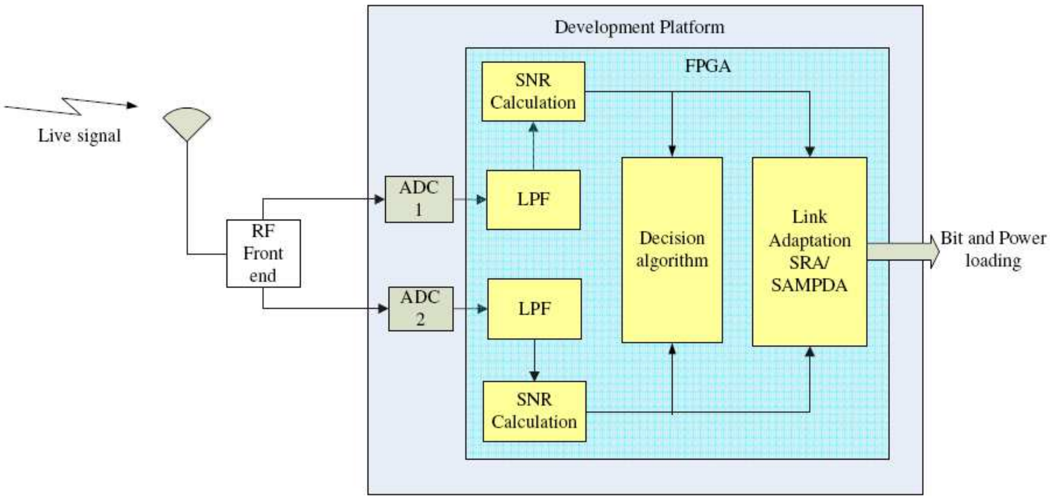
5.2 State of the Art

Works related to the use of changing architecture behavior for the continuously changing needs of link-adaptation algorithms, have to be looked from both the theoretical as well implementation side. As such we believe that this aspect is largely unexplored and very few attempts have been made to tackle this joint algorithm-architecture aspect in context of link-adaptation algorithms in time-varying channels.

5.2.1 From Theoretical Perspective

From a pure theoretical aspect, precisely we will use the *bit-loading* algorithms as the link-adaptation method. Recent works relating to bit-loading algorithms and the details on different bit-loading algorithms, that have been used by us, are given in chapter 2. In a Frequency-Division-Duplex (FDD) system the channel cannot be assumed to be reciprocal (the channel conditions on which it has received information are not the same as the ones on which it is going to transmit), and hence each time the channel varies significantly, the new channel state has to be sent back from the receiver to the transmitter via a feedback loop, which incurs a feedback delay. Therefore, in a closed-loop adaptive system, the overall delay till the availability of the optimal bit-configuration in an adaptation interval includes both the feedback delay and the delay incurred by execution of the link-adaptation algorithm. The overall delay has a strong impact on the

Figure 5.5 A model of the Re-configurable architectural platform used for implementation of Link-Adaptation Algorithms in [Kulkarni07]



performance of an adaptive system. Conventionally, the delay introduced by the link-adaptive algorithm has not been considered as a source of significant delay, introduced in the system. This is mainly because of the use of statistics-based bit-allocation algorithms in practical wireless systems, which incur negligible computational delays. However, with the increasing processing capability of DSP processors used for MAC and Link-layer algorithms, the feasibility of implementing computationally expensive subcarrier-based link adaptation algorithms is increasing day-by-day. With increasing use of subcarrier-based adaptation algorithms, their computational delay will play an important part in determining the overall delay of the system. Some notable works dealing with the effect of delay on the performance of an adaptive system can be found in [122] and [123].

5.2.2 From Implementation Perspective

From the implementation perspective, we have cited in the introduction section some important works related to the use of re-configurable architectures for the signal processing algorithms in a communication protocol. Some other interesting works can be found in [55] where a MP-SoC (MultiProcessor-System-on-Chip) architecture has been employed as the implementation platform for an adaptive OFDM cognitive radio. In [56], the mapping of various baseband algorithms in the HiperLan protocol was investigated over a flexible architecture, and the resources (ALUs etc.) were allocated on-the-need basis to different algorithms. Similarly [57] investigates the use of FPGAs for the implementation of an OFDM system.

More particularly, since we are concerned with the implementation aspects of the bit-loading/link-adaptation algorithms, Cudnoch [58] has given a DSP implementation of an adaptive bit-loading algorithm, intended for a wireless multicarrier system. The performance of the implementation was studied in terms of throughput, given time varying conditions of IEEE Std. 802.11a. In [59], implementation aspects of the link adaptation algorithms with respect to the time-varying constraints of a WiMax channel are targeted. In [60] the optimized design of a couple of link-adaptation algorithms for OFDM based mobile broadband wireless access is implemented and its feasibility evaluated in the framework of WiMax specifications and similarly in [59] the feasibility of dynamic partial re-configuration of an FPGA is analyzed for an OFDM system with WiMax specifications. However, no effort was made to dynamically allocate the resources based upon the constraints put-forward by the channel (coherence time etc.), which is the goal of this chapter.

Concerning the use of Genetic Algorithms for the design-space exploration of a configurable processor architecture, some notable works in this regard, though not necessarily for link-adaptation algorithms, include those of [124–126].

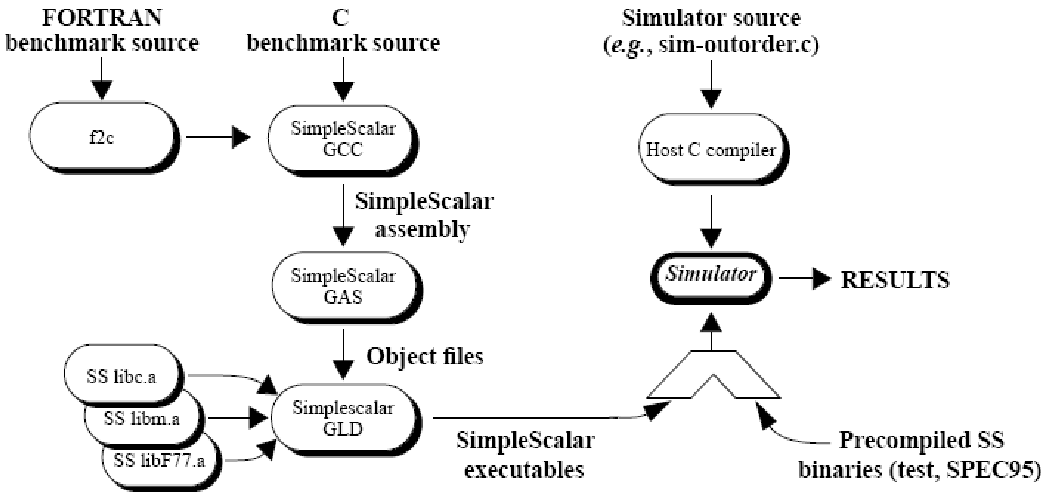
5.3 Tools (SimpleScalar) and Techniques (Genetic Algorithms) Employed

This section will introduce the optimization techniques and the tool which were employed for the investigating the relationship between the underlying implementation architecture of a bit-loading algorithm and the channel delay constraints represented by the channel coherence time for different scenarios.

