A Delta Diagram Synthesis for IoT Optimization with Grey Wolf Driven Multi-Objective Automation

by

Prathap Siddavaatam

Master of Science, Brock University, 2011

Bachelor of Engineering, Bangalore University, 2001

A dissertation

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A DISSERTATION

I hereby declare that I am the sole author of this dissertation. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

A Delta Diagram Synthesis for IoT Optimization with Grey Wolf Driven Multi-Objective Automation

Doctor of Philosophy, 2018

Prathap Siddavaatam

Electrical and Computer Engineering

Ryerson University

Abstract

Today, Internet of Things (IoT) is a major paradigm shift that will mark an epoch in communication technology such that every physical object can be connected to the Internet. With the advent of 5G communications, IoT is in urgent need of optimized architectures that can efficiently support wide ranging heterogeneous multi–objective requirements of communication, hardware and security aspects. The optimization challenges are rooted in the technology and how the information is acquired and manipulated by this technology. My research in this thesis provides a description of compelling challenges faced by IoT and how to mitigate these challenges by designing resource–aware communication protocols, resource–constrained device hardware with low computing power and low–powered computational security enhancements. This thesis lays the foundation for optimizing these challenging IoT paradigms by introducing a novel Delta–Diagram based synthesizing model. The Delta–Diagram provides a road–map linking the behavioral and structural domains of a given IoT

paradigm to generate respective optimizer domain parameters, which can be utilized by any optimizer framework. The fundamental part of the communication synthesizer is a mathematical model, developed to obtain the best possible routing paths and communication parameters among things. The ultimate aim of the entire synthesis process is to devise a design automation tool for IoT, which exploits the interrelations between different layer functionalities. This thesis also proposes a novel cross–layer Grey wolf optimizer for IoT, which outperforms some of the contemporary optimizer algorithms such as Particle Swarm, Genetic Algorithm, Differential Evolution optimizers in solving unimodal, multi–modal and composition benchmark problems. The purpose of this optimizer is to accurately capture both the high heterogeneity of the IoT and the impact of the Internet as part of delta diagram synthesis enabled network architecture. In addition, the Grey wolf optimizer for IoT plays a crucial role in design exploration of system on chip architecture for IoT device hardware. The results generated by the optimizer yielded the most optimum feasible solutions in the design space exploration process of the IoT.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Professor Reza Sedaghat for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me throughout the time of research and writing of this dissertation.

My special thanks goes to members of my dissertation examination committee for their valuable feedback.

I owe my deepest gratitude towards my wife Richa for her eternal support and understanding of my goals and aspirations. Her infallible love and support has always been my strength. Her patience and sacrifice will remain my inspiration throughout my life. Without her help, I would not have been able to complete much of what I have done and become who I am. Also, I am thankful to my son Aarav and daughter Reeva for giving me happiness during the course of my studies and being supportive. Without their emotional support in my life, I would have never been able to dedicate the time and effort for pursuing this research and achieving the requisite goals. I would like to thank my mother Meena for her everlasting support and constant encouragement.

Lastly, and most importantly, I wish to thank my late grandmother Jayalakshmamma. She raised me, supported me, taught me about life and loved me. Her words of inspiration and encouragement still lingers in me today. I dedicate this Ph.D dissertation to her.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Internet of Things (IoT) is an rapidly emerging concept in the today's world. It encompasses various objects and communication methods for exchanging data. In practice, IoT is a generalized term that describes a vision where everything should be connected to the Internet. IoT is a ground breaking technology for present and future since the concept opens up seamless opportunities for new services and innovations. IoT is revolutionizing many sectors such as logistics and transportation, manufacturing plants and agriculture by increasing their operational efficiency, reducing the energy usage and better management of resources. As IoT expands with unique opportunities, Cisco estimates that there will be around 50 billion IoT devices by 2020[2]. The growth of IoT devices depicted in Fig. 1.1 since its inception in 2009 is phenomenal. As the number of connected devices increases exponentially, achieving higher network capacity and reliability with minimal latency and energy consumption is highly challenging. It is estimated that the IoT will cause a substantial increase in the Internet Protocol (IP) traffic to the tune of 300% by year 2020[4].

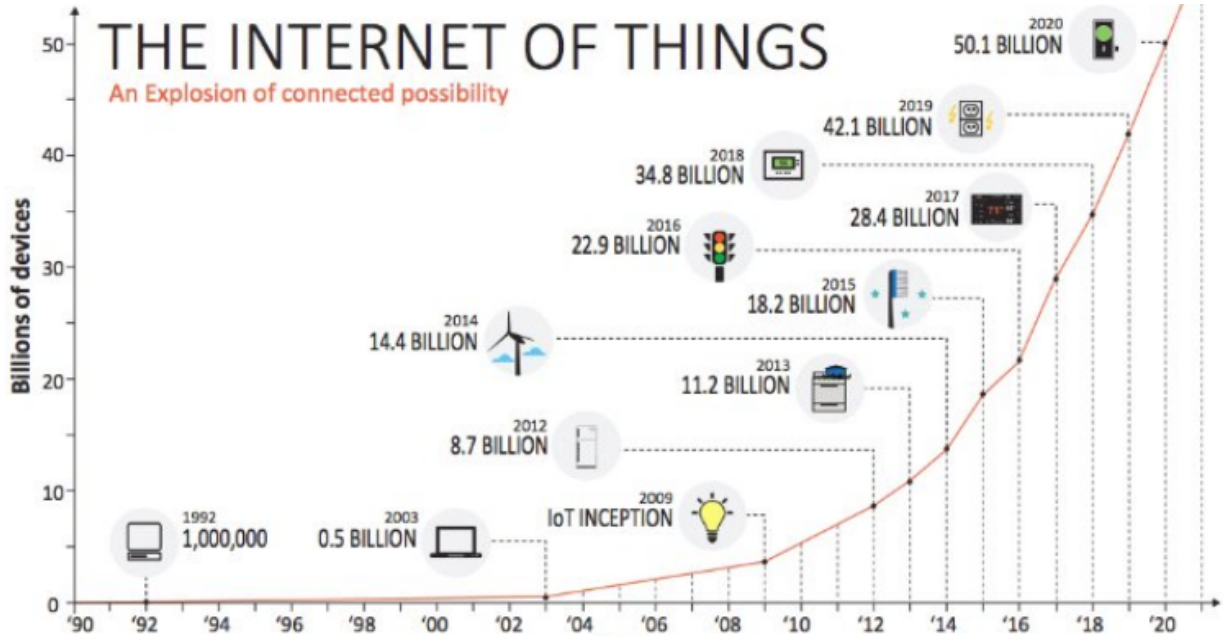


Figure 1.1: Predicted explosive growth of IoT [2]

1.1 IoT

Defining the term IoT can be cumbersome since its definitions are usually ambiguous since it is loosely used as a buzzword in scientific research and marketing strategies.

The basic notion of IoT is to link things together which allows them to communicate with each other thereby enabling people to share information [5]. "Things", in the IoT sense, can mean a wide variety of devices depending on the context of term used. However, in general terms legal scholars suggest regarding "things" as an "inextricable mixture of hardware, software, data and service" [6]. In this thesis we follow the definition of IoT proposed by ITU-T's Telecommunication Standardization Sector (a United Nations agency which specializes in Information and Communication Technology): "*A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving inter-operable information and communication technologies*".

1.2 Interconnection Challenge among Devices

The three main tasks of any IoT network are sensing, analyzing and transmitting. Because of the communication capability, thousands of individual nodes form a single entity and provides collective intelligence on the topic of interest. Due to the limited transmission range, IoT devices may have to multi- hop their sensed data possibly over long distances. Therefore, connectivity or communication between individual devices is crucial for the successful operation of the network. Main factors that plague the reliability of the communication are discussed in the following sub-sections.

1.2.1 Interference: Internal and External

An inherent problem associated with the wireless networks is interference. Interference can be internal or external. External interference is caused by other sources such as domestic appliances, Bluetooth devices. Internal interference occurs when the nodes within the same network transmit at the same time.

Wireless Sensor Networks (WSN) operate on an unlicensed ISM bands and therefore the radio spectrum is shared with several other devices[7]. As Wi-Fi uses 2.4GHz frequency, the WSN such as IoT might have to compete to utilize the frequency. Appliances such as microwave oven, plasma lighting system, cordless phones that generate electromagnetic noise are also known to increase the packet loss rate, this, in turn, increases the retransmission rates and latency[7, 8]. Similarly, internal interference due to concurrent transmission also negatively impacts the performance of the network[9]. It is also empirically shown that a cross-channel interference also decreases the packet delivery rate[10].

1.2.2 Environmental Obstacles

Natural obstacles such as office walls, human and vehicular movements, dense vegetation generally deteriorate the signal strength. Humidity and temperature also influences the

radio waves[11, 12, 13, 14].

1.2.3 Deployment Environments

When the area to be monitored is large and in a remote location, sensors are usually randomly deployed. The position of the sink node a data aggregation point also plays a crucial role in the performance of the network. Imbalanced traffic load causes some of the nodes in the network to expend more energy, risking the network to become unconnected and thus rendering it useless[15].

If the average number of hops between the source and the sink node is less, the latency and the energy consumption is also relatively lesser compared to the larger number of hops between source and the sink. In addition, if the number of hops between the source and the sink is less, the collision rate is less which in turn results in better packet delivery rate.

1.2.4 Hardware and Software Configuration

Application data rate, the amount of control messages generated by the routing protocol at the network layer, collision avoidance technique employed at the MAC layer greatly influences the performance of the network. Choosing the routing protocol such as Collection Tree Protocol (CTP) that generates the fairly large amount of control packets along with high application data rate increases the contention in the network[16].

Even when the transceivers of the sensor nodes in an IoT network are configured exactly, in the same way, they may distort transmitted or received a signal due to their internal noise[17, 18]. The low power transmits signals are more susceptible to interference and the multi-path distortion. In addition, the remaining battery life is also known to affect the sensitivity of the transceivers[19]. Furthermore, most of the IoT devices have an in-built antenna with irregular radiation pattern. All these factors contribute to the degradation of the communication quality.

1.3 Motivation

Multi-objective optimization techniques applied for communication system synthesis have the attractive advantage of explicitly capturing the constraints and preferences of the communicating device that use both small scale as well as large scale networks, and provide unique insight into trade-offs among competing requirements. The next-generation communication systems aim to build an intelligent and efficient transmission by connecting a variety of heterogeneous networks to fulfill information exchange tasks. The motivation (MO) for this thesis is to address various challenging aspects of communication systems such as: (a) the development of new efficient protocols and optimization of existing communicating devices (b) the hardware design improvement of low power embedded devices for network architectures and (c) the security analysis and privacy improvement of low computing powered systems.

MO1: High-reliability Low-latency factor. As discussed in detail from the previous sections, each day the world embraces more devices to connect everything, everywhere, and everyone. Driven by global digitalization and the emerging IoT or IoE paradigms, the number of connected devices are increasing at an exponential rate from the current circa 10 billion to an estimated 20-fold within the next five years (refer Fig. 1.1). The abundant exploitation of wireless sensors, gadgets, multimedia services, autonomous robots or tactile Internet, augmented reality, and other similar applications, will require unparalleled access rates with high reliability and low latency.

MO2: Competing Resources in 5G. The basic idea of this new type of networking and computing paradigm is the pervasive presence of a variety of objects (things), such as WSN, actuators, mobile devices, and RFID tags, which are able to interact with each other and communicate with the Internet infrastructure. Designing such a scalable network that can incorporate dynamic elements such as device-2-device communications, small cell access points, network cloud, and the IoT which is an integral part of 5G cellular network architecture faces many challenges. These challenges are mainly caused by the limited resources

available, especially reliable security, limited battery lifetime, computing capacity, and memory space at sensor nodes, and also by dynamic network topology especially in ad hoc sensor networks. On the other hand, special features of WSN such as limited computing power, storage space, battery capacity and limited area for hardware sensory circuits, in particular bring in new requirements on time and space efficiency of the MoAs that can be applied in such an environment.

MO3: Scalability of IoT. Many contemporary IoT designs fail to treat the trade-off problems as a MOO issue; instead they choose an objective to optimize, leaving the others as restrictions. This leads to most of them being applied only to networks of limited magnitude, and in very restricted situations. As the scale of commercial systems grows, there is a complete lack of tools to aid in the designing process and the age-old methodology of trial and error is neither effective nor efficient.

MO4: Routing in IoT. Routing strategy is another fairly important problem in IoT design [20, 21], which may have a direct impact on multiple aspects of network performance, including data transmission delay, throughput, security and lifetime of individual nodes as well as the entire network. Mobile agents are often used in IoT to visit a sequence of sensors and fuse impotent data. Optimal agent routes also often need to meet multiple objectives such as minimizing total path delay, loss, and energy consumption as well [22]. These MOO trade-off problems are challenging issued that need to be addressed.

MO5: IoT Hardware Optimization. To devise a design space exploration (DSE) methodology to perform multi–objective optimization (MOO) subject to constraints with a trade–off requirement from the very earliest stage of designing IoT having low computing power hardware. This will enable the IoT hardware device designers to start the development with an architecture that is already specification aware (high level optimized) from the highest level of abstraction thus rendering more chances that final design (logic/layout) corresponds to the given low computational constraints. A stellar synthesis system can produce several designs from a given hardware specification within reasonable amount of time.

However, the final selection can be challenging with many choices of designs produced (near optimal set). Therefore, an efficient exploration method is currently needed for IoT hardware to tackle the problem from a very high abstraction level. This assures the IoT hardware designer a greater chance of optimization and flexibility to control architecture based on user requirements. More importantly, this allows the developer to explore different trade–offs between cost, speed, power etc. or to take an existing design and produce a functionally equivalent one that is more efficient.

1.4 Research Objectives and Goals

The research objective (RO) is to develop an adaptive framework for decomposition of IoT communication system devices and protocols using multi-objective optimization techniques. These techniques are primarily driven by cost-benefit analysis using power-security-execution time and area trade-off circumstances. The results can then be used to assist communicating network interfaces in arriving using optimal solutions that depend on communicating device capabilities. The following research objectives are achieved.

RO1: Develop Novel Cross-layer Schemes for the IoT Network. The use of cross-layer communication schemes to provide adaptive solutions for the IoT is motivated by the high heterogeneity in the hardware capabilities and the communication requirements among things. In this thesis, a novel Delta Diagram synthesis for the IoT is proposed to accurately capture both the high heterogeneity of the IoT and the impact of the Internet as part of the network architecture.