5.3.1 SimpleScalar Tool and the Design Space of the Superscalar Architecture

SimpleScalar is an execution driven cycle accurate instruction set simulator (ISS) of a superscalar microprocessor [61]. A complete development chain (compiler, debugger, profiler) comes with the tool, which allows the quick porting of any ANSI C application to SimpleScalar. The SimpleScalar toolset is composed of a gcc compiler ported for the

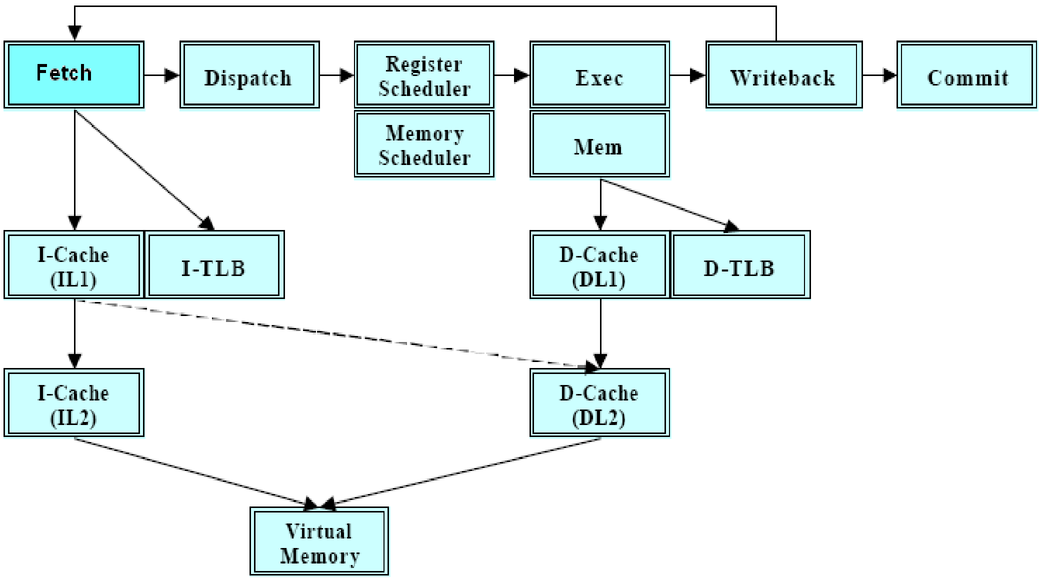
Figure 5.6 Working Methodology of SimpleScalar Toolset



SimpleScalar architecture which generates SimpleScalar binary files. The SimpleScalar assembler and loader along with the necessary ported libraries produce SimpleScalar executables which can be fed directly into one of the provided simulators. The simulator themselves are compiled with the host platforms native ANSI C compiler. The basic functioning of the SimpleScalar tool can be represented by figure 5.6 The support libraries can be modified in that case the glibc source must be installed, modified and built. The simulators come equipped with their own loader and thus one does not need to build the GNU binary libraries to run simulations. The toolset comes with a variety of simulators ranging from untimed functional simulators to cycle-accurate complex simulators.

The SimpleScalar microprocessor micro-architecture is derived from the MIPSIV ISA with a semantics which is a superset of MIPS [80] with a few exceptions: (1) delay slots are not used therefore instructions succeeding load, stores and control transfers are not executed, (2) load and stores support 2 addressing modes: indexed and autoincrement/decrement (3) a square root instruction is included and (4) there is an extended 64 bits instruction encoding. The SimpleScalar microprocessor models an out-of-order superscalar architecture based on a RUU (Register Update Unit), as shown in figure 5.7. The RUU exploits a reorder buffer to automatically rename registers and hold the results of pending instructions. However, completed instructions are retired in program order to the register file. The microarchitecture supports speculative execution. The memory system uses a load/store queue and store values are placed in this queue if the store is speculative. Load instructions are dispatched to the memory system when the addresses

Figure 5.7 Pipeline Structure of a SuperScalar Architecture



of all previous stores are known and loads may be satisfied either by the memory system or by an earlier store value reading in the queue if their addresses match. Speculative loads may generate cache misses but speculative TLB misses stall the pipeline until the branch condition is known.

The rich set of the various tunable cache, functional units and pipelining parameters of the processor are described in the table 5.1, where the default values are indicated with bold. The details on each of these parameters can be found in [61]

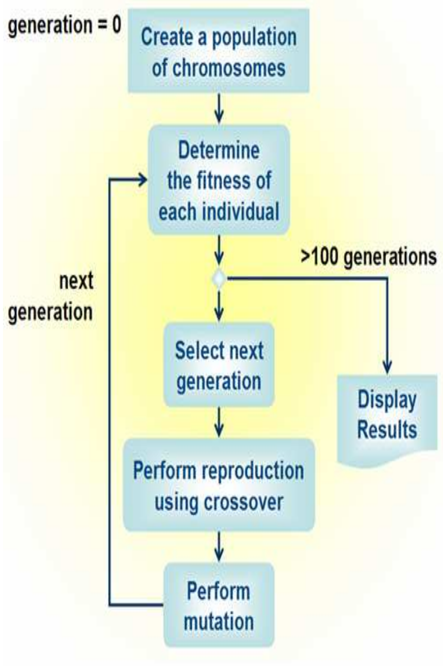
5.3.2 Genetic Algorithms for Design Space Exploration

For the class of optimization problems which cannot be directly solved, efficient probabilistic algorithms are required whose solution is approximately optimal. Hill-climbing, Simulated Annealing, Monte-Carlo methods are all part of such approximate optimization solutions. For small spaces, classical exhaustive methods usually suffice; for larger spaces special artificial intelligence techniques must be employed. Genetic Algorithms (GA) are among such techniques; they are stochastic algorithms whose search methods model the natural phenomena of genetic inheritance and darwinian strife for survival. GAs have been quite successfully been applied to optimization problems like wire routing, scheduling, adaptive control, transportation problems etc. The structure of a simple genetic algorithm is the same as the structure of any evolution program and at each iteration

Table 5.1: Tunable Parameters of the Superscalar Architecture by SimpleScalar

Module	Parameter Name	Possible Values
Instruction Cache (Number of Sets)	nsets	64,128,**256**,512
Instruction Cache (Block Size)	bsize	8,16,**32**,64 (Bytes)
Instruction Cache (Associativity)	assoc	1,2,**4**,8
Data Cache (Number of Sets)	nsets	64,128,**256**,512
Data Cache (Block Size)	bsize	8,16,**32**,64 (Bytes)
Data (Associativity)	assoc	1,2,4,8
Integer ALU	ialu	1,2,3,**4**,5,6,7,8
Integer Multiplier	imult	**1**,2,3,4,5,6,7,8
Floating-Point ALU	fpalu	1,2,3,**4**,5,6,7,8
Floating-Point Multiplier	fpmult	**1**,2,3,4,5,6,7,8
Register Update Unit (RUU) Size	ruu	2,4,8,**16**,32,64,128,256 (Instructions)

Figure 5.8 Structure of a Basic Genetic Algorithm



the relatively *good* solutions reproduce, while the relatively *bad* solutions die. The basic structure of a Genetic Algorithm is given in figure 5.8

Important components of a GA can be identified as:

- A genetic representation for potential solutions to the problem and a way to create an initial population of potential solutions
- During iteration *t*, a genetic algorithm maintains a population of potential solutions and each solution is evaluated by means of a *fitness function* to give it a measure of *fitness*.
- A new population (iteration *t+1*) is formed by selecting the best fitted individuals while some members of this population undergo genetic operations, such as Crossover and Mutation. Crossover combines the characteristics of two parent solutions to form two offspring solutions containing some characteristics of both parents, while mutation arbitrarily alters one or more characteristics of a selected solution by a random change with respect to a given mutation rate.