RO2: Develop a Novel Optimizer Framework based on Grey Wolf Optimizer Algorithm for IoT domain. Design an effective resource management meta–heuristic decision maker using a novel Grey Wolf Optimizer (GWO) framework is proposed to obtain optimal routing paths and the communication parameters among things, by exploiting the interrelations among different layer functionalities in the IoT. Moreover, a cross-layer

communication protocol is utilized to implement and test this optimization framework in practical scenarios. The results show that the proposed solution can find near-optimal solutions constructively and outperforms existing layered solutions. The novel Delta-diagram is a preliminary step towards providing efficient and reliable end-to-end communication in the IoT which may be extended to other dimensions of IoT like security and hardware synthesis.

RO3: Develop and demonstrate a Novel design space exploration scheme for IoT hardware architecture synthesis. To introduce a novel design space exploration (DSE) approach which is based on combination of the Utility Ranking (UR) method and Grey wolf Search based framework that is rapid and accurate in IoT hardware architecture evaluation and selection. The goal for the novel DSE is to apply on a number of standard benchmarks which yielded superior results compared to the current existing DSE approaches [23, 24, 25] for architecture selection. The comparison results of our devised novel approach with the current existing approaches for different benchmarks will be shown and the speed boosts obtained will be presented.

RO4: Designing a universal synthesizer for automating communication protocol optimization A novel synthesizer based on Delta diagram approach is developed and can utilized universally for plotting parameters for any layer in the communication protocol stack and subsequently utilized for model optimization strategies for the network. This approach leads to development of more customizable system designs since designers using our framework have the flexibility to choose the most optimized communication channel to meet their trade-off objectives.

RO5: Designing a new Modulo Addition for enhancing security of light weight ciphers in IoT We introduce a new Modulo Addition structure that includes components such as Input Expansion, Modulo Addition, and Output Compaction. The security of the new structure is scalable and completely user-defined. It also substantially increases the algebraic degree of the existing cryptographic system and thwarts the probabilistic conditions ideal for hacking attacks. We develop an adaptive and extensible Addition Modulo

for IoT based ultra lightweight ciphers which includes both stream and block ciphers, that not only attains optimum security but also its design results in less gate–equivalents (GE's), less memory size, and less power dissipation compared to complete cipher upgrade. This approach is crucial in bringing online certain things with limited security architectures. These things can be older versions of small–scale embedded devices with restricted hardware and cannot be outright replaced but can incorporate certain security upgrades like our proposed model.

1.5 Contributions

The fundamental contribution and novelty of this work is to develop innovative optimizer schemes to address three main verticals of IoT domain such as communication system, device hardware and security ciphers and provide a novel seamless solutions to cater these pressing issues in IoT especially with the advent of 5G technologies. The research objectives are realized by employing the following contributions.

Automated Communication Synthesizer. We design and develop a IoT communication framework that utilizes a novel Delta Diagram based model for rapid choosing of network layer parameters to initiate a cross-layer optimization. A fundamental model is developed that caters to the optimization parameters synthesized pertaining to energy, timing, and packet error rates of the network. In addition, the new framework utilizes a Grey wolf optimizer to optimize Quality of Service (QoS) requirements in terms of delay, energy consumption and reliability requirements of an IoT communication network and arrives at near optimal solutions.

Novel DSE methods for synthesizing IoT device hardware. We design and develop a new framework for an accelerated design space exploration of resource constrained IoT hardware. The approach was successful in laying the foundation for exploring the design points from the architecture design space according to the performance objective and in-

tended functionality. This novel method determines the utility coefficients of each resource for designing hardware of IoT. After the architectural design specification were organized in sorted order based on the utility coefficients calculated, the procedure for applying the optimizer algorithm becomes simpler. The research objective is to drastically reduce the number of architectural variants to be analyzed for selection of the system architecture. The proposed mechanism for DSE was able to resolve conflicting objectives in HLS by utilizing a novel Grey Wolf Optimizer. A case study with IoT device based Elliptic Wave filter was presented for further analysis.

Security Enhancement Techniques for IoT device hardware. We devise a new type of scheme for Modulo Addition to defend against algebraic attacks involving multivariate polynomials for lightweight ciphers. This caters to both Stream an Block type in IoT communication nodes. The developed model features three components: an Expandable Input, Modulo Addition, and a compression module called Deflate function. In addition, the new design framework utilizes an expanding and compacting structure that can fit into various lightweight cryptographic systems based on user–based requirements and depends on the constrained computational power of specific IoT nodes.

1.6 Thesis Organization

An outline linking our research objectives in IoT domain to the goals realized in following chapters is shown using the Fig. 1.2. This makes the reader gain better understanding of the aims in this thesis. The organization of this thesis linking the chapters to IoT themed research goals is illustrated using organization chart shown in Table. 1.1. The rubric in Table. 1.1 organizes the chapters as follows: Chapter 1 will describe the motivation and objective behind this thesis. Chapter 2 will outline in detail about current research emphasis on MOO techniques and survey the respective methods. Chapter 3 synthesizes the network of resource deprived IoT communication systems using a novel delta diagram based

Chapters	Research Goals
1.	Defines IoT and uses, Connects our research goals to current industry trends.
2.	IoT objectives achieved and potential future work.
3.	Surveys contemporary Multi-objective optimizers, choose the best suited for our IoT based research.
4.	Utilize a novel approach to choose levelized parameters to initiate cross-layer optimization of IoT network.
5.	Novel optimization framework for initiating design space exploration on resource constrained IoT Hardware.
6.	Novel design for updating IoT based lightweight ciphers sans complete replacement.

Table 1.1: Organization Rubric relating Chapters to Research Goals of this Thesis.

parameter capture method and utilizes a novel Grey Wolf Optimizer to narrow the search space finding near–optimal solutions constructively and compare state-of-the art methods. The research goal described in Chapter 4 formulates High Level Synthesis tools for Design Space Exploration of IoT specific resource constrained devices and also utilizes the novel Grey Wolf Optimizer driven multi-objective optimization algorithm described in Chapter 2 to shortlist near–optimal solutions to get the best hardware IoT configuration. Chapter 5 outlines research based design for security enhancement of IoT based lightweight ciphers on existing devices rather than a complete replacement of the things. In the end, Chapter 6 outlines conclusions drawn from this thesis work and potential future path.

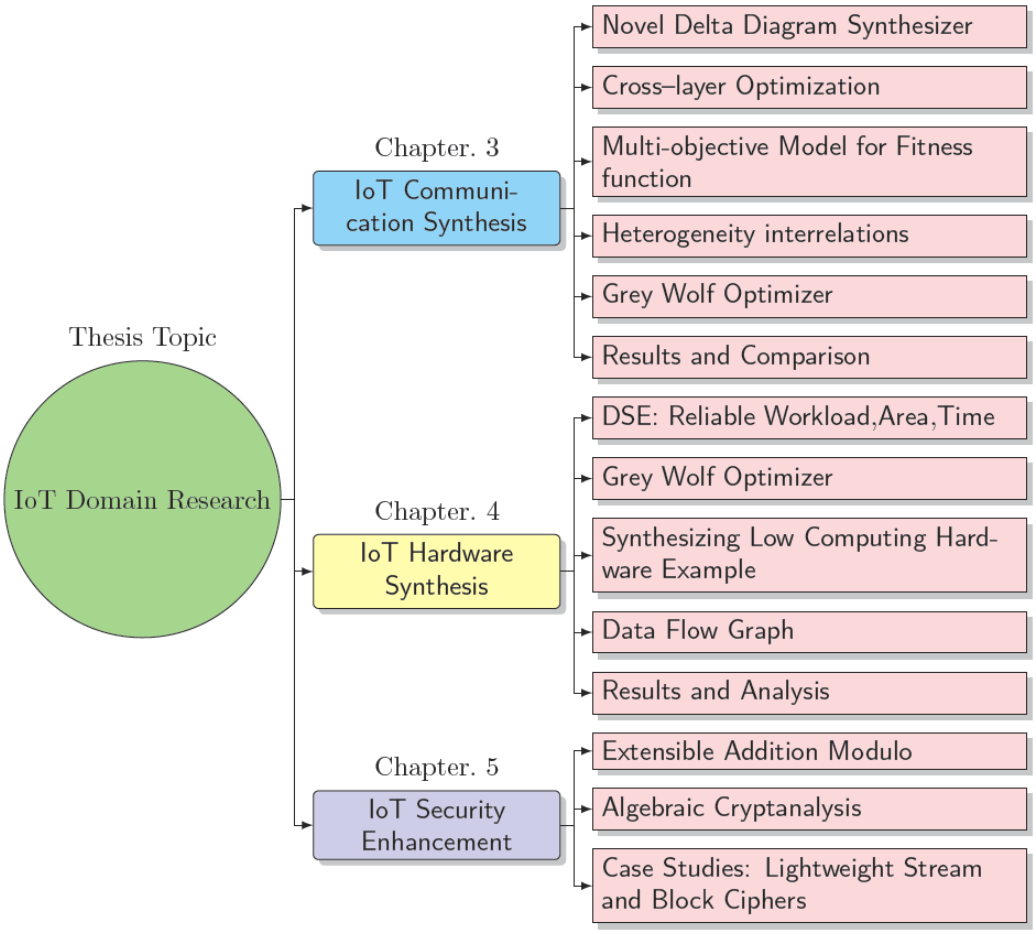


Figure 1.2: Thesis Work Breakdown Structure.

Chapter 2

REVIEW: MULTI–OBJECTIVE OPTIMIZATIONS

In this section, we briefly discuss some important Multi–Objective Optimization (MOO) techniques proposed in the literature that are only relevant while tackling constraints associated with resource-aware IoT domain. It is important to note that this chapter is not a comprehensive literature review of general MOO techniques.

2.1 Strategies for IoT Optimization

Optimization covers almost all aspects of human life and work. In practice, the resources are limited, hence optimization is significant. Most research activities in IoT engineering involve a certain amount of modeling, data analysis, computer simulations and mathematical optimization. This branch of applied science aims for finding the particular values of associated variables, which results in either the minimum or the maximum values of a single objective function or multiple objective functions [26]. A typical optimization process is composed of three components [27]: the model, the optimizer/algorithm and the evaluator/simulator, as

shown in Figure. 2.1. The representation of the physical problem is carried out by using mathematical formulations to establish a mathematical model.

There is no single algorithm that is suitable for all problems arising in IoT domain and an efficient optimizer or algorithm must be designed to ensure that an optimal solution set is obtained. In general, optimization algorithms can be classified as:

1. Finitely terminating algorithms, such as the family of simplex algorithms and their extensions, as well as the family of combinatorial algorithms;

2. Convergent iterative methods that

   - Evaluate Hessians (or approximate Hessians, using finite differences), such as Newtons method and sequential quadratic programming;
   - Evaluate gradients or approximate gradients using finite differences (or even sub-gradients), such as quasi-Newton methods, conjugate gradient methods, interior

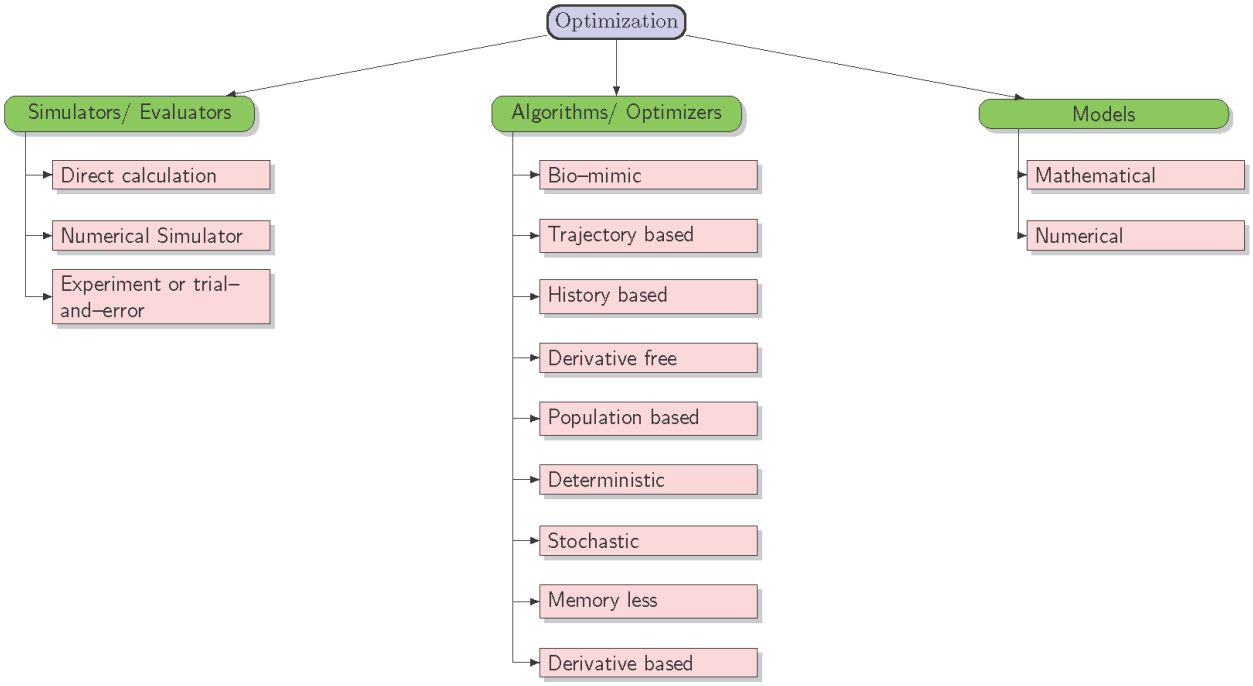


Figure 2.1: A typical schema for Optimization Process

point methods, gradient descent (alternatively, "steepest descent" or "steepest ascent") methods, sub-gradient methods, bundle method of descent, ellipsoid method, reduced gradient method, and simultaneous perturbation based stochastic approximation methods; and

- Evaluate only function values, such as interpolation methods and pattern search methods.

3. Heuristics/Meta-heuristics that can provide approximate solutions to some optimization problems particularly in resource starved IoT.

Recently, bio-mimetic heuristics/meta-heuristics based strategies have been widely used to solve MoA for designing WSN in the IoT [28], since they are capable of obtaining near-optimal solutions to optimization problems characterized by non-differential nonlinear objective functions, which are particularly hard to deal with using classical gradient- or Hessian-based algorithms.

2.2 Multi–Objective Optimization

Multi-objective optimization (MOO) is a process of simultaneously optimizing multiple conflicting goals or objectives. The difference with single-objective optimization is that MOO results in several/many equally efficient solutions, known as Pareto optimal solutions, instead of one single solution. A MOO problem consists of a set of objective functions that can be either maximized or minimized. A set of constraints limits the set of possible outcomes, known as the solution space.

Without loss of generality, a multi-objective minimization problem having n variables and $m(m > 1)$ objectives can be formulated as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \mathcal{F}(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ \text{subject to} \quad & g_j(x) \geq 0, j = 1, 2, \ldots, m_{je} \\ & h_k(x) \geq 0, k = 1, 2, \ldots, m_{ek} \end{aligned} \tag{2.1}$$

where we have $x \in R^n$ with R^n being the decision space, $(g_j(x),\ h_k(x))$ are inequality constraint functions and $\mathcal{F}(x) \in R^m$ with R^m representing the objective space.

2.3 Parto-Optimality

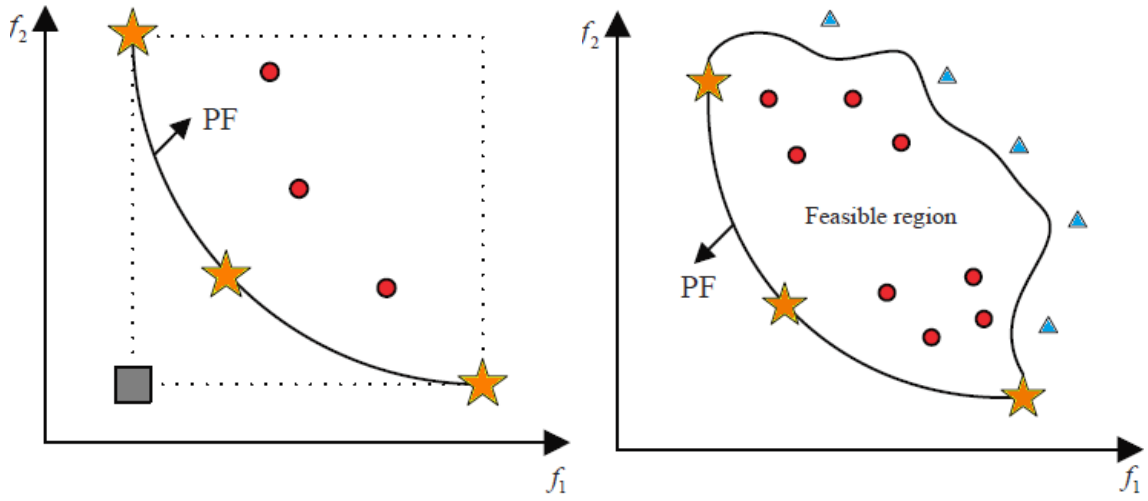
The improvement of one of the objectives in Eq. 2.1 may result in the degradation of other objectives, thus it is important to achieve a *Pareto-optimality*, which represents the conditions when none of the objective functions can be reduced without increasing at least one of the other objective functions [29]. For the minimization of m objectives $\mathcal{F}(x) = (f_1(x), f_2(x), \ldots, f_m(x))$, we have the following definitions.

- *Non-dominated solutions*: A solution **a** is said to dominate a solution **b** if and only if [30]:

  1. $f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \forall\ i \in \{1, 2, , m\}$,
  2. $f_i(\mathbf{a}) < f_i(\mathbf{b}) \exists\ i \in \{1, 2, , m\}$.

  Solutions that dominate the others but do not dominate themselves are termed non-dominated solutions.

- *Local optimality in the Pareto sense*: A solution a is said to be locally optimal in the Pareto sense, if there exists a real $\mathcal{E} > 0$ such that there is no other solution **b**



(a) PF of unconstrained MoP (b) PF of constrained MoP

Figure 2.2: ⋆=Solutions of the objective functions in PF, ●=Feasible solutions and ▲=Infeasible solutions where $m = 2$.

dominating the solution a with $\mathbf{b} \in R^n \cap B(\mathbf{a}, \mathcal{E})$, where $B(\mathbf{a}, \mathcal{E})$ shows a bowl having a center a and a radius $\mathcal{E}$.

- *Global optimality in the Pareto sense*: A solution a is globally optimal in the Pareto sense, if there does not exist any vector $\mathbf{b}$ that dominates the vector $\mathbf{a}$. The main difference between global and local optimality lies in the fact that for global optimality we no longer have a restriction imposed on the decision space R^n.

- *Pareto-optimality*: A feasible solution is said to be Pareto-optimal, when it is not dominated by any other solutions in the feasible space. Pareto sense, which is also often referred to as the efficient set, is the collection of all Pareto-optimal solutions and their corresponding images in the objective space are termed the PF.

We observe from Figure. 2.2a that the Pareto-optimal solutions of the objective functions in the PF (marked as asterisk) provide better values than any other solution in R^m. The ideal solution marked by a square indicates the joint minimum of the objective values f_1 and f_2 and it is often difficult to reach. The remaining solutions marked as solid circles are

all dominated by at least one solution of the PF. In contrast to the unconstrained scenario of Figure. 2.2a, in Figure. 2.2b, the curve illustrates the PF of a constrained MoP. The solid circles in the feasible region represent the feasible solutions, while the remaining points outside the feasible region (e.g. the points marked by triangles) are infeasible [31].

2.4 MOA application for IoT

A variety of algorithms have been developed for solving MoPs in communication networks and hardware systems related to IoT devices. Multi-objective procedures can be broadly classified into two categories – scalarization and evolutionary methods [32]. Scalarization methods apply in mathematically well defined problems with explicit formulations of objectives and constraints, whereas evolutionary strategies mainly apply in black-box problems, where objectives and/or constraints are evaluated for each value of the optimization variable set. Examples of such evolutionary algorithms are Genetic Algorithms and Swarm Intelligence Algorithms. Some state-of-the art examples of Swarm Intelligence Algorithms include Particle Swarm Optimization (PSO), Grey Wolf Optimizer, Whale Optimizer and Artificial Bee Colony (ABC) algorithms [3, 33, 34, 35].

2.4.1 Scalarization Methods

We briefly describe the most prevalent algorithms in this category: Weighted Sum and $\mathcal{E}$-Constraint Methods, and other scalarizing approaches are not relevant to our discussion in this thesis.

A. Weighted Sum Method

The weighted sum method assigns a non-negative weight to each objective and normally the weights sum up to one. The mathematical definition is shown in Equation 2.2. The different objectives do not have to be scaled because the weights merely serve to find solutions on

the Pareto front. By changing the set of weights, a different point on the Pareto front can be obtained. However, there are three difficulties with the weighted sum method [36]: (i) there is no satisfactory (a priori) selection method to determine the weights that guarantee the final solution to be acceptable, (ii) it cannot find solutions on non-convex regions of the Pareto front, and (iii) varying the weights may not result in an evenly distributed and accurate/complete representation of the Pareto front.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \mathcal{F}(x) = \sum_{m=1}^{M} w_m f_m(x) \\ \text{subject to} \quad & g_j(x) \geq 0, j = 1, 2, \ldots, J \\ & h_k(x) \geq 0, k = 1, 2, \ldots, K \end{aligned} \tag{2.2}$$

B. $\mathcal{E}_{\texttt{constraint}}$ Method

The $\mathcal{E}_{\texttt{constraint}}$ Method optimizes one objective, while the other objectives are used as constraints. It overcomes the limitations of the Weighted Sum Method. Consequently, compared with the weighted sum method (Equation 2.2) there is only one objective function ($\mathcal{F}_\mu(x)$) and additional constraints ($f_m(x) \leq \mathcal{E}_{\texttt{m}}$) that require the other objectives do not exceed the user-defined values. The new equation is shown in Equation 2.3.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \mathcal{F}_\mu(x), \\ \text{subject to} \quad & \mathcal{F}_m(x) \leq \mathcal{E}_{\texttt{m}},\ m = 1, 2, \ldots, M \text{ and } m \neq \mu \\ & g_j(x) \geq 0, j = 1, 2, \ldots, J \\ & h_k(x) \geq 0, k = 1, 2, \ldots, K \end{aligned} \tag{2.3}$$

Although the $\mathcal{E}_{\texttt{constraint}}$ method has various advantages over the weighted sum method such as its ability to produce non-extreme efficient solutions, one of the biggest drawback is selecting the value of $\mathcal{E}$. Since the best values are not known beforehand, increasing the $\mathcal{E}$ value with small steps would result in a lot of redundant runs. Increasing with larger steps would

result in it missing the pareto optimal solution while optimizing resource-aware IoT systems.

2.4.2 Knapsack Problem

The classical *Knapsack Problem* (KP) is a variation of the $\mathcal{E}_{\texttt{constraint}}$ method where is objective function is a dual viz., we maximize it instead of minimization. We can set $N := \{1, \ldots, n\}$ of items and a knapsack of limited capacity. To each item we associate a positive profit p_j and a positive weight w_j. The problem calls for selecting the set of items with maximum overall profit among those whose total weight does not exceed the knapsack capacity $c > 0$. KP has the following Integer Linear Programming (ILP) formulation:

$$\underset{x}{\text{maximize}} \quad \mathcal{F}(x) = \sum_{j \in N} p_j x_j \tag{2.4}$$

$$\text{subject to} \quad \sum_{j \in N} w_j x_j \leq c \tag{2.5}$$

$$x_j \in \{0, 1\}, j \in N \tag{2.6}$$

where each binary variable x_j, $j \in N$, is equal to 1 if and only if item j is selected. In general, we cannot take all items because the total weight of the chosen items cannot exceed the knapsack capacity c. In the sequel, without loss of generality, we assume that $\sum_{j \in N} w_j > c$ and $w_j \leq c$ for every $j \in N$. The *k-item Knapsack Problem* (kKP), is a KP in which an upper bound of k is imposed on the number of items that can be selected in a solution. The problem can be formulated as (2.4)-(2.6) with the additional constraint

$$\sum_{j \in N} x_j \leq k, \tag{2.7}$$

with $1 \leq k \leq n$. KP has widely been discussed in the literature and we refer to [37] for a comprehensive illustration of the problem. The KP and its multidimensional version (MKP) are basic problems in combinatorial optimization. We utilize this approach in [38] whose objective was to obtain or approximate the set of efficient Pareto solutions for high level synthesis of IoT hardware.

2.5 Genetic Algorithms

A genetic algorithm (GA) applied to a MOO begins with an initial set of solutions, represented by chromosomes. In each generation, new solutions are generated using genetic operators such as recombination, crossover and mutation. For each individual in the population (in each generation), a fitness value $\mathcal{F}(x)$ is calculated representing the goodness of the solution. The algorithm stops when an end condition is satisfied. Crossover takes two individuals and uses random point(s) to cut the chromosome in two segments, a 'head' and 'tail' segment [39, 40, 41]. The tail segments are swapped over to produce two new chromosomes.

GA suffers from innumerable disadvantages such as no guarantee in finding global minima, impractical time taken for convergence, producing incomprehensible engineering solutions, etc. Consequently the GA method is discarded for dealing with critically resource depraved IoT optimization approaches.

2.6 Swarm Intelligence Algorithms

The various MoAs discussed so far in the previous sections are subject to many disadvantages. The salient drawbacks include impractical time to reach termination condition, egregious time delay to reach near–optimal solutions, complex aspects such as calculating the fitness function which requires inherent tweaks and adjustments at every iteration, incomprehensible solutions produced leading to multiple re-runs, etc. These drawbacks are

in in direct contradiction to optimizing resource-aware and time critical IoT domain. Thus Swarm Intelligence Algorithms seem to be an obvious fit due to their features such as scalability, system robustness, adaptability and flexibility.

Swarm intelligence (SI), an integral part in the field of artificial intelligence, is gradually gaining prominence, as more and more high complexity problems require solutions which may be sub-optimal but yet achievable within a reasonable period of time. Mostly inspired by biological systems, swarm intelligence adopts the collective behavior of an organized group of animals, as they strive to survive. Our primary emphasis is on most popular in literature and state–of–the–art algorithms: PSO and GWO. In essence, we focus on the algorithms inspired by birds and wolves.

2.6.1 Particle Swarm Optimization

PSO is a stochastic optimization algorithm which was derived from the flight patterns of a flock of birds in search for food, overcame majority of the problems that were faced by the algorithms discovered before it.

Two main parameters of the PSO algorithm is finding the most optimal position and the most optimal velocity of the swarm as the birds strive to achieve their personal best position and velocity for themselves as well as the best position and velocity depending upon the group.

The PSO algorithm is similar to GAs because of how randomly ordered the search space is, initially. However, the biggest advantage this algorithm has over GA is that the PSO algorithm has fewer parameters that are needed to be adjusted after every iteration. The change in position and velocity is given by the following equations:

$$x_i\left(t+1\right) = x_i + V_i\left(t+1\right) \tag{2.8}$$

Here x denotes the position of the bird in the current time frame, which depends on the position of the bird in the previous time frame and the current velocity of the bird. Note: In the PSO described below (Algorithm. 1), the initialization step updates the position of the primary particles with velocity factor and hence the step of update function is mentioned after velocity adaptation step. Therefore, for this research work we adopt this variation of PSO. The current velocity or the updated velocity of the bird is found using the following equation:

$$V_i\left(t+1\right) = \omega V_i\left(t\right) + c_1 r_1 \left(x_i^{lb} - x_i\left(t\right)\right) + c_2 r_2 \left(x_i^{gb} - x_i\left(t\right)\right) \tag{2.9}$$

The parameters in the Eq. 2.8 can be summarized as follows:

- ω is the inertia weight
- c_1 and c_2 are the cognitive and the social learning factors respectively.
- r_1 and r_2 are random numbers in the range [0,1].
- x_i^{lb} is the position of the local best particle with respect to the minimization problem, and
- x_i^{gb} is the position of the global best particle.

The entire optimization algorithm aims at achieving x_i^{gb} viz., all the particles try to converge at the best position of one particle, thus reaching a termination point. The pseudo code for PSO described above is outline in Alg. 1. In the process of Systems of Chip (SoC) development, the problem of an exponentially exploding design space can be a major challenge. PSO algorithm can be employed to realize the most optimum resource configuration [42].
In search of the global best solution, the PSO algorithm divides a swarm into multiple sub-swarms and thus, the objectives could only be handled by one sub-swarm. As a result of this, a bigger swarm size is needed to solve the multi-objective optimization problem. Thus in each iteration, the computation time rises [43].