5.3.2.1 Population

A population is initiated of legal solutions, selected by choosing random input values. There are no fixed rules for how large the population should be. The answer is dependent upon the type of problem. For a simple problem with a regular search space a small population of 40 to 100 will probably be sufficient. For larger more complex problems and especially those with an irregular search space larger populations of 400 or more are recommended. The clue is diversity : a diverse population, i.e. a large one will tend to search out niches. In engineering terms that means finding elusive, difficult to find solutions to problems.

5.3.2.2 Fitness

The fitness of individual chromosomes is a relative matter. For example when maximising a function; if one individual has a higher value than another, then the first individual is considered fitter. Things get a little more involved with multi-criteria problems. In these cases comparisons can be carried out to see if an individual dominates other members of a population by taking all criteria into consideration. If they do, they are considered fitter. The most dominant, i.e. those who dominate all others, are referred to as Pareto solutions. These are considered as candidate solutions to whatever problem is being solved.

5.3.2.3 Selection of the Fittest

GAs operate over a number of generations. Following the evolutionary theme of this method, this means fitter solutions will tend to survive to the next generation. The selection method employed by many approaches is the roulette wheel selection process. In nature all individuals have a chance of surviving from one generation to the next. Fitter solutions (i.e. those the most dominant ones) have a better chance of survival than the weaker, more dominated individuals, but weaker individuals still have an opportunity of surviving. A Roulette Wheel [127] based selection process is commonly employed.

5.3.2.4 Crossover

Nature generates the next generation using a mating process. As a result two parents create offspring, who consist of the genetic material of both parents. These offspring can be weaker or fitter than their parents (or similar). If they are weaker they will tend to die out and if they are stronger their chances of survival are better. GAs try to replicate this by using a crossover operator. This emulates the mating process by exchanging chromosome patterns between individuals to create offspring for the next generation.

5.3.2.5 Mutation

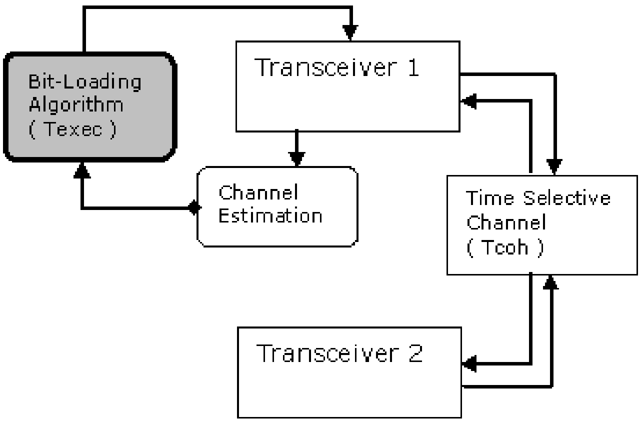
Mutation exists in nature and causes an unanticipated change in a chromosome pattern. This can result in a much weakened individual and occasionally a much stronger one. Either way the principle behind mutation from an evolutionary point of view is that it occurs rarely, spontaneously and without reference to any other individual in the population. If the change is beneficial to the general population, then that individual will tend to survive and will pass this trait on in future replication processes. Because of the way that GAs represent individuals, this process is a very simple one and a typical mutation operator is relatively easy to implement. These processes occur very infrequently otherwise they would have a disruptive effect on the overall population.

Further details on Genetic Algorithms can be found in [127]. The values for various parameters that we employed for our usage of genetic algorithm (population size, probabilities of applying different genetic operators etc.) will be given in the next section.

5.4 Algorithm - Architecture Co-Optimization for Delay Constrained Link-Adaptive Systems

Wireless channels, because of the presence of Inter-Symbol Interference, are selective both in time and frequency. If in addition to this a user is moving and significant amount of doppler is present, the coherence-time of the channel gets greatly reduced. For adaptive systems to work in such channels, the adaptation needs to take place very rapidly corresponding with the variations in the channel. Starting from the networks, where user have low mobility (WiFi), now broadband wireless (WiMax) is targeting those scenarios where the user speed may vary from pedestrian speeds to the high-velocity inter-city trains (200 km/h). This strengthens the need to increase the convergence time of link-adaptation (bit-loading) algorithms or the development of those methodologies that respond on run-time to the varying delay-constraints of the time-varying channel.

Figure 5.9 An Adaptive Time-Division Duplex (TDD) System



Our first response to such problematic was to improve the existing bit-loading methodologies from an algorithmic point of view and hence the development of the 3-dB subgroup algorithm [22] which has proven to be far-less complex than some other competitive bit-loading algorithms. However, for the case where even the best algorithm convergence time supercedes the coherence delay-constraints of a fast-varying channel, other methods must be looked for to improve the running-time of the algorithm.

One such possibility is to explore the optimal underlying architecture parameters with respect to a given algorithm and hence obtain a reduction in algorithm running-time by optimizing the architecture parameters for a given algorithm. With the increasing use of re-configurable implementation platforms (FPGAs, programmable processors etc.), the idea of optimizing processor configuration becomes realizable.

The basic architecture of a Time Division Duplex (TDD) system is shown in the figure below. A receiver feedback for the updated Channel State Information (CSI) is not required in TDD systems because based on the reciprocity assumption, the channel condition with which information is received can be considered for transmission as well.

Our proposed work methodology is depicted in the figure 5.10. We propose that initially based upon the system parameters (No. of subcarriers, No. of bits to be allocated, coherence time etc.), a database for the performances of different processor architecture configurations is created for all the algorithms available. The fitness function of this genetic algorithm is the running time of an algorithm over a particular architecture, which requires actual execution over the processor. Since this execution and evaluation of a large number of configurations (individuals) is not possible at run-time, considering the rapidly varying time-selective wireless channel, a database is created initially, linking the different configurations with their corresponding total execution time.

This gives the possibility of selecting at run-time the best processor configuration corresponding to an algorithm and to the varying coherence time constraints and is based upon the assumption that for a given algorithm and given wireless system parameters, the optimal processor configuration remains the same for different channel scenarios, which is what we expect because the form of the algorithm does not change.

Then an algorithm with default processor parameters is assigned for the bit-allocation process, and checked whether it meets the present coherence time delay constraints or not; if not, another algorithm is chosen. Once all the algorithm options have been exhausted, the best configuration meeting the delay constraints is selected and applied for the process of bit-allocation. In this regard it is important to mention that this proposal of using different processor configurations at run-time will get more feasible and will result is higher performance gains as the programmable DSP processor technology is improved. Also, only the fitness criterion of total execution time is considered in this work, which is the most relevant in delay constrained time varying channels. However, the same methodology can be extended to include other system constraints such as the total energy consumption and total silicon area for different processor configurations. For this purpose, the single objective Genetic Algorithm for processor configurations optimization will have to be replaced by the Multi-Objective Genetic Algorithm, which takes into account all the objectives and constraints at the same time.

We employed the SimpleScalar tool [61] and an underlying Superscalar processor architecture to calculate the exact number of execution cycles of three algorithms for optimal bit allocation. The number of execution cycles were evaluated for different cases corresponding to different number of subcarriers as shown in the horizontal axis of the figure 5.11 , where the first number indicates the number of subcarriers and the latter the total number of bits to be allocated. All of these simulations were made employing the default architecture (No. of ALUs, pipeline width etc.) parameters as described in the previous section.