ALGORITHM 1: (PSO) Pseudo Code

Input: Initialize a population of particles with random values positions and velocities from D dimensions in the search space.

while *Termination condition not reached* **do**
  foreach *every particle* i **do**
    Evaluate the fitness $\mathcal{F}(\overrightarrow{X}_i)$
    if $\mathcal{F}(\overrightarrow{X}_i) < \mathcal{F}(\overrightarrow{P}_i)$ **then**
      $\overrightarrow{P}_i \leftarrow \overrightarrow{X}_i$
    end

    if $\mathcal{F}(\overrightarrow{X}_i) < \mathcal{F}(\overrightarrow{P}_g)$ **then**
      $\overrightarrow{P}_g \leftarrow \overrightarrow{X}_i$
    end

    Adapt velocity of the particle using Equation 2.8
    Update the position of the particle using Equation 2.9
  end
end
return $\overrightarrow{X}_i^{gb}$

2.6.2 Grey Wolf Optimizer (GWO)

Grey wolf optimizer (GWO) [3] is one of recent meta-heuristics swarm intelligence methods. It has proved its efficiency for solving capacitated and non-capacitated optimization problems. It has been widely tailored for a wide variety of optimization problems due to its impressive characteristics over other swarm intelligence methods: it has very few parameters, and no derivation information is required in the initial search. Also it is simple, easy to use, flexible and scalable. It has a special capability to strike the right balance between the exploration and exploitation during the search process which can adapt quickly to different search space terrains and avoid premature convergence. The GWO has recently gained a very big research interest with tremendous audiences from several domains in a very short time [44].

Grey wolfs (*Canis lupus*) are apex predators belonging to the Canidae family[45], which

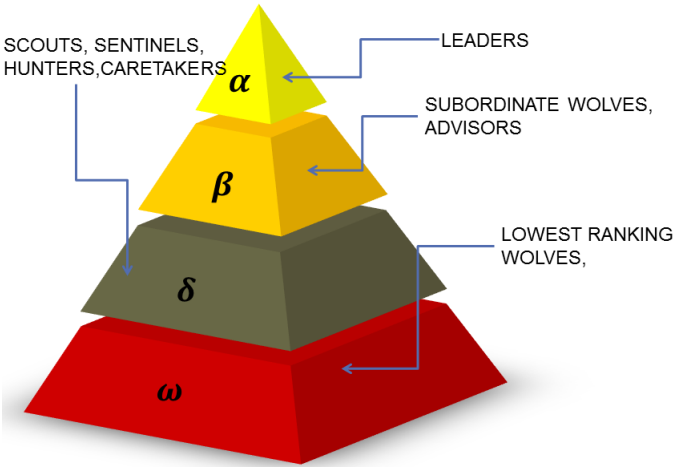


Figure 2.3: Social hierarchy of Grey Wolves.

implies that they are at top of the food chain. The crux of this algorithm is to simulate the Grey wolf behavior which tend to live in a pack. They have a stringent social dominant hierarchy described as follows: The top level comprises the leaders, called alpha. The alpha is the commander of the pack and is responsible for making the decisions. The persistence of the wolf pack is based on alpha's decision. The beta forms the second level which constitute the subordinate wolves. The purpose of a beta is to help the alpha in making important decisions or other activities. The penultimate level consists of subordinate wolves, called delta. The members in this category consist of scouts, sentinels, elders, hunters and caretakers. Scouts are liable for observing the boundaries of region and warn the rest of the pack in case of imminent danger. The sentinels are mandated to protect and guarantee the safety of the pack. Elders are the expertise wolves who used to be alpha or beta. Hunters help the alphas and betas when hunting prey and providing food for the pack and the caretakers are responsible for caring for the weak, ill, or wounded wolves in the pack. The bottom level in the pecking order is omega. The omega wolves have to comply with all the other dominant wolves. In some cases the omega even substitutes as a babysitter in the pack.

The key attribute of Grey wolves is in its ability to memorize the positions of prey and to encircle them. The alpha leads the hunting pack. Consequently, to model the social hierarchy

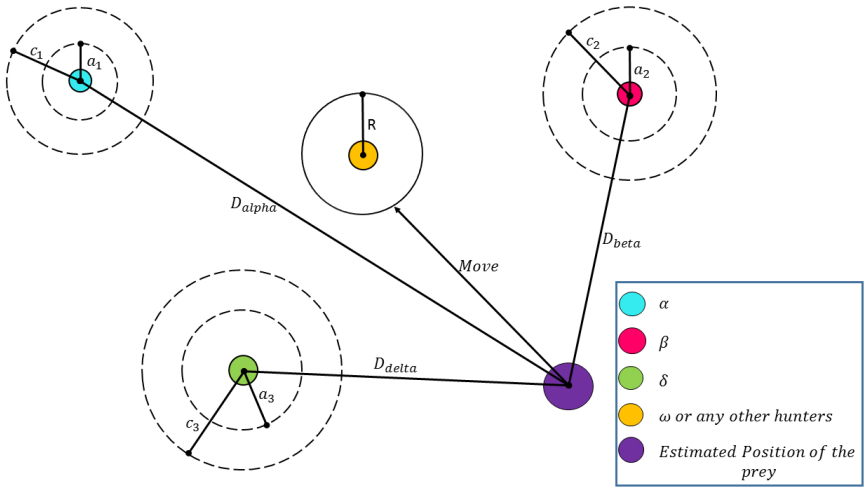


Figure 2.4: The design search space positioning for GWO-DSE [3]

of wolves (shown in Figure. 2.3) mathematically in GWO, the fittest solution is considered to be the alpha (α). The beta (β) and delta (δ) is similar to the second and the third optimal solutions, respectively. The rest of the candidate solutions are assumed to be omega (ω). The alpha, beta and delta guide the entire hunting (optimization) process to form three best solutions which in turn assists the ωwolves to update their positions.

A. Notations for GWO

Consequently, the prey location will be the optimal solution and the wolves represent the possible solutions in search space. The wolves with least proximity to the prey are the α wolves and they represent the best solutions among the available solutions. Next to best solutions are the β solutions while δ wolves form the third best solutions. Let X_α, X_β and X_δ denote their locations in the search space respectively. The ωwolves update their position in the search space based on their relative positions from α, β, and δ wolves. Figure. 2.4 shows the positioning of the wolves and prey and the parameters involved in the equations used for updating the positions of the wolves in the search space. The following set of steps must be applied for hunting a prey:

- Initialization

- Prey encircling
- Hunting
- Attack
- Search again

B. Initialization of GWO

The magnitude of the wolf pack is assumed as W_n. The locations for W_n candidate solutions are generated randomly during the phase of initialization as described by Eqs. (2.10) and (2.11) where each candidate wolf is $\overrightarrow{X_i}$:

$$\overrightarrow{X_i} = [f_{\text{rand}}(W_n) \cdot (ub - lb) + lb]$$
$$\text{where } f_{\text{rand}}(x) \sim \text{U}([1, \text{x}]), \quad (2.10)$$

where f_{rand} is a uniformly distributed random variable between 0 and x, ub and lb represent upper and lower bounds. If the candidate solutions generated by Eq. (2.10) is beyond the specified range of $[lb,\ ub]$, the GWO framework must reevaluate the search space using Eq. (2.11), thereby ensuring the initial solution lies within the requisite boundaries.

$$\overrightarrow{X_i} = [\overrightarrow{X_i} \cdot (u + \ell)] + ub \cdot u + lb \cdot \ell$$
$$\text{where } u = \overrightarrow{X_i} > ub,\ \ell = \overrightarrow{X_i} < lb \quad (2.11)$$

The GWO framework records fitness values ${}^{fit}\mathcal{F}$ using a fitness function similar to Eq. 2.1 depending on the context of the problem being solved for each and every candidate solution $\overrightarrow{X_i}$. The three most optimal candidate solutions are shortlisted as $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ respectively.

C. Prey Encircling

The pack encircles a prey by repositioning individual agents according to the prey location, as follows: A prey is encircled by the wolf pack by changing the position of individual agents in accordance to the prey location defined by Eq. (2.12).

$$\overrightarrow{X}(t\ +\ 1) = \overrightarrow{X_p}(t) + \overrightarrow{A}\ \cdot\ \overrightarrow{D} \tag{2.12}$$

where t is the iteration, $\overrightarrow{X_p}$ is the prey position, $\overrightarrow{X}$ is the Grey wolf position and $\overrightarrow{D}$ is given by,

$$\overrightarrow{D} =\mid \overrightarrow{C} \cdot \overrightarrow{X_p}(t) - \overrightarrow{X}(t) \mid \tag{2.13}$$

where $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors calculated as follows:

$$\overrightarrow{A} = 2a \cdot \overrightarrow{r_1} - a \tag{2.14}$$

$$\overrightarrow{C} = 2\overrightarrow{r_2} \tag{2.15}$$

where a linearly diminishes over the course of iterations controlling *exploration* and *exploitation*, and $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are random vectors over the range of [0, 1]. The value of a is the same for all wolves. The Eqs. (2.14) and (2.15) imply that a wolf can update its position in the design search space around the prey at any random location.

D. Hunting

The entire wolf pack participates in the activity of hunting based on information relayed by the α, β and δ wolves, which are expected to record the prey location as specified below,

$$\overrightarrow{X}(t\ +\ 1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{2.16}$$

where $\overrightarrow{X_1}$, $\overrightarrow{X_2}$ and $\overrightarrow{X_3}$ are defined as follows:

$$\overrightarrow{X_1} =\mid \overrightarrow{X_\alpha} - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha} \mid \tag{2.17}$$

$$\overrightarrow{X_2} =\mid \overrightarrow{X_\beta} - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta} \mid \tag{2.18}$$

$$\overrightarrow{X_3} =\mid \overrightarrow{X_\delta} - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta} \mid \tag{2.19}$$

where $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ are the first three best solutions for the given iteration t, $\overrightarrow{A_1}$, $\overrightarrow{A_2}$ and $\overrightarrow{A_3}$ are defined as in Eq. (2.14), and $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ and $\overrightarrow{D_\delta}$ are assigned using the following:

$$\overrightarrow{D_\alpha} =\mid \overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \overrightarrow{X} \mid \tag{2.20}$$

$$\overrightarrow{D_\beta} =\mid \overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \overrightarrow{X} \mid \tag{2.21}$$

$$\overrightarrow{D_\delta} =\mid \overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \overrightarrow{X} \mid \tag{2.22}$$

where $\overrightarrow{C_1}$, $\overrightarrow{C_2}$ and $\overrightarrow{C_3}$ are specified in Eq. (2.15).

The Eqs. (2.20, 2.21, 2.22) are interpreted by the fact that α, β, and δ wolves know the best position of the prey and the remaining wolves adapt their positions with respect to position of these wolves.

Attacking Stage

The agents surround the prey, which is achieved by decrementing the exploration rate a. The parameter a is linearly updated for every iteration to range from 2 to 0 using Eq. (2.23).

$$a = 2 - t \cdot \left(\frac{2}{\text{Iter}_{\text{MAX}}} \right) \tag{2.23}$$

where t specifies the iteration count and Iter_{MAX} specifies the maximum number of iterations allowed for the GWO. The exploration and exploitation are guaranteed by the adaptive nature of a value[3], thus allowing GWO to ensure simple transition between exploration and exploitation cycles. Therefore half of the iterations are reserved for exploration while

the later half is dedicated to exploitation. The interpretation of this stage is analogous to wolves moving or switching their position to any random position sandwiched between their current position and the position of the prey.

E. Search Again

In order to search for the prey, the wolves begin to diverge their paths from each other. This behavior is modeled in GWO by initializing large values for parameter a to allow for exploration of the design search space. The motive is to ensure that the wolves diverge as far as possible from each other to ensure better exploration of the design search space and subsequently converge later to attack, when they spot a better prey. Any wolf can find a better prey (optimal solution). As they get closer to the prey, they will transform into new alphas and the remaining wolves shall be split into beta, delta, and omega depending on the proximity to the prey. The parameter a assigns random weights to prey and specifies the impact of the prey in characterizing the separation of wolves as shown in Eqs. (2.12 and (2.13). This concept helps GWO to exhibit necessary random behavior, which favors exploration and evading local optima. Therefore it can be concluded that the parameter a provides random values throughout the process of exploring design search space which accentuates exploration not only at the beginning of the optimization process but also until its completion.

2.6.3 Advantages of GWO

GWO has several advantages over the existing meta-heuristic algorithms such as PSO, ABC, GA, DE [46]: (1) It is simple to implement, (2) It can maintain the information about the search space and keeps the best solution obtained, (3) It has fewer parameters for fine tuning, and (4) It is a derivative–free algorithm. Due to these comparatively beneficial properties, the performance of GWO far outpaces existing population-based meta-heuristic techniques and it is the hallmark of our optimization procedure for IoT research in this thesis. Consequently

we choose GWO algorithm as a optimizer algorithm to build our IoT specific GWO-DSE and GWO-Cross-layer frameworks described in following Chapters. 3 and 4 respectively.

Chapter 3

IoT Network Architecture Cross-layer Optimization

In this chapter, we propose a novel cross-layer optimization framework and communication module for the IoT using optimizing parameter road map provided by delta diagrams illustrated in [38], 3.3 and 3.4. Our proposed framework captures the high device and service heterogeneities of the IoT. In particular, it exploits the interrelations among the device specifications, physical layer, link layer and network layer, to find the optimal routing paths and their corresponding communication parameters, which jointly optimize the point-to-point delay and energy consumption for given QoS requirements. In addition, the impact of the Internet on the achieved QoS is also taken into account. The results show that the proposed solution outperforms existing layered solutions and the joint-objective cross-layer solution can balance between different design objectives.