It is evident from the figure 5.11 that our algorithm takes a lot fewer execution cycles than the other algorithms. To see the impact of the underlying architecture parameters, we implemented the three algorithms for a given set of system parameters (Number of Subcarriers, Number of Bits to be allocated etc.) for different architecture configurations.

Figure 5.10 Adaptive Algorithm-Architecture Co-Exploration Framework

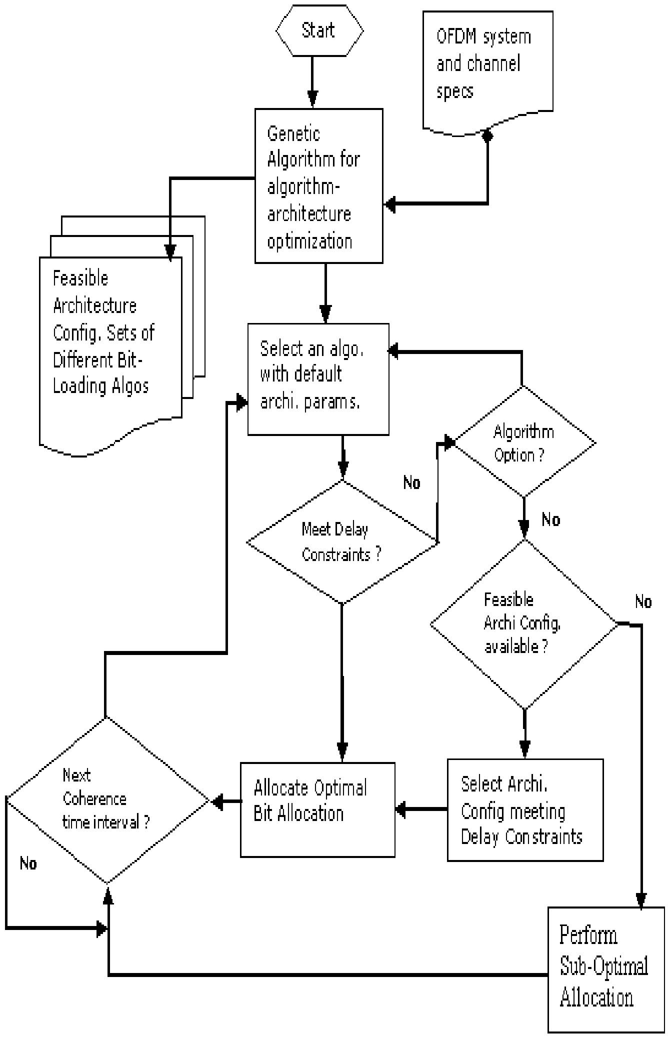


Figure 5.11 No. of Execution cycles for different bit-loading algorithms for Different System Parameters

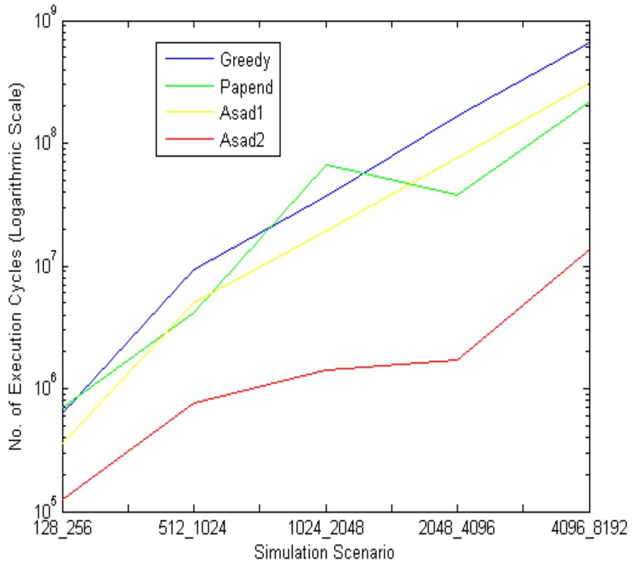
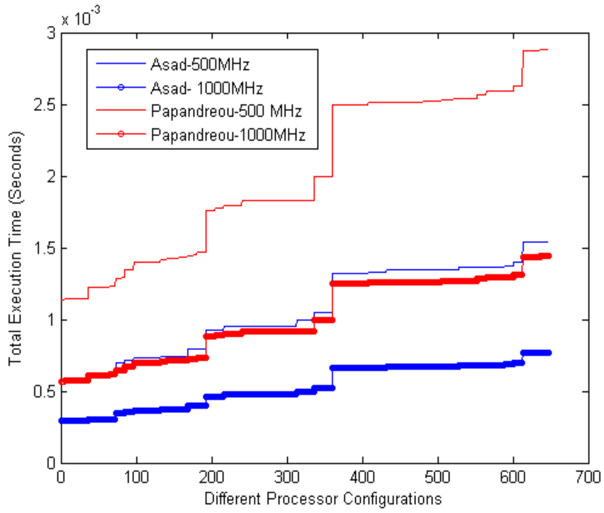


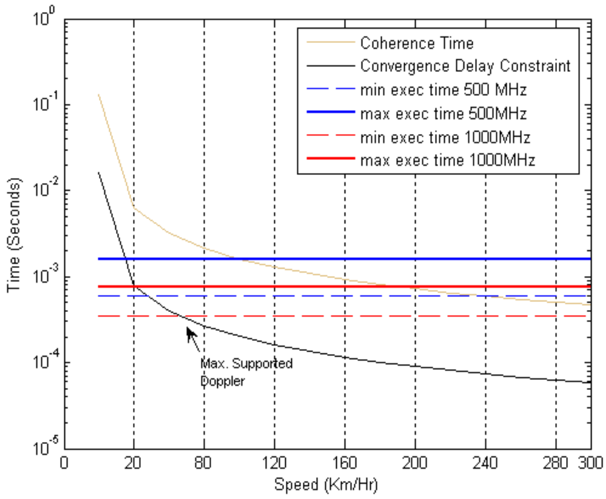
Figure 5.12 Exact convergence time of different algorithms for different processor architecture configurations



The exact algorithm convergence time (in seconds) is given in figure 5.12 for the concerned algorithms, over various architecture configurations.

To evaluate the gain in the execution times with realistic values, the vehicular speed range and the operating carrier frequency pertaining to a possible WiMax scenario were employed, allowing to calculate realistic coherence time constraints. The maximum

Figure 5.13 Coherence time



doppler shift [128] corresponding to the operation at 3.5 GHz (selected as a middle point in the 2-6 GHz frequency range) is given by

$$f_d = v/\lambda = v/0.086m \tag{5.1}$$

where v represents the user velocity. The coherence time of the channel which is dependent on the maximum doppler spread is given by [128]

$$T_c = \sqrt{\frac{9}{16\pi {f_d}^2}} \tag{5.2}$$

If we define the maximum algorithm convergence time constraint as 1/8th of the coherence time, then the coherence time as well as the respective delay constraint for upto a maximum of 300 km/h vehicular speed is given in the figure 5.13

Based upon the best convergence time of our algorithm as well as that of Papandreou et al. as given in the figure 5.12, it can be observed that by varying the underlying architecture parameters, the range of coherence times for which the algorithm was able to converge within the given delay constraint gets increased, and can be further increased as the processing frequency of the underlying processor is increased.