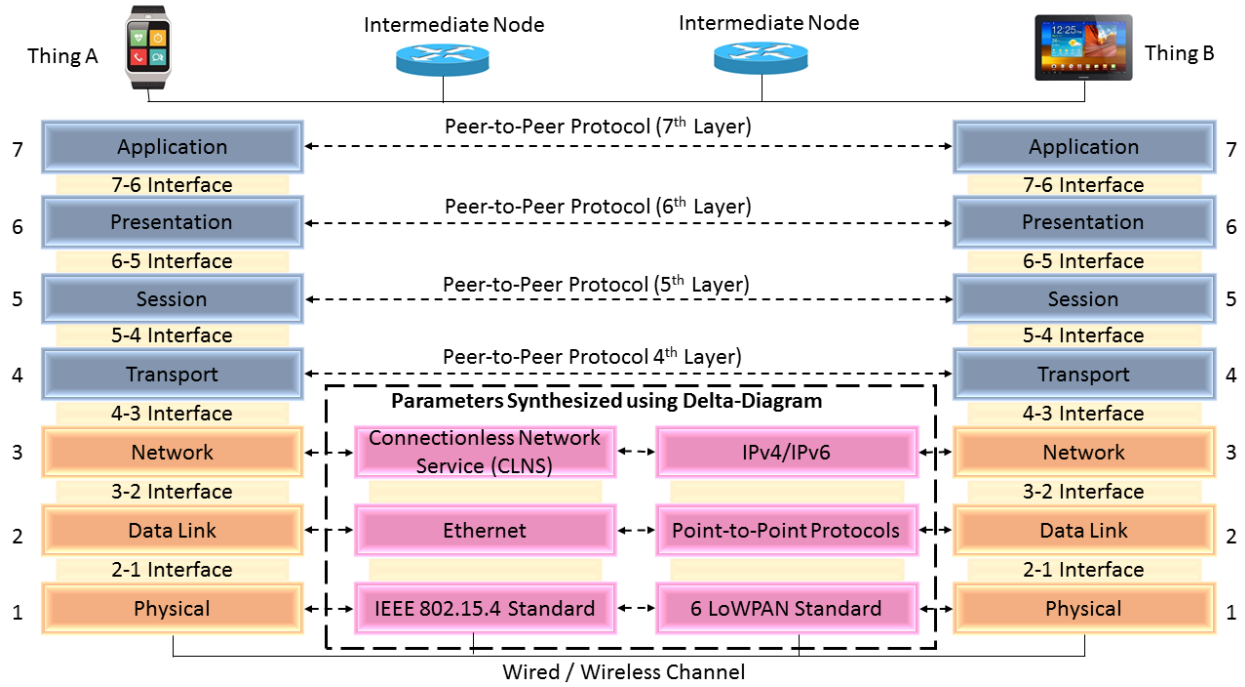


Figure 3.1: Open Systems Interconnection reference model magnifying Network, Data Link and Physical layers

3.1 Problem Statement

Cross-layer communication schemes[47, 48] are contemplated to provide adaptive solutions for the IoT as they are most suitable for high heterogeneity in device hardware capabilities and the communication requirements among things. In this chapter, a novel cross-layer framework for the IoT is proposed based on GWO to accurately capture both the high heterogeneity of the IoT and the impact of the Internet as part of the network architecture. The fundamental part of this framework is developed to obtain the optimal routing paths and the communication parameters among things, by exploiting the interrelations among different layer functionalities in the IoT. The parameter paths for optimizing the layers is provided by a novel approach in form of delta diagram defined in Figures 3.2, 3.3 and 3.4. Moreover, a cross-layer communication protocol called *intel*-LEACH[49] is devised to implement and test this optimization framework under practical scenarios. The results show that the proposed

solution can achieve a global communication optimum and outperforms existing layered solutions and highly flexible in nature, so that the concept can be adopted to synthesize any given communication protocol. This novel cross-layer framework is a primary step towards providing efficient and reliable point-to-point communication in the IoT.

One of the major hindrances in the IoT is is due to a very high heterogeneity exhibited by both the hardware capabilities and the communication requirements among different types of things. From the hardware perspective, things can have very different computation, memory, power or communication capabilities. For instance, a cellular phone or a tablet has much better communication and computation capabilities than a single-purpose electronic product such as a heart rate monitor watch.

Consequently, things can have very different Quality of Service (QoS) requirements in terms of delay, energy consumption or reliability. For example, minimizing the energy for communication/computation purposes is a major constraint for the batter-powered devices without efficient energy harvesting techniques. On the contrary, this energy constraint is not critical for the devices with power supply connection.

These two traits pose a conflict in designing a unifying framework which can take care of diversity of capabilities and functionalities of things. As a result, the heterogeneity traits of the network motivate the use of adaptive cross-layer communication schemes for the IoT. Several cross-layer protocols exist for Wireless Sensor Networks (WSNs)[50, 51], Wireless Mesh Networks (WMNs)[52] or Ad Hoc Networks (AHNs)[53]. However, they cannot be applied to the IoT due to following: First, the heterogeneity of the iot incurs that things have largely diverse hardware capabilities, different QoS requirements and individual goals. On the contrary, in WSNs, nodes usually have very similar hardware specifications, common communication requirements and a shared goal. Second, the Internet is involved in the IoT network architecture, from which it inherits a centralized and hierarchical architecture. In comparison, in WSNs, WMNs and AHNs, highly flat network architectures are considered, in which nodes communicate in a multi-hop fashion and also the Internet is not involved.

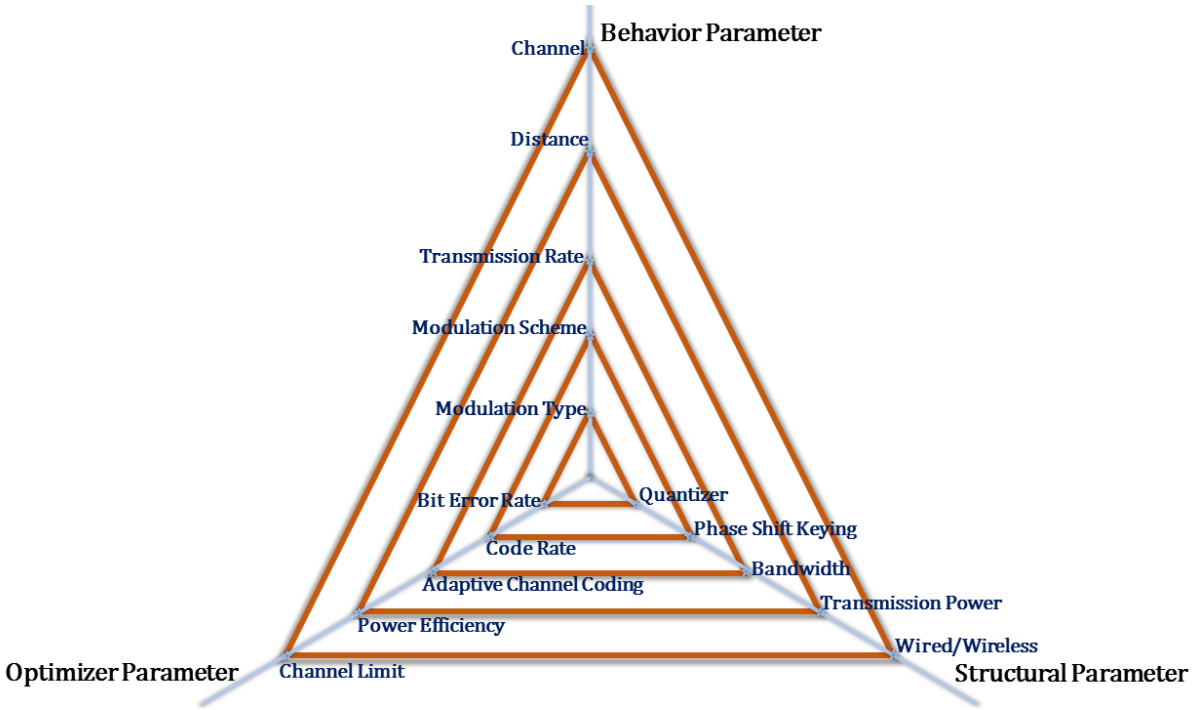


Figure 3.2: Delta chart synthesizer for Physical Layer

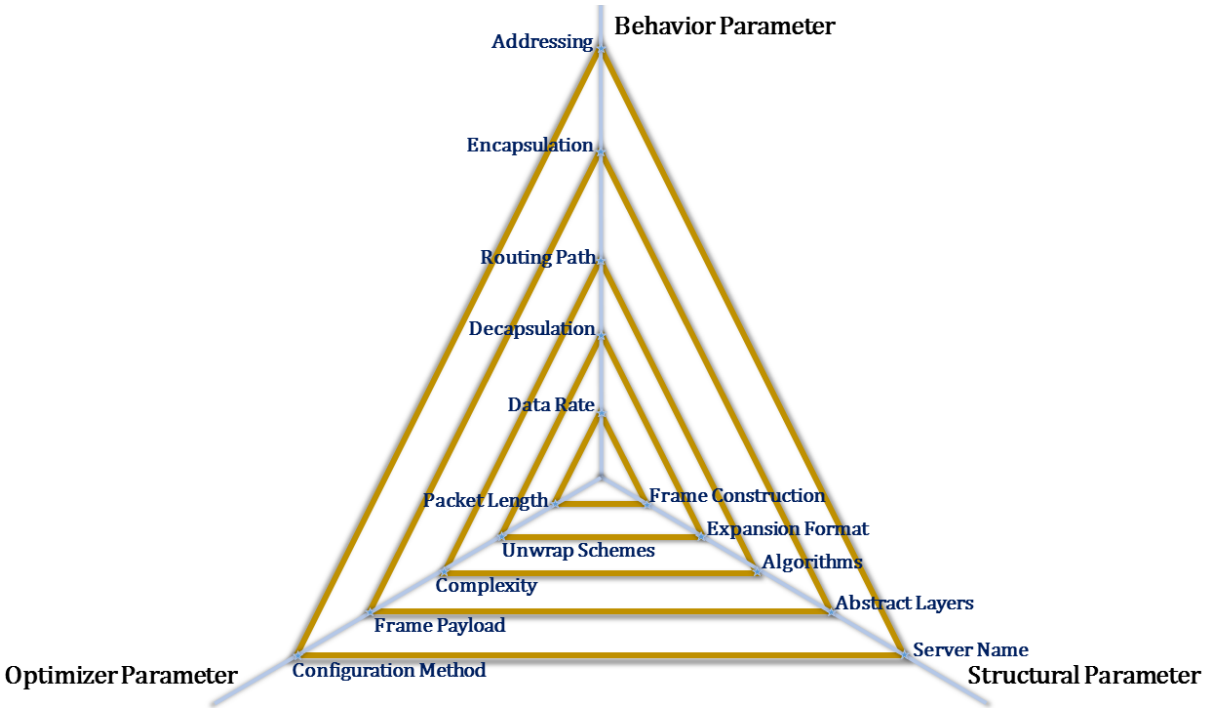


Figure 3.3: Delta chart synthesizer for Network Layer

3.2 Contribution

The contributions of this chapter are summarized as follows: We provide an in-depth review of the state of the art in IoT-related initiatives and standardization efforts, to better motivate the need of a unified cross-layer solution. We identify the interrelations among the device capabilities, physical layer, link layer and network layer, and explain how these are captured in our solution. We define a cross-layer optimization framework with a single weighted joint-objective function according to the service-dependent QoS requirements. A delta diagram as defined in Figures 3.2, 3.3 and 3.4 for physical, network and data link layers respectively for charting optimization path will greatly influence the choice of adaptive parameters used in our cross-layer framework.

A custom cross-layer protocol is utilized[49] to implement the cross-layer optimization framework in practical scenarios. We provide extensive simulation results which show that our proposed solution outperforms existing layered protocols. The proposed delta diagram (Figures 3.2, 3.3 and 3.4) based approach for synthesizing a communication system is the first unified optimization solution for point-to-point efficient and reliable communications in the IoT. The remainder of this chapter is organized as follows.

- discuss related work in terms of on-going standardization efforts for the IoT.
- describe the reference network architecture for the IoT that we consider throughout this chapter.
- introduce a delta diagram based synthesizer which abets in choosing the parameters for optimization as per system requirements.
- describe our design approach and develop the GWO-Cross-layer optimization framework which captures the existing relations among the different layers of the protocol stack described in Figure 3.1.

- describe the GWO-Cross-layer protocol operation needed to implement our optimization framework in practical scenarios.
- evaluate the performance of the proposed solution by means of simulation.

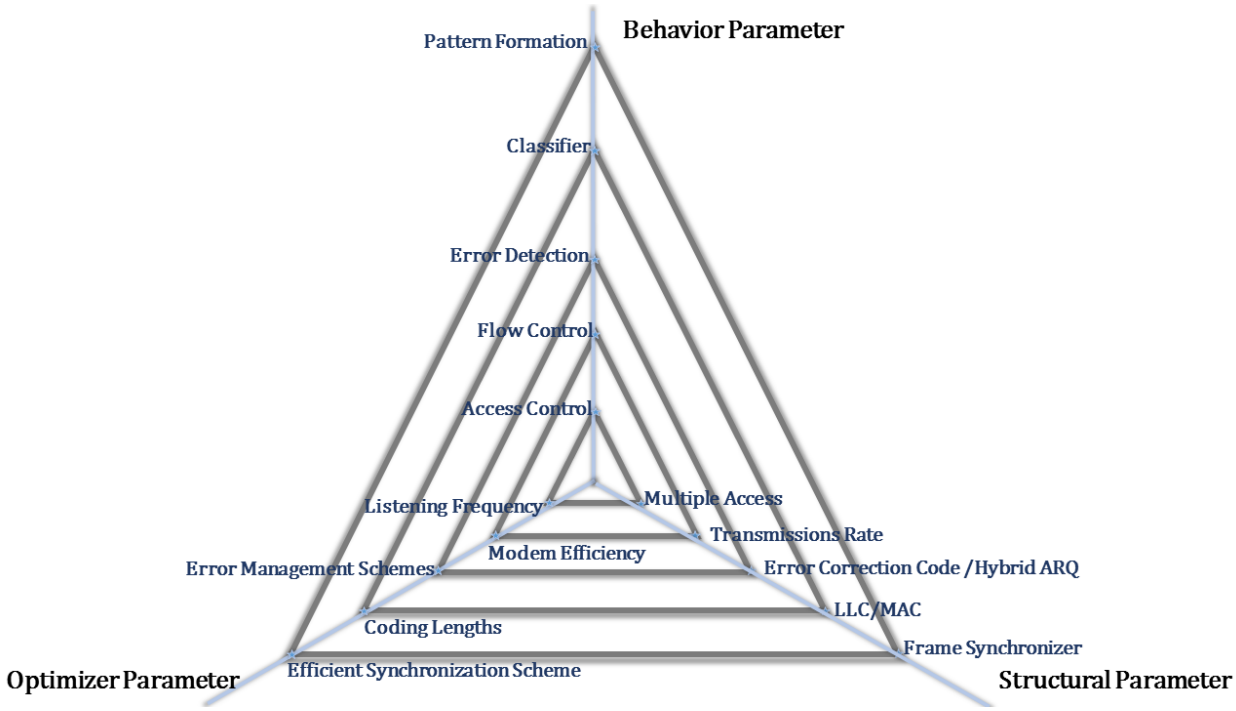


Figure 3.4: Delta chart synthesizer for data link Layer

3.3 IoT Standards

A few IoT deployment and standardization attempts by Research organizations still date are listed as follows.