However it is important to mention here that it does not suffice to select that configuration which results in the least execution time, but that configuration is selected which barely meets the hard delay constraints so as to economize on the use of the area and energy. In the continuation of this work, we plan to employ Multi-Objective Genetic

Algorithms so as to pick that configuration which is optimal with respect to multiple constraints of running time, total power consumption and total area usage etc.

5.5 Conclusion

Efficient resource allocation algorithms are the key to optimize the usage of scarce bandwidth and other system resources (total energy, area etc.). The delay associated with the running-time of the resource-allocation algorithms incurs a strong impact on the performance of the system. Conventionally, only algorithm level modifications are proposed for the link-adaptation algorithms so as to reduce their complexity or the corresponding delay. The novel paradigm of Run-Time-Reconfiguration (RTR) opens new possibilities of reducing the computational delay of the link-adaptation algorithms by exploiting the architectural flexibility of the underlying platform on the need basis. Hence, it is the goal of this chapter to run-time tune the parameters of the processor on which the link-adaptation algorithm is being run, in order to adapt to the time-varying coherence-time needs of the channel.

Earlier [21] we proposed an improvement from algorithmic point of view to the optimal discrete bit loading algorithm which resulted in reducing the conventional algorithm complexity by a significant amount. In this work, we have explored the options where the algorithm convergence time can be further improved by run-time selection of the underlying processor architecture, which results in meeting the convergence time delay constraint corresponding to a given coherence time. It was observed that a run-time tuning of the processor parameters results in the operation of link-adaptation algorithm at higher doppler frequencies/user mobility speeds. The usage of Genetic Algorithm helps avoid the large simulation times which would be required otherwise for evaluating the complete design space of a processor architecture. Finally, we believe that the proposed methodology of run-time exploration of the optimal algorithm as well as the optimal underlying architecture corresponding to multiple system constraints (convergence delay constraint, power, area etc.) as well as to the time-varying parameters of a mobile wireless channel, can greatly increase the range of doppler frequencies (coherence times) for which the optimal bit-allocation can be performed.

Chapter 6

Conclusion

This chapter gives a collective view of the contributions made in this thesis in perspective of the problematic tackled. Important results and deductions that can be inferred from those results will be mentioned. In addition, directions for the possible continuation of the work will also be outlined, while taking into consideration the trends and future of the telecommunications industry.

The world of telecommunications is evolving at a very high pace, with an increasing number of technologies, services and standards appearing with each passing day. While any effort to converge towards a universal technology of communication seems futile and useless, the problematic of anywhere, anytime, any-network, any-service, any-terminal etc. etc. access is of huge interest for both military and the commercial industry. Convergence, Inter-operability, Adaptation and Co-operation has been found to be the answer to this problematic. From IP-based 4G networks to multi-mode cognitive radios, from newly evolved paradigm of Co-operative networks to Spectrum sharing, all are the components of the adaptive and co-operative telecommunications future towards which the world is heading.

The goal of our thesis is to target the complexity issues of the adaptive algorithms in state-of-the-art communication systems. This complexity issue was not only targeted from an algorithmic/theoretical point of view, leading to the complexity reduction of some of the algorithms, but also an effort was made to establish a quantified trade-off link between the performance and the complexity i.e. between the performance parameters of an adaptive wireless communication system to the precise complexity issues of an adaptive algorithm. While a system can be mode adaptive with respect to a large number of its parameters, instead of targeting any specific parameter of adaptation, the adaptation problem was optimized with respect to discrete modulation size (bits), to the continuous power variable as well as with respect to the channel coding gain of a system and different contributions were made in each case, as mentioned below.

In chapter 2, our goal was to reduce the complexity of the bit-loading algorithms. A large number of practical algorithms exist, some of them even applied in wireline (DSL) systems, but for their usage in wireless systems, the complexity needs to be further reduced. We have proposed an algorithm, which to our knowledge converges towards the optimal with the least complexity involved than any other existing practical bit-loading algorithm. By making use of the Gap Approximation, the algorithm is based on the classification of all the subcarriers into subgroups of 3-dB with respect to their channel to noise ratios (SNR) or the corresponding bit-incremental powers. It was discovered that an inherent pattern of allocation is present underneath the classical optimal Hughes-Hartogs allocation procedure. This underlying pattern was brought into use for devising the 3-dB subgroup based allocation algorithm.

The algorithm complexity was not only compared with the classical optimal Hughes-Hartogs solution but also to that of a recently proposed discrete bit-loading algorithm by Papandreou et al. [15]. Our proposed algorithm was found distinctively less complex than the rest of the algorithms. Theoretical complexity analysis results were verified by comparing the running-time of the three algorithms in terms of the actual number of execution-cycles over a Superscalar processor by means of SimpleScalar tool, confirming the reduced complexity of our algorithm.

In chapter 3, our goal was to target the power/energy optimization algorithm. In this regard, instead of the classical capacity maximizing problem, we tackled the more practical problem of optimal energy distribution such that the aggregate BER is minimized. Increasing constraints on power-spectral masks are being posed by the regulatory authorities, dictating the amount of maximum energy/power allowed to be transmitted at a particular frequency. Hence, alongwith the constraints on total available energy, the constraint on peak-energy per subcarrier was also included in the optimization problem.

Theoretical developments for BER-optimized energy allocation taking into account the peak-energy constraint were made and finally a computationally efficient algorithm for peak-energy constrained energy-allocation was proposed, that converges faster towards the optimal as compared to other algorithms. The algorithm complexity was compared with that of the *Iterative Waterfilling* solution, which has been proposed in the literature as an alternate solution. Complexity of the two algorithms was compared by means of the theoretical complexity analysis as well as by comparing the simulation-time of the two algorithms for different parameters and the reduced complexity of our algorithm was demonstrated by means of both of them.

In Chapter 4, our goal was to target the complexity issues of the adaptive channel coding techniques. This chapter is unique to the previous two in the respect that while a large number of works and optimization techniques exist for the previous cases, very few works exist in the literature for the optimization of adaptive coding, particularly for our choice of Irregular LDPC codes, which have gained a lot of popularity in recent years because of their superior performance over other competitive coding techniques.

In this regard, it was observed that conventionally the optimization problematic of irregular LDPC codes has been limited to asymptotic constraints with the assumptions on infinite code-length and infinite independent decoding iterations, these techniques are not suitable for the design of finite-length irregular Codes. At the same time, complex optimization techniques conventionally employed for the optimization of the irregularity profile for a given channel renders difficulty to optimize the irregularity-profile for the cases where it needs to be re-optimized a number of times e.g. where the channel is time-varying. Therefore, we tried to address these problems in chapter 4, where a Greedy Irregularity Construction (GIC) algorithm was proposed as a technique to progressively construct the irregularity profile for finite-length codes. This algorithm is based on the Quantization of the well-known 'Wave-Effect' phenomenon, that exists in the Irregular codes.

The quantization methodology relies on the Majority-Based (MB) hard-decoding probability of error analysis for irregular codes. Where the classical calculation methods for the irregular MB hard-decoding analysis are computationally prohibitive to be used, we proposed a Sum-of-Product-of-Combinations (SPC) based analysis methodology which has shown to reduce the simulation times by a large factor thereby making the Wave-Quantization feasible for implementation. Since, to our knowledge this is the first attempt to perform the 'quantification' of the wave-effect, we believe that a continuation of this work for multiple decoding methods and multiple channels can result in producing yet more significant results.

Finally, in chapter 5, our goal was to link the complexity issues of an adaptive algorithm to the eventual performance of the system thereby making the connection where the performance constraints dictate the usage of architectural resources or vice vera. It has been found, that the delay associated with the running-time of the resource-allocation algorithms incurs a strong impact on the performance of the system. While conventionally only algorithm level modifications are proposed for the complexity reduction of the link-adaptation algorithms , the novel paradigm of Run-Time-Reconfiguration (RTR) opens new possibilities of reducing the computational delay of the link-adaptation algorithms by exploiting the architectural flexibility of the underlying platform.

Hence, in chapter 5, we proposed a methodology for the optimal selection of the processor resources/parameters (ALUs, cache etc.) on which the link-adaptation algorithm is being run, to adapt to the time-varying coherence-time needs of the channel.It was observed that a run-time tuning of the processor parameters results in the usage of link-adaptation algorithm at higher doppler frequencies/user mobility speeds. The usage of Genetic Algorithm helps avoid the large simulation times that would be required otherwise for evaluating the complete design space of a processor architecture

The contributions made in this thesis can be taken along in a number of interesting research directions. As mentioned in chapter 2, the 3-dB subgroup based bit-loading algorithm is inherently adapted to the performance to the complexity tradeoff based upon how large or small the subgroups are made. This, paradigm can be exploited in a number of channels, to quantify the exact improvements. The *Iterative Surplus Re-distribution* (ISR) algorithm that we proposed in chapter 3 can be modified to take into account the case with multiple peak-energy-constraints, as the case with many emerging wireless systems. From chapter 4, our proposed methodology of *Wave-Quantization* can be exploited for a variety of channels and variety of code-lengths to see the different improvements in performance. From chapter 5, the Genetic-Algorithm based methodology for the optimal selection of the processor architecture parameters based on the *speed* of the algorithm can be extended to the Multi-Objective case, where consumed energy is also taken into account alongwith the execution-time of the algorithm. The optimal combination of the different adaptable parameters (bits, power, coding rate) can also be found, in a scenario where they all are variables at the same time. Finally, the different levels of improvements caused by the adaptation of each the parameters, can be quantified by linking it with the amount of architecture resources consumed, thereby proposing a performance-to-complexity based evaluation *unit* so as to establish a universal criterion for accurately judging the complexity improvements of an algorithm, the absence of which today is resulting in a significant loss of resources.

Bibliography

[1] J. F. Hayes, "Adaptive feedback communications," *IEEE Trans. on Communications Technology*, pp. 29–34, Feb 1968.

[2] J. K. Cavers, "Variable-rate transmission for rayleigh fading channels," *IEEE Trans. on Communications*, pp. 15–22, Feb 1972.

[3] W. T. Webb and R. Steele, "Variable rate qam for mobile radio," *IEEE Trans. on Communications*, pp. 2223–2230, July 1995.

[4] M. Alouini and A. Goldsmith, "Adaptive m-qam modulation over nakagami fading channels," in *Proc. IEEE Global Communications Conf. - Communication Theory MiniConf.*, Nov. , 1997, pp. 218–223.

[5] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Mag.*, vol. 9, May, 1990.

[6] P. S. Chow, J. C. Tu, and J. M. Cioffi, "Performance evaluation of a multichannel transceiver system for adsl and vhdsl services," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 6, pp. 909–919, 1991.

[7] J. A. Bingham, *ADSL, VDSL, and Multicarrier Modulation: Wiley Series in Telecommunications and Signal Processing.* New York, NY, USA: John Wiley & Sons, Inc., 2000.

[8] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding digital subscriber line technology*, 1999.

[9] R. G. Gallager, "Information theory and reliable communication," *Newyork, Wiley*, 1968.

[10] D. Hughes-Hartogs, "Ensemble modem structure for imperfect transmission media," *U.S Patent No. 4,833,796*, May 1989.

[11] J.Campello, "Practical bit loading for dmt," in *IEEE International Conference on Communications*, 1999, pp. 801–805.

[12] L. Piazzo, "Fast optimal bit-loading algorithm for adaptive ofdm systems," in *Technical Report No. 002-04-03 Univ of Rome*, September 2003.

[13] B. S. Krongold, K. Ramchandran, and D. L. Jones, "Computationally efficient optimal power allocation algorithm for multicarrier communication systems," in *IEEE Transactions on Communications*, vol. 48, January 2000, pp. 23–27.

[14] R. V. Sonalkar and R. R. Shively, "An efficient bit-loading algorithm for dmt applications," *IEEE Communication Letters*, vol. 4, pp. 80–82, March 2000.

[15] N. Papandreou and T. Antonakopoulos, "A new computationally efficient discrete bit-loading algorithm for dmt applications," vol. 53, May 2005.

[16] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "A practical discrete multi-tone transceiver loading algorithm for data transmission over spectrally shaped channels," in *IEEE Transactions on Communications*, vol. 43, April 1995, pp. 773–75.

[17] A. Czylwik, "Adaptive ofdm for wideband radio channels," in *Proceedings IEEE Globecom*, 1996, pp. 713–718.

[18] R. F. H. Fischer and J. B. Huber, "A new loading algorithm for discrete multi-tone transmission," in *Proceedings IEEE Globecom*, 1996, pp. 724–728.

[19] J. M. Cioffi, "A multicarrier primer," in *ANSI T1E1.4*, November 1991.

[20] E. Baccarelli, A. Fasano, and M. Biagi, "Novel efficient bit-loading algorithms for peak-energy-limited adsl-type multicarrier systems," vol. 50, pp. 1237–1247, May 2002.

[21] A.Mahmood and E. Jaffrot, "An efficient methodology for optimal discrete bit-loading with spectral mask constraints," in *Proceedings IEEE Globecom*, 2006.

[22] A. Mahmood and J. C. Belfiore, "Improved 3-db subgroup based bit-loading algorithm for multicarrier systems," in *Proc. IEEE Sarnoff Symposium, Princeton, US*, April, 2008.

[23] J. R. F. et.al., "Ieee 802.15 channel modeling sub-committee final report," *IEEE P802.15-02/490r1-SG3a*, February 2003.

[24] V. L. L. Goldfeld and D. Wulich, "Minimum ber power loading for ofdm in fading channel," *IEEE Transactions on Communications*, vol. 50, no. 11, pp. 1729–133, November 2002.

[25] C. S. Park and K. B. Lee, "Transmit power allocation for ber performance improvement in multicarrier systems," vol. 52, no. 10, October 2004.

[26] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering.* New York: Wiley, 1965.

[27] I. Kalet, "The multitone channel," *IEEE Transactions on Communications*, vol. 37, no. 2, pp. 119–124, February 1989.

[28] S.Boyd and L. Vandenberghe, "Convex optimization," *CUP, Cambridge , UK*, 2004.

[29] J.G.Proakis, *Digital Communications, 4th Edition.* NewYork, US: McGraw-Hill, 2000.

[30] M.-S. Alouini and A. J. Goldsmith, "Adaptive modulation over nakagami fading channels," *Wireless Personal Comm.*, vol. 13, no. 1-2, pp. 119–143, 2000.

[31] B. Vucetic, "An adaptive coding scheme for time varying channels," *IEEE Commun. Mag.*, vol. 9, no. 5, May, 1991.

[32] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch, "Multiuser ofdm with adaptive subcarrier, bit, and power allocation," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 10, pp. 1747–1758, 1999.