- **IETF Low power Wireless Personal Area Networks (6LoWPAN) Standard[54]:** defines a set of protocols to integrate low-complexity devices which operate under the IEEE 802.15.4 Standard into IPv6 networks. However, several challenges appear due to, among others, the mismatch between the minimum packet size for IPv6 networks and that of IEEE 802.15.4, or the difficulty to manage routing tables for the expected number of nodes involved in the IoT. Mechanisms to guarantee point-to-point reliability are not provided, either. QoS requirement such as point-to-point delay or reliability.

- **ETSI Machine to Machine (M2M) Technical Committee[55]:** concentrates mainly on ad hoc networks among things in which Internet is not part of the system. For the time being, no specifications are provided for things addressing, location, and QoS. Furthermore, different devices use different network protocols to communicate within the same M2M network [15]. This dramatically increases the burden on the gateway, which needs to adapt every transmission among various devices.

- **IEEE 802.15.4 Standard[56]:** provides the specifications for the physical layer (frequency spectrum allocation, modulation, data rates, and power control) and the link layer (MAC and error control) for Low-Rate Wireless Personal Area Networks (LR-WPANs). However, it does not specify the implementation of higher-layer functionalities (e.g., routing, point-to-point reliability) or how the communication among things over the Internet is realized.

Contemporarily there are many independent solutions for the different layer functionalities in the protocol stack(Figure 3.1). Some of the related work can be outlined as follows,

- **Link layer:** the authors in [57] present a broad overview of the MAC protocols conducted in the field of wireless sensor and ad hoc networks. However, they fail to provide a vivid guidance on the choice of MAC techniques and the associated parameters. By contrast, the authors in [58] propose spatial correlation-based collaborative medium access control (CC-MAC), an energy efficient MAC that exploits spatial correlation in wireless sensor networks on the MAC layer. Both of them admit that the MAC layer plays an important role in the performance of the overall system and affects other layers, while they ignore these effects and their impact on the system performance. A delta diagram for synthesizing link layer is illustrated in Figure 3.4 which provides a road map for picking parameters to be optimized while framework execution is in progress.

- **Network layer:** the authors in [59] summarize the data routing algorithms and classify the approaches into three categories: data-centric, hierarchical and location based. In addition, in [60], the authors study the design trade-offs between energy and communication overhead savings for the existing routing protocols. However, neither of them is appropriate for IoT since the physical attributes of nodes are not taken into account, which have direct impact on the performance and even the validity of the routing algorithms. Furthermore, they omit the interactions between routing algorithm and other layers. A delta diagram for synthesizing network layer is illustrated in Figure 3.3 which provides a road map for picking parameters to be optimized while framework execution is in progress.

- **Physical layer:** in [61], the authors advocate a physical layer-driven approach to protocol design for wireless sensor networks with emphasis on the underlying hardware parameters. Referring [58], the authors demonstrate the importance of the physical layer modeling on the performance evaluation. Nevertheless, neither of them exploits the interrelation among the MAC and other layer functionalities with the physical layer and hence, their joint influence on the point-to-point communication performance. A

delta diagram for synthesizing physical layer is illustrated in Figure 3.2 which provides a road map for picking parameters to be optimized while framework execution is in progress.

From the above, it is clear that these layered solutions cannot successfully capture the twofold heterogeneity of the IoT. Cross-layer protocols have been successfully developed in other paradigms such as Wireless Sensor Networks (WSNs)[50, 51], Wireless Mesh Networks (WMNs)[52] or Ad Hoc Networks (AHNs)[53], among others. Consequently these cannot be directly used in the IoT, due to salient differences in the paradigms.

3.4 Network Architecture

The Network Architecture for our Cross-layer framework comprises the following.

- **Gateway Access Points (GAPs):** advanced devices which play the role of local network coordinator as well as interface and gateway for the communication over the Internet. We refer to the set of things under the control of a single GAP as the GAP domain.

- **The Internet:** fundamental component of the IoT. In our analysis, we treat the Internet as a black box which is characterized by an stochastic queuing delay model and a stochastic packet loss model[62].

- **Things:** physical objects with very diverse hardware specifications in terms of communication, computation, memory and data storage capacity, or transmission power. Personal electronic devices, home appliances or all sorts of equipment, are examples of things.

Figure 3.5 illustrates the network architecture of the IoT, in which several things are connected to the Internet via a common GAP. In common scenarios (e.g., at home, in the office),

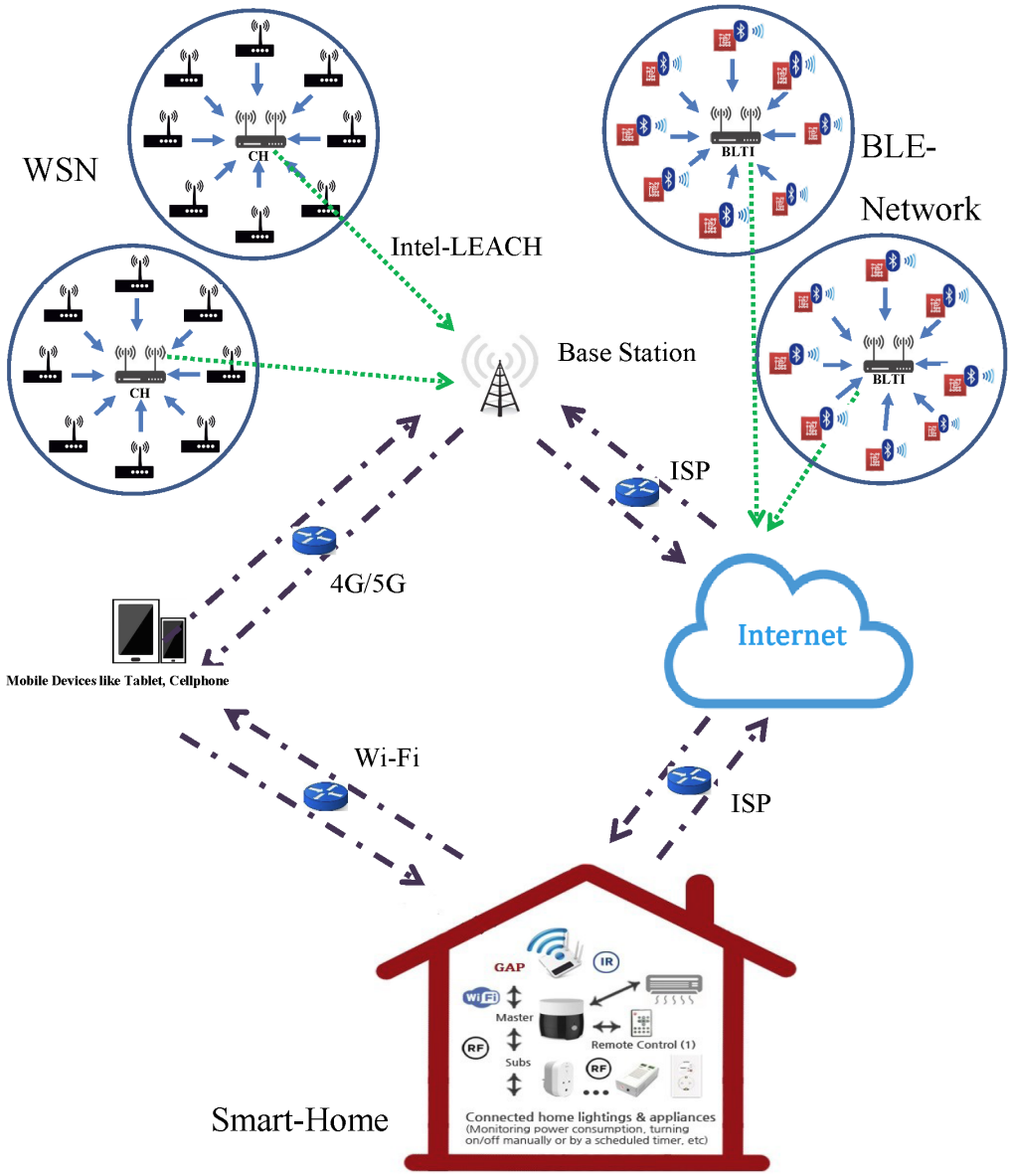


Figure 3.5: A example Network Architecture for IoT

each GAP domain is composed by a few tens of things.

The Additional network considerations are summarized as follows.

- **Interconnection between things and GAP:** the GAP is able to directly communicate with all the things in its domain in a single hop. However, not all the things are able to directly communicate with the GAP, in which case multi-hop links are required.

This asymmetry in the links results from the fact that the transmission power of the GAP is generally higher than that of many things.

- **Network knowledge:** the GAP is able to collect information about all the things in its domain (e.g., during the network association phase), such as device type, approximate location, communication or computation capabilities, among others. By contrast, things might only know the GAP ID, but have no information about other things. Things are not able to perform complex computation locally.

- **Communication types:** we distinguish two types of communication, namely, intra-GAP and inter-GAP. Intra-GAP communication is established among things within the same GAP domain. Despite direct transmission among things might be possible, different capabilities among them motivates the use of the GAP for coordination. Inter-GAP communication is established among things in different GAP domains. In this case, the GAP serves as a gateway as well as the coordinator.

- **Centralized decision:** the GAP is able to run optimization algorithms locally by exploiting its network knowledge. Therefore, the computation complexity is shifted from things as in a distributed manner to the GAP in a centralized fashion. As a result, global optimal routes and communication parameter values for the GAP domain can be obtained. This is feasible since the size of the GAP domain is in the order of a few tens of things and, thus, the resulting complexity is affordable for standard GAP hardware capabilities.

3.5 Cross-layer Optimization

In this section, we propose a novel cross-layer optimization model for the IoT using a parameter synthesizing road map provided by Delta diagrams as illustrated in Figures. 3.2, 3.3 and 3.4 for physical, network and data link layers respectively. Furthermore the Delta

diagram synthesizer can be generalized and extended to any layer chosen by the optimization model adopted by a communication system. For the IoT network architecture specified in Fig. 3.5, the Delta diagram is applied to synthesize three layers: Physical, network and data link layers of the protocol stack described in Fig. 3.1.

We adopt a resource allocation approach which involves centralized management [63] to estimate resource availability and environmental dynamics, coordinate the allocation of resources across applications and nodes, and therefore adapt the protocol parameters at each layer based on the synthesizer road map provided by Delta diagram. This approach assists in integrating scattered communication functionalities into a united coherent optimization model and provide an flexible solution for cross-layer design and control. We use this approach to jointly control and synthesize a select case of Quality of Service (QoS) requirements. Based on our selected QoS case, we synthesize the physical layer of network architecture of Fig. 3.5 using the Delta diagram in Figures 3.6 that results in parameters, channel and modulation. Similarly, data link layer synthesizer (Figures 3.7) utilizes only MAC layer parameters like Access Control and error control and network layer synthesizer in Figures 3.10 produces parameters like addressing and routing for the selected case. The synthesizer produces these parameters based on case with differentiated services for applications having contrasting QoS requirements, ranging from error-limited applications or minimum energy consumption applications to highly-delay-sensitive applications or any combination of them. We can model this case as a multi-objective optimization problem that must simultaneously optimize multiple conflicting objectives of QoS requirements subject to certain constraints of Delta diagram synthesizer, given by a minimization fitness function that produces a triple

as follows,

$$
\begin{aligned}
\mathcal{F}^{\mathtt{fit}} &= \underset{x}{\text{minimize}}\ J(\mathsf{x}) \\
\iff \underset{x}{\text{minimize}}\ J(\mathsf{x}) &= \Big(\ x_0,\ x_1,\ x_2\ \Big) \\
&= \Big(\ \xi^{\mathtt{p2p}}_{min}, \mathcal{T}^{\mathtt{p2p}}_{min}, \mathsf{PE}^{\mathtt{p2p}}_{min}\ \Big) \\
\text{subject to } O_1 &= \left|\frac{\xi^{\mathtt{opt}} - \xi^{\mathtt{p2p}}}{\xi^{\mathtt{opt}}}\right| \cdot \left|\frac{\mathsf{PE}^{\mathtt{p2p}} - \mathsf{PE}^{\mathtt{Th}}}{\mathsf{PE}^{\mathtt{Th}}}\right| \\
O_2 &= \left|\frac{\mathsf{PE}^{\mathtt{p2p}} - \mathsf{PE}^{\mathtt{opt}}}{\mathsf{PE}^{\mathtt{opt}}}\right| \\
O_3 &= \left|\frac{\mathcal{T}^{\mathtt{p2p}} - \mathcal{T}^{\mathtt{0pt}}}{\mathcal{T}^{\mathtt{0pt}}}\right| \\
\text{where } \delta^{\xi} \cdot O_1 \cdot \delta^{\mathsf{PE}} &+ O_2 \cdot \delta^{\mathsf{PE}} + O_3 \cdot \delta^{\mathcal{T}} \\
\delta^{\mathsf{PE}} &= 1 - (\delta^{\xi} + \delta^{\mathcal{T}})
\end{aligned}
\tag{3.1}
$$

The fitness function $\mathcal{F}^{\mathtt{fit}}$ is a minimization function. It takes a vector x of dimensions 3×1 having elements x_0, x_1 and x_2 and produces a minimized triple ($\xi^{\mathtt{p2p}}_{min}$, $\mathcal{T}^{\mathtt{p2p}}_{min}$, $\mathsf{PE}^{\mathtt{p2p}}_{min}$) for a given instance when subjected to constraint conditions in Eq. (3.1). For the IoT network architecture in Figure. 3.5, we consider the end-to-end communication as point-to-point communication due the vast diversity of end nodes and its heterogeneousness with respect to its computing capacity, storage, etc. The terms in Eq. (3.1) are as follows: $\mathsf{PE}^{\mathtt{p2p}}$ is point-to-point packet error rate ratio which is the number of error packets after applying Forward Error Correction (FEC) divided by the total number of received packets. It is important to note that a packet is the unit of data for radio transmission with applicable FEC. $\mathcal{T}^{\mathtt{p2p}}$ and $\xi^{\mathtt{p2p}}$ represent point-to-point delay and energy consumption, while δ^{ξ}, δ^{PE}, $\delta^{\mathcal{T}}$ are linear weighting coefficients for energy consumption, time delay and point-to-point packet error rate respectively. $\mathsf{PE}^{\mathtt{opt}}$, $\mathcal{T}^{\mathtt{0pt}}$ and $\xi^{\mathtt{opt}}$ are defined as the quintessential values[64] for point-to-point energy consumption, delay and packet error rate for normalizing purposes respectively. These quintessential values are defined to be the most optimistic and usually unattainable minimum values, which are used to provide the non-dimensional objective functions and can be computed off line. These values are used to normalize each of constraint terms O_1, O_2

and O_3, and optimize their deviations with respect to a pre-defined threshold, instead to minimize their absolute values. This is done due to the fact that there are three differing objectives which are measured in separate units as well as their order of magnitude. The constraints terms in Eq. (3.1) are defined as follows,

Point-to-point energy coefficient is

$$O_1 \leftarrow \left| \frac{\xi^{\mathtt{opt}} - \xi^{\mathtt{p2p}}}{\xi^{\mathtt{opt}}} \right| \cdot \left| \frac{\mathsf{PE}^{\mathtt{p2p}} - \mathsf{PE}^{\mathtt{Th}}}{\mathsf{PE}^{\mathtt{Th}}} \right| \tag{3.2a}$$

Point-to-point packet error rate coefficient is

$$O_2 \leftarrow \left| \frac{\mathsf{PE}^{\mathtt{p2p}} - \mathsf{PE}^{\mathtt{opt}}}{\mathsf{PE}^{\mathtt{opt}}} \right| \tag{3.2b}$$

Point-to-point timing coefficient is

$$O_3 \leftarrow \left| \frac{\mathcal{T}^{\mathtt{p2p}} - \mathcal{T}^{\mathtt{0pt}}}{\mathcal{T}^{\mathtt{0pt}}} \right| \tag{3.2c}$$

3.6 Physical Layer Traits

At the level of physical layer (Figure. 3.1), the things of IoT can have their communication synthesized on any of the network domains relating to Behavioral, structural and Optimizer parameters depicted in Figure 3.2. For example, in case of network architecture of Figure. 3.5, different things have a variety of maximum transmission power parameter which tend to choose differing modulation schemes and have varying data storage capacities (i.e., the number of packets that things can locally queue). As a consequence the physical layer Delta diagram synthesizer of Figure 3.6 depicts this action corresponding to parameter level parsing across specified domains.