[33] M. Huang and S. Dey, "Combined rate and power allocation with link scheduling in wireless data packet relay networks with fading channels," *EURASIP J. Wirel. Commun. Netw.*, vol. 2007, no. 2, pp. 4–4, 2007.

[34] R. G. Gallager, "Low density parity check codes," *IRE Trans. Info. Theory*, vol. IT-8, pp. 21–28, Jan 1962.

[35] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," vol. 2, 1993, pp. 1064–1070 vol.2.

[36] MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, 1999.

[37] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[38] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[39] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity approaching low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 1729–133, February 2001.

[40] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, Oct, 2001.

[41] R. Storn and K.Price, "Differential evolution : A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[42] R. Chung and Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, 2001.

[43] M. D. D.J.C. MacKay, S.T. Wilson, "Comparison of constructions of irregular gallager codes," *IEEE Transactions on Communications*, vol. 47, Oct, 1999.

[44] J. Campello, D. Modha, and S. Rajagopalan, "Designing ldpc codes using bit-filling," *Proc. Int. Conf. Communications (ICC), Helsinki, Finland*, 2001.

[45] X. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[46] J. V. T. Tian, C. Jones and R. Wesel, "Selective avoidance of cycles in irregular ldpc code construction," 2004.

[47] R. Gallager, *Low Density Parity Check Code.* PhD Disseration, Massachusetts Institute of Technology, Cambridge, MA, US, 1960.

[48] M. Sipser and D. A. Spielman, "Expander codes," pp. 1710–1722, 1996.

[49] P. Zarrinkhat and A. H. Banihashemi, "Threshold values and convergence properties of majority-based algorithms for decoding regular low-density parity-check codes," *IEEE Transactions on Communications*, vol. 52, no. 12, pp. 2087–2097, 2004.

[50] J. Rabaey, "Reconfigurable processing: The solution to low-power programmable dsp," in *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 1*, 1997, p. 275.

[51] P. R. S. R. S. D. M. J. Helmschmidt, E. Schuler and R. Bonitz, "Reconfigurable signal processing in wireless terminals," in *DATE-03: Proceedings of the conference on Design, Automation and Test in Europe*, 2003.

[52] J. A. J. Kevin and P. Chau, "Reconfigurable-hardware-based digital signal processing for wireless communications," in *Proc. SPIE : Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, vol. 3162, oct 1997, pp. 529–540.

[53] J. M. III, "Cognitive radio for flexible mobile multimedia communications." *MONET*, vol. 6, no. 5, pp. 435–441, 2001.

[54] H. Arslan and I. Joseph Mitola, "Special issue: Cognitive radio, software-defined radio, and adaptive wireless systems: Guest editorials," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 9, pp. 1033–1035, 2007.

[55] Q. Zhang, A. B. J. Kokkeler, and G. J. M. Smit, "Adaptive ofdm system design for cognitive radio," in *11th International OFDM-Workshop, Hamburg, Germany*, August 2006, pp. 91–95.

[56] G. K. Rauwerda, P. M. Heysters, and G. J. M. Smit, “Mapping wireless communication algorithms onto a reconfigurable architecture,” *J. Supercomput.*, vol. 30, no. 3, pp. 263–282, 2004.

[57] S. R. J. Veilleux, P. Fortier, “An fpga implementation of an ofdm adaptive modulation system,” in *IEEE-NEWCAS Conference, 2005*, 2005.

[58] M. Cudnoch, A. M. Wyglinski, and F. Labeau, “Dsp implementation of a bit loading algorithm for adaptive wireless multicarrier transceivers,” *Wirel. Commun. Mob. Comput.*, vol. 7, no. 9, pp. 1117–1128, 2007.

[59] S. H. Chitti and G. Kulkarni, “Dynamic reconfiguration of link adaptation algorithms for a multi carrier system,” *Masters Thesis, Aalborg University*, June 2007.

[60] Y. W. N. M. B.A. Jati, C.F. Mayorga, “Exploiting channel dynamics for bit and power allocation,” *Masters Thesis, Aalborg University*, Spring, 2006.

[61] D. C. Burger and T. M. Austin, “The simplescalar tool set, version 2.0, Tech. Rep. Technical Report CS-TR-1997-1342, 1997.

[62] A. R. Bahai and B. R. Saltzberg, *Multi-Carrier Digital Communications: Theory and Applications of Ofdm.* Plenum Publishing Co., 1999.

[63] T. Keller and L. Hanzo, “Adaptive multicarrier modulation: A convenient framewrok for time-frequency processing in wireless communications,” *Proceedings of the IEEE*, vol. 88, pp. 611–640, May 2000.

[64] B. Fox, “Discrete optimization via marginal analysis,” *Management Science, 13*, November, 1966.

[65] R. V. Sonalkar, “Bit- and power-allocation algorithm for symmetric operation of dmt-based dsl modems,” *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 902–906, June 2002.

[66] C. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[67] J. C. Campello, *Discrete Bit-Loading for Multicarrier Modulation Systems.* PhD Disseration, Stanford Univ., CA., 1999.

[68] S. Nader-Esfahani and M. Afrasiabi, "Simple bit loading algorithm for ofdm-based systems," *IET Communications*, vol. 1, no. 3, pp. 312–316, June 2007.

[69] J. Lee, *Power Allocation for Multi-user Multi-carrier Communication Systems.* PhD Disseration, Stanford Univ., CA., March, 2003.

[70] P. Bansal and A. Brzezinski, "Adaptive loading in mimo/ofdm systems," in *Tech. Report, http://web.mit.edu/brzezin/www/359/359.pdf*, December 2001.

[71] D. Bartolome and A. I. Pérez-Neira, "Practical implementation of bit loading schemes for multiantenna multiuser wireless ofdm systems." *IEEE Transactions on Communications*, vol. 55, no. 8, pp. 1577–1587, 2007.

[72] M. Cudnoch, A. M. Wyglinski, and F. Labeau, "Dsp implementation of a bit loading algorithm for adaptive wireless multicarrier transceivers," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 9, pp. 1117–1128, 2007.

[73] Y. Li and W. E. Ryan, "Mutual-information-based adaptive bit-loading algorithms for ldpc-coded ofdm," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1670–1680, 2007.

[74] N. Papandreou and T. Antonakopoulos, "Bit and power allocation in constrained multicarrier systems: The single-user case," *EURASIP Journal on Advances in Signal Processing*, 2008.

[75] K. C. W. Choi and J. Cioffi, "Adaptive modulation with limited peak power for fading channels," vol. 3, 2000, pp. 2568–2572 vol.3.

[76] A. G. Armada, "Snr gap approximation for m-psk-based bit loading," vol. 5, January 2006.

[77] A. Batra, J. Balakrishnan, A. Dabakand, R. Aiello, and J. Foerester, "Design of a multiband ofdm system for realistic uwb channel environments," *IEEE Transactions on Microwave theory and techniques*, vol. 52, pp. 2123–2138, 2004.

[78] R. Saleh, A. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 119–124, February 1987.

[79] L. L. C. Snow. and R. Schober, "Enhancing multiband ofdm performance: capacity-approaching codes and bit loading," 2005.

[80] D. A. Patterson and J. L. Hennessy, *Computer organization and design (2nd ed.): the hardware/software interface.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[81] G. D. B. A. Fasano, "The duality between margin maximization and rate maximization discrete loading problems," 2004.