3.6.1 Channel model and frequency allocation

Our model adopts the frequency spectrum allocation based on IEEE 802.15.4 standard[56] where things are able to dynamically select any of the 5-MHz-wide sub-channels in $2400 -$

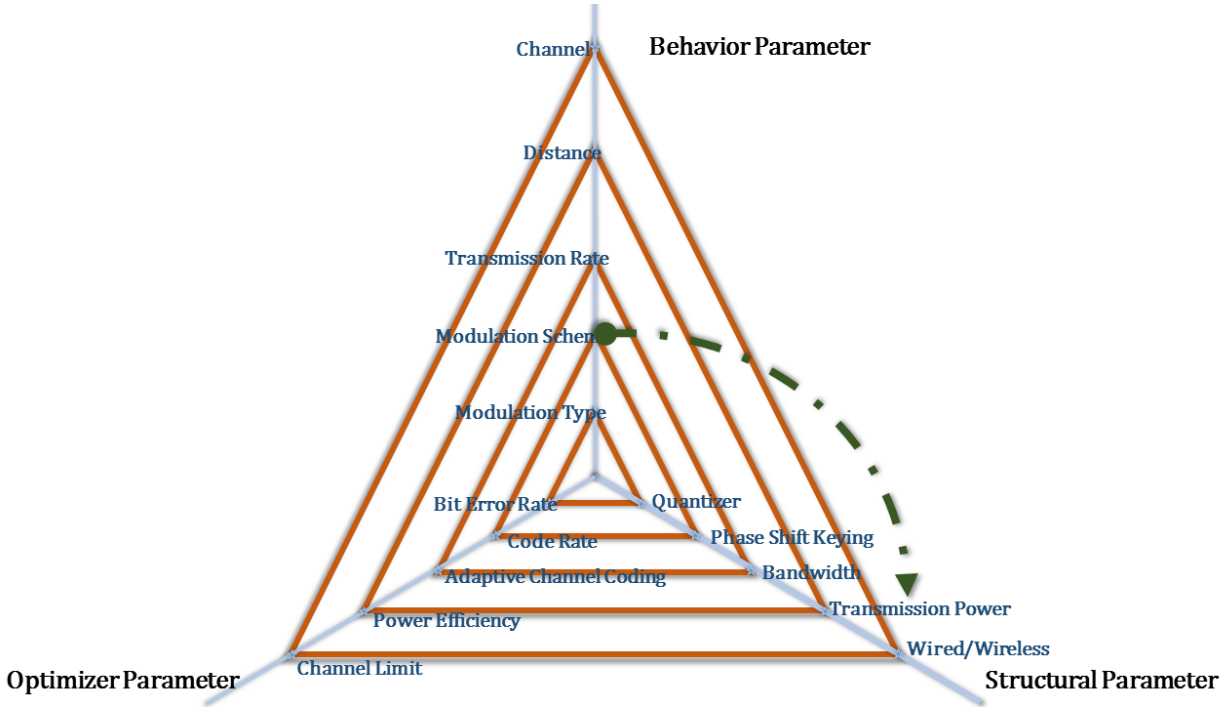


Figure 3.6: Parameter synthesizing action for Physical Layer Synthesis

2480 MHz band. Due to the fact that many applications of IoT are based indoor (e.g., home, office, warehouses), we set the ITU channel model for indoor propagation [65]. The total path loss $\mathcal{L}$ in dB is given by

$$\mathcal{L}(d, f_c) = A \log_{10}(d) + \mathcal{L}_f(r) + 20 \log_{10}(f_c) - 28 \tag{3.3}$$

where f_c represents carrier frequency in MHz, d is the distance in meters to transmit, r is the number of floors between the transmitter and the receiver (we consider only one floor in our current scenario, i.e., $r = 1$ and A is the distance attenuation coefficient (i.e., $A = 20$ in our simulations), and $\mathcal{L}_f$ is the floor penetration loss factor since whose value is zero for only one floor.

3.6.2 Power, Modulation and BE for Transmission

The Bit Error Rate (BE) is directly affected by the transmission power and modulation. The Bit Error Rate over a data link ℓ denoted by $\mathsf{BE}_\ell^{\mathrm{DL}}$ is determined by its respective Signal-to-Noise Ratio $\mathsf{SNR}_\ell^{\mathrm{DL}}$ and modulation M_ℓ as,

$$\begin{aligned} &\mathsf{BE}_\ell^{\mathrm{DL}}(d,\ f_c) = \mathcal{S}_\psi\Big(\mathsf{M}_\ell,\ \mathsf{SNR}_\ell^{\mathrm{DL}}(d,\ f_c)\Big) \\ &\text{where } \mathsf{SNR}_\ell^{\mathrm{DL}}(d,\ f_c) = \mathsf{P}_\ell^{Tx} - \mathsf{P}^{\mathrm{noise}} - \mathcal{L}_\ell^{\mathrm{DL}}(d,\ f_c) \end{aligned} \tag{3.4}$$

where the BE is calculated by function $\mathcal{S}_\psi$ for a given modulation and SNR, and it is common knowledge for standard modulations. From Eq. (3.4), we observe that the SNR (in dB) of a data link ℓ has a transmission power P_ℓ^{Tx} in dB, total noise power $\mathsf{P}^{\mathrm{noise}}$ in dB at the receiver and a path loss $\mathcal{L}_\ell^{\mathrm{DL}}(d, f_c)$ for data link ℓ estimated using Eq. (3.3)

The three most predominant modulations, namely, BPSK, QPSK and 16-QAM are utilized for simulation purposes using intel-LEACH protocol[49], but any other modulation can be easily included in the framework. These modulations are distinguishable in terms of achievable BER for a given SNR, i.e., $\mathcal{S}_\psi$ in Eq. (3.4), and spectral efficiency, i.e., theoretical achievable data bit-rate for a given transmission bandwidth. A higher complexity modulation exhibits higher bandwidth efficiency, which results in a higher transmission data rate or a shorter transmission time, ${}_i^{data}T$. However, these come with the cost of an increase in the energy consumption at the transmitter and the receiver due to the increase in the computation complexity, as well as, potentially, also in the processing time. Furthermore, more complex modulations require a higher $\mathsf{SNR}_\ell^{\mathrm{DL}}$, thus, higher P_ℓ^{Tx}, to achieve the same $\mathsf{BE}_\ell^{\mathrm{DL}}$. At the same time, though, the transmission time, $\mathcal{T}_\ell^{\mathtt{Data}}$, is shorter, which also affects the link energy consumption. These interrelations are properly captured in the following GWO framework.

3.6.3 Probability of Packet Drop-out and Data Buffer Capacity

The probability of packet drop-out is directly influenced by data storage capacity μ of the things. Thus, the probability of discarding a packet at link ℓ, $\mathbb{P}_{\ell}^{\text{drop}}$ is related to the fact that it cannot be queued at the transmitter or at the receiver given by,

$$\mathbb{P}_{\ell}^{\text{drop}} = \mathcal{G}_{\psi}(\mu_{\ell}, \Re_{\ell}) \tag{3.5}$$

such that the maximum number of packets μ_{ℓ} that can be queued at the transmitter or the receiver and the total local traffic $\Re_{\ell}$ (self traffic and relayed traffic) is influenced by function $\mathcal{G}_{\psi}$. Considering a simplest case of Poisson traffic, the transmitter and receiver are modeled as single serve First In First Out (FIFO) queue having data buffer capacity of μ_{ℓ}.

3.7 Data Link Layer Synthesis

In IEEE 802 LAN/MAN standards, the medium access control (MAC) sublayer (also known as the media access control sublayer) and the logical link control (LLC) sublayer together make up the data link layer. Within that data link layer, the LLC provides flow control and multiplexing for the logical link (i.e. EtherType, 802.1Q VLAN tag, etc.), while the MAC provides flow control and multiplexing for the transmission medium.

For the IoT network architecture in Figure. 3.5, the delta diagram for data link layer synthesizer (Figure 3.3) utilizes only MAC layer parameters like Access Control and error control as per QoS requirements defined for Eq. (3.1). We analyze the impact of the error control mechanism and the MAC protocol on the network performance, as well as, their interrelations with other layers and the limitations imposed by the things capabilities. This parameter level parsing action is depicted sequentially in data link layer delta diagram synthesizer of Figures 3.7, 3.8 and 3.9.

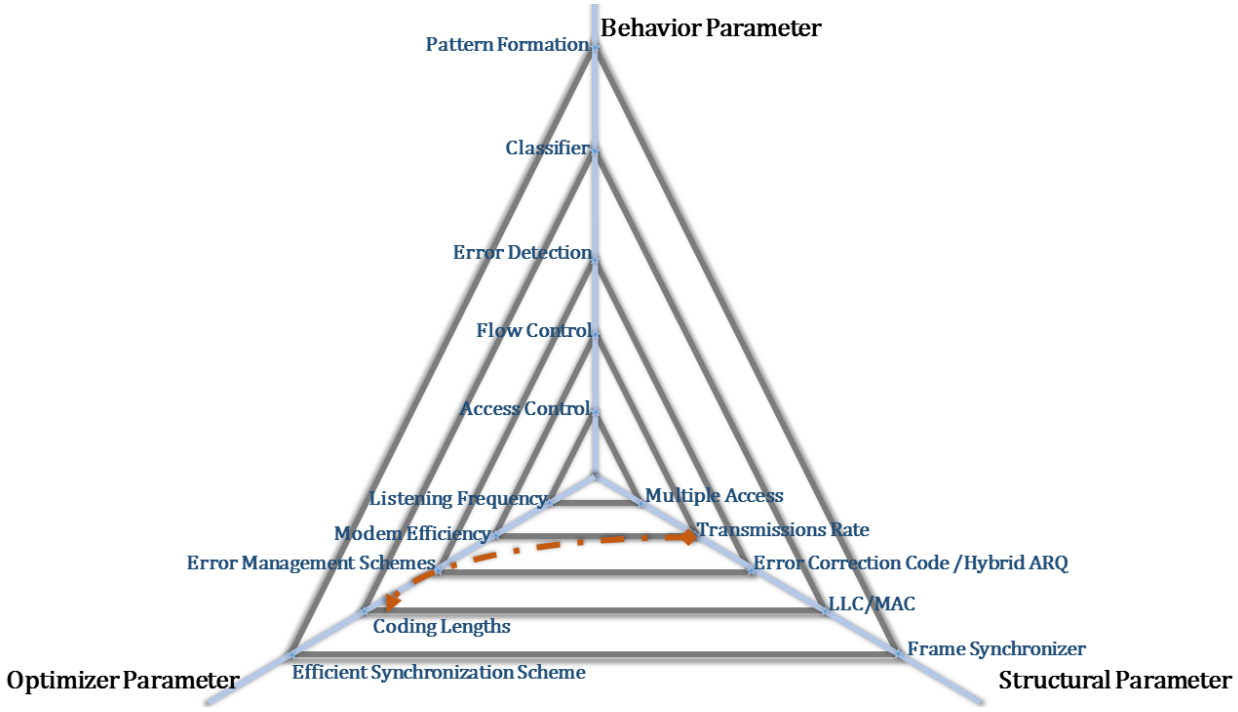


Figure 3.7: Parameter synthesizing start action for Data Link Layer Synthesis

3.7.1 Packet Error Rate and Error Control

For our cross layer model we adopt a Hybrid ARQ error control scheme [66], the overall packet error rate over data link ℓ is given by

$$\mathsf{PE}_{\ell}^{\text{overall}} = \mathcal{K}_{\psi}\left(\mathsf{PE}_{\ell}^{\text{non-Code}},\ \mathsf{N}_{\ell}^{\max},\ \mathsf{N}_{\text{FEC}}^{\text{bits}}\right) \tag{3.6}$$

where $\mathcal{K}_{\psi}$ is a function that relates the PE of data link ℓ after Hybrid ARQ error control, $\mathsf{PE}_{\ell}^{\text{overall}}$, with the uncoded data link $\mathsf{PE}_{\ell}^{\text{non-Code}}$. In the above equation, $\mathsf{N}_{\text{FEC}}^{\text{bits}}$ is the FEC redundancy length and $\mathsf{N}_{\ell}^{\max}$ is the maximum number of transmissions including retransmissions, which can be adjustable in our cross-layer model.

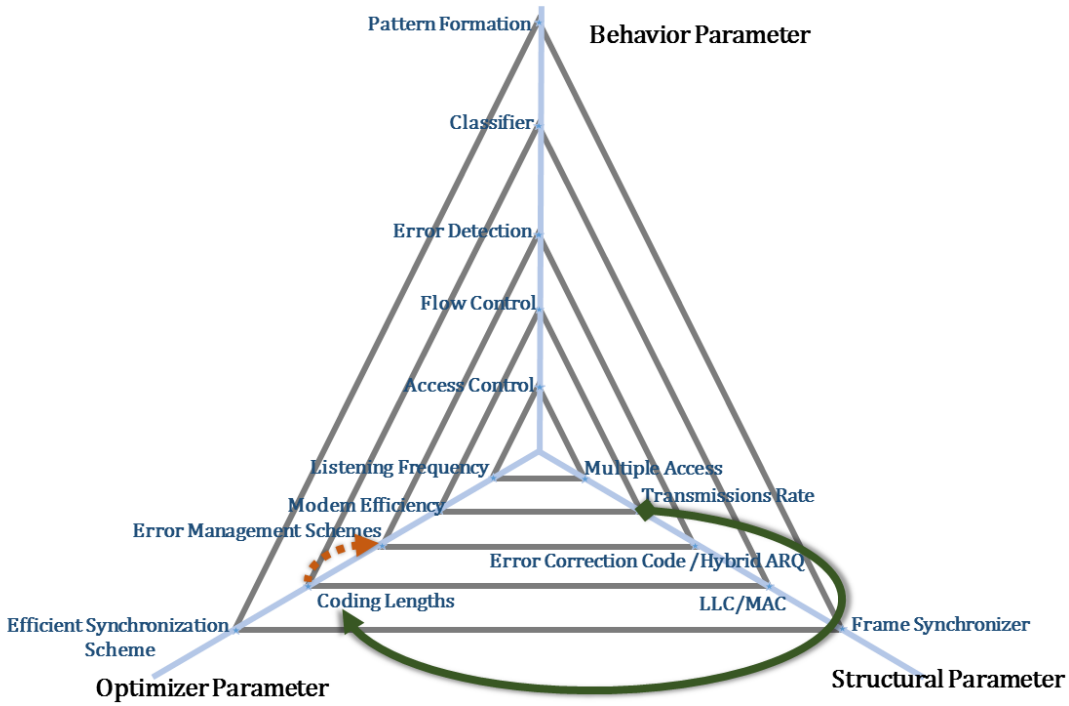


Figure 3.8: Parameter synthesizing sophomore action for Data Link Layer Synthesis

The uncoded packet error rate $\mathsf{PE}_{\ell}^{\text{non-Code}}$ over a link ℓ in Eq. (3.6) is given by,

$$\mathsf{PE}_{\ell}^{\text{non-Code}} = \left[1 - \left(\, 1 - \mathsf{BE}_{\ell}^{\text{DL}} \,\right)^{\mathsf{N}^{\text{bits}}}\right] \cdot\left(\, 1 - \mathbb{P}_{\ell}^{\text{drop}} \,\right) \tag{3.7}$$

where N^{bits} is obtained by network layer synthesis calculation specified in Sec. 3.8

Using Eq. (3.6), $\mathsf{PE}^{\text{m-Hops}}$ is defined as a function of PE over link ℓ given by,

$$\mathsf{PE}^{\text{m-Hops}} = 1 - \prod_{\ell=1}^{\text{m Hops}} \left(1 - \mathsf{PE}_{\ell}^{\text{overall}}\right) \tag{3.8}$$

From Eq. (3.6) we can infer that higher magnitude values of $\mathsf{N}_{\text{FEC}}^{\text{bits}}$ and $\mathsf{N}_{\ell}^{\text{max}}$ lead to (i) a lower transmission data rate or larger link delay, (ii) a lower PER, and (iii) a higher total energy consumption. The impact of the error control parameters interplays with those made by adjusting the transmission power and the modulation scheme at the physical layer. Finally,

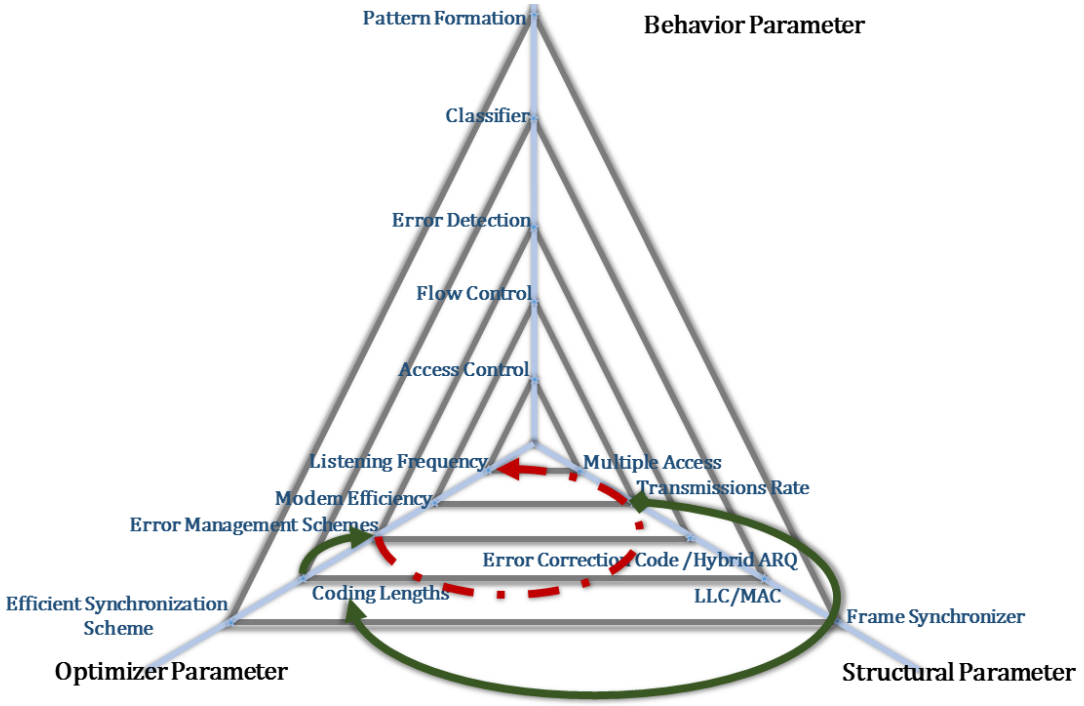


Figure 3.9: Parameter synthesizing final action for Data Link Layer Synthesis

the point-to-point packet error rate constricted by the threshold $\mathsf{PE}^{\mathtt{Th}}$ dependent on $\mathsf{PE}^{\text{m-Hops}}$, and $\mathsf{PE}^{\mathtt{Net}}$ (PE over the Internet) is defined by,

$$\mathsf{PE}^{\mathtt{p2p}} = 1 - (1 - \mathsf{PE}^{\text{m-Hops}})(1 - \mathsf{PE}^{\mathtt{Net}}) \;\leq\; \mathsf{PE}^{\mathtt{Th}} \tag{3.9}$$

3.7.2 Medium Access Control

We consider a variation of Sleep MAC (SMAC) [30] and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), for mainly two reasons. First, idle listening is a major source of energy consumption in low traffic applications such as those expected in the IoT, and many of the physical objects in the IoT may have very limited energy storage. Therefore, we adopt the idea of SMAC in which things periodically listen and sleep [30]. For example, almost, 90% of the total energy consumed during idle listening can be saved when things use an awake/sleep duty cycle of 10%. However, by decreasing the duty cycle, the chances for

things to be connected decrease too and the link delay increases. Second, the interference among the things in one GAP domain or in close GAP domains affects the SNR and thus degrades the BER and the PER. In the CSMA/CA, a node attempts to reserve the channel after it sees the medium idle for an Inter-Frame Space (IFS) amount of time. If the node fails to reserve the medium, it switches to sleep mode to save energy and waits for the next listening cycle. This medium access method can eliminate the interference drastically if the carrier sensing is properly performed. As a result, the hybrid of SMAC and CSMA/CA medium access protocol can save the energy as well as reduce the interference among the things.

In our framework, the duration of the listen and sleep cycles ($\mathcal{T}^{\texttt{Listen}}$,$\mathcal{T}^{\texttt{Sleep}} = 9 \times \mathcal{T}^{\texttt{Listen}}$ for a 10% duty cycle) are adaptive to the QoS requirements and they are set the same for all nodes in one GAP domain. The longer the sleep duration is, the lower the idle energy consumption, but the longer point-to-point delay. This duration parameter in the MAC protocol is taken into account in our cross-layer framework to interplay with the physical layer parameters as well as Hybrid ARQ parameters.

3.8 Network Layer Synthesis

We discuss the impact of the packet size on the different layers. During the stage of network layer, the things of IoT can have variations based on the Behavioral, structural and Optimizer parameters depicted in synthesizer Figure 3.3. We analyze the addressing and information routing on the network performance, as well as, their interrelations with other layers and the limitations imposed by the things capabilities. The initiation of parameter level parsing action is depicted in network layer delta diagram synthesizer of Figures 3.10.

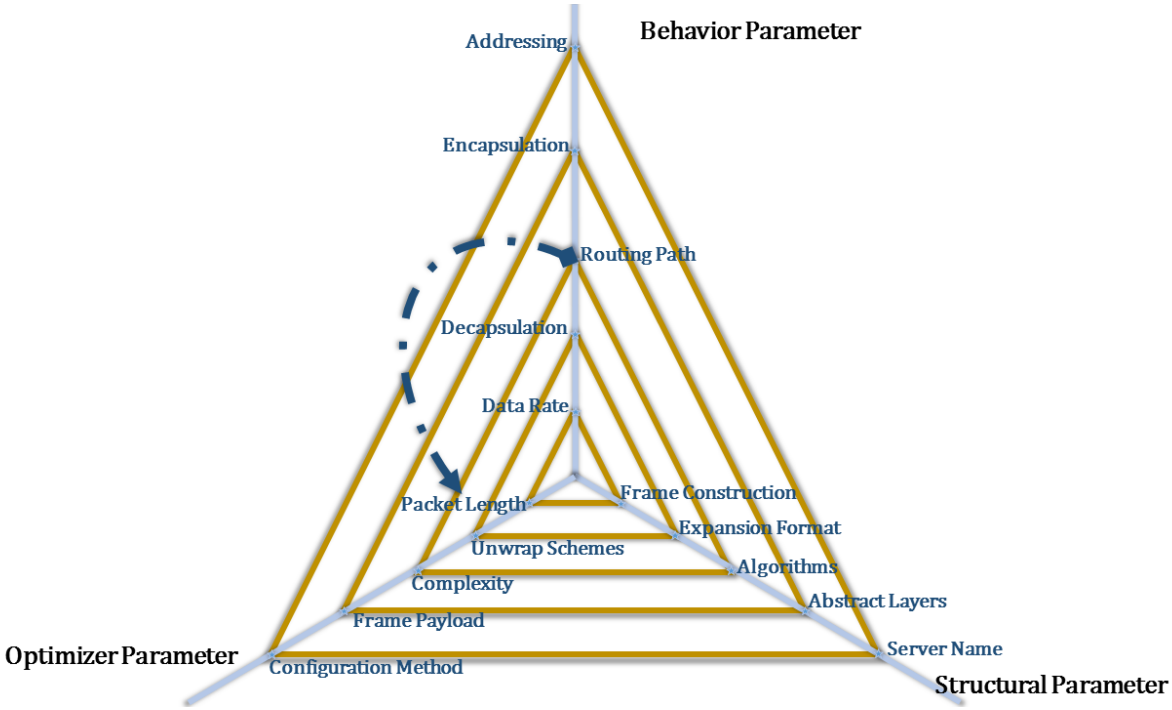


Figure 3.10: Parameter synthesizing begin action on Network Layer

3.8.1 Addressing of things and Routing

Since a huge amount of things are part of the IoT architecture depicted in Fig. 3.5, it is imperative to use IPv6 addressing for the IoT. IPv6 addresses are expressed by 128 bits, which allow the definition of 10^{38} unique addresses (these are expectedly enough for the time being). However, IPv6 addresses are only used for inter-GAP communications, while much shorter local addresses are used in intra-GAP communications.

Our model for routing mechanism uses destination-based routing mechanism [67] where the GAP selects the point-to-point route and the configuration parameters for each data link in a centralized manner resulting from cross layer modeling.

3.8.2 Packet Size Impact

In our model, a fixed packet size N^{bits} is selected and used for all the data links ℓ throughout a given path. A larger packet size results in a reduced point-to-point delay by saving the handshake time $\mathcal{T}^{\text{hshake}}$, the acknowledgement time $\mathcal{T}^{\text{ack}}$, and the queuing time $\mathcal{T}^{\text{Queue}}$ that are required for each packet. Additionally, the reduction of the total number of packets to be sent has an impact on the total energy consumption, while at the same time, transmitting more bits of information in a packet affects the PER. All these interrelations are incorporated in our cross-layer framework.

$\mathsf{N}^{\text{bits}} = \mathsf{N}^{\text{bits}}_{\text{Header}} + \mathsf{N}^{\text{bits}}_{\text{FEC}} + \mathsf{N}^{\text{bits}}_{\text{Data}}$ where N^{bits} is packet size containing header size, data length, and FEC redundancy length, $\mathsf{N}^{\text{DL}}_{\ell} = (1 - \mathsf{PE}^{\text{non-Code}}_{\ell})^{-1}$ is the upper bound for the total number of transmissions of a packet with correct decoding over link ℓ. We can now describe the overall energy consumption and timing delays (for complete point-to-point path of the IoT network) having constraints $\xi^{\texttt{Th}}$ and $\mathcal{T}^{\texttt{Th}}$ in evaluating the Fitness function in Eq. (3.1) as follows.

$$\begin{aligned} {}^{k}\xi &= \mathsf{N}^{\text{bits}} \cdot \xi^{k}_{b} \\ \iff \xi^{k} &\leq \xi^{\texttt{Th}}_{k} \end{aligned} \tag{3.10}$$

where energy on the k$^{\text{Th}}$ node is ${}^{k}\xi$ resulting from the product of packet size N^{bits} and the energy per bit ξ^{k}_{b} calculated by,

$$\begin{aligned} \xi^{k}_{b} &= 2\xi^{\text{Eqp}}_{b} + \frac{\mathsf{P}_{\text{Tx}}}{\Re_{k}} \\ \iff \xi^{\text{Eqp}}_{b} &= \xi^{\text{Eqp}}_{b-Tx} = \xi^{\text{Eqp}}_{b-Rx} \text{ J/bit} \end{aligned} \