[82] T. J. Willink and P. H. Wittke, "Optimization and performance evaluation of multicarrier transmission." *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 426–440, 1997.

[83] P. Ligdas and N. Farvardin, "Optimizing the transmit power for slow fading channels," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 565–576, 2000.

[84] A. Goldsmith, *Design and Performance of High Speed Communication Systems over Time-Varying Radio Channels.* PhD Disseration, Univ. of California Berkeley, US, 1994.

[85] M. Alouini, *Adaptive and Diversity Techniques for Wireless digital communications over fading channels.* PhD Disseration, Caltech Institute, US, 1998.

[86] N. Wang and S. D. Blostein, "Approximate minimum ber power allocation for mimo spatial multiplexing systems," *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 180–187, 2007.

[87] D. P. Palomar and J. R. Fonollosa, "Practical algorithms for a family of waterfilling solutions." *IEEE Transactions on Signal Processing*, vol. 53, no. 2-1, pp. 686–695, 2005.

[88] E. Cianca, A. D. Luise, M. Ruggieri, and R. Prasad, "Channel-adaptive techniques in wireless communications: an overview," *Wireless Communications and Mobile Computing*, vol. 2, no. 8, pp. 799–813, 2002.

[89] V. Mannoni, D. Declercq, and G. Gelle, "Optimized irregular low-density parity-check codes for multicarrier modulations over frequency-selective channels," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 1546–1556, 2004.

[90] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003. [Online]. Available: http://www.cambridge.org/0521642981

[91] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Feb 1981.

[92] L. N. M. Dempster, A. P. and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[93] D. Haley, A. Grant, and J. Buetefuer, "Iterative encoding of low-density parity-check codes," in *Proc. IEEE Globecom.*

[94] H. Fujita and K. Sakaniwa, "Some classes of quasi-cyclic ldpc codes: Properties and efficient encoding method."

[95] Y. Kaji, "Encoding ldpc codes using the triangular factorization," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E89-A, no. 10, pp. 2510–2518, 2006.

[96] V. Mannoni, D. Declercq, and G. Gelle, "Low-density parity-check codes," in *Proc. 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002.*, vol. 1, September 2002, pp. 222–226.

[97] J. Hou, P. H. Siegel, and L. B. Milstein, "Performance analysis and code optimization of low density parity check codes on rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 924–934, 2001.

[98] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Transactions on Information Theory*, vol. 48, pp. 3017–3028, 2000.

[99] S. M. A. Thangaraj, "Thresholds and scheduling for ldpc-coded partial response channels," *IEEE Transactions on Magnetics*, vol. 38, no. 5, pp. 2307–2309, September 2002.

[100] C. M. A. Kavcic and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1636–1652, July 2003.

[101] A. deBaynast and D. Declercq, "Gallager codes for multiple access," in *ISIT-02*, Lausanne, Switzerland, July 2002.

[102] P. Berlin and D. Tuninetti, "Ldpc codes for fading gaussian broadcast channels," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 2173–2182, 2005.

[103] R. N. K.R. Narayanan, I. Altunbas, "On the design of ldpc codes for msk," in *IEEE Globecom Conference*, 2001.

[104] X. W. B. Lu, G. Yue, "Performance analysis and design optimization of ldpc-coded mimo ofdm systems," vol. 52, no. 2, pp. 348–361, February 2004.

[105] J. C. W. Tan, "Array codes for erasure correction in magnetic recording channels," *IEEE Transactions on Magnetics*, vol. 39, pp. 2579–2581, 2003.

[106] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.

[107] B. H. B. Ammar, "Construction of low density parity check codes: Bibd and vandermonde," 2004.

[108] J. Campello and D. Modha, "Extended bit-filling and ldpc code design," *IEEE Global Telecommunications Conference, 2001*, vol. 2, pp. 985–989, 2001.

[109] Y. Mao and A. Banihashemi, "A heuristic search for good LDPC codes at short block lengths," in *IEEE International Conference on Communications*, June 2001.

[110] L. Bazzi, T. J. Richardson, and R. L. Urbanke, "Exact thresholds and optimal codes for the binary-symmetric channel and gallager's decoding algorithm a," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2010–2021, 2004.

[111] P. Zarrinkhat and A. H. Banihashemi, "Hybrid hard-decision iterative decoding of irregular low-density parity-check codes." *IEEE Transactions on Communications*, vol. 55, no. 2, pp. 292–302, 2007.

[112] Y. Inaba and T. Ohtsuki, "Bootstrapped modified weighted bit flipping decoding of low density parity check codes," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E89-A, no. 4, pp. 1145–1149, 2006.

[113] Y. X. H. Xiang, "Performance analysis of finite-length ldpc codes," 2003.

[114] R. Tessier and W. Burleson, "Reconfigurable computing for digital signal processing: A survey," *Journal of VLSI Signal Processing*, vol. 28, no. 1, pp. 7–27, June 2001.

[115] T. Report, "Iternational technology roadmap for semiconductors," in *www.itrs.net*, 2007.

[116] P. Project, "Ultra-low-power reconfigurable computing," *http://bwrc.eecs.berkeley.edu/Research/Configurable-Architectures.*

[117] G. X. M. S. C. Ebeling, C. Fisher and H. Liu, "Implementing an ofdm receiver on the rapid reconfigurable architecture," *IEEE Trans. Computers*, vol. 53, no. 11, pp. 1436–1448, 2004.

[118] A. D. F. D.-D.-D. J. R. N. S. W. Luk, P. Andreou and D. Siganos, "A reconfigurable engine for real-time video processing," in *Field-Programmable Logic: From FPGAs to Computing Paradigm.* LNCS 1482, Springer-Verlag, Berlin, 1998, pp. 169–178.

[119] G. J. M. S. Lodewijk T. Smit and J. Hurink, "Run-time adaptation of a reconfigurable mobile umts receiver," in *Proc. ERSA 2004*, 2004.

[120] J. M. Smit and G. K. Rauwerda, "Reconfigurable architectures for adaptable mobile systems," in *Proc. ERSA 2005*, 2005, pp. 17–25.

[121] A. Sloman, "Exploration in design space," in *European Conference on Artificial Intelligence*, 1994, pp. 578–584.

[122] Y. F. H. A. E. Ekpenyong, "Feedback constraints for adaptive systems," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 69–78, May, 2007.

[123] Y. Rong, S. A. Vorobyov, and A. B. Gershman, "Adaptive ofdm techniques with one-bit-per-subcarrier channel-state feedback." *IEEE Transactions on Communications*, vol. 54, no. 11, pp. 1993–2003, 2006.

[124] R. B. Mouhoub and O. Hammami, "Mocdex: multiprocessor on chip multiobjective design space exploration with direct execution," *EURASIP J. Embedded Syst.*, vol. 2006, no. 1, pp. 12–12, 2006.

[125] V. Krishnan and S. Katkoori, “A genetic algorithm for the design space exploration of datapaths during high-level synthesis,” *IEEE Trans. Evolutionary Computation*, vol. 10, no. 3, pp. 213–229, 2006.

[126] M. Holzer, B. Knerr, and M. Rupp, “Design space exploration with evolutionary multi-objective optimisation,” in *2007 Symposium on Industrial Embedded Systems Proceedings*, Lisbon, Portugal, July 2007, pp. 126–133.

[127] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.

[128] T. S. Rappaport and T. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, December 2001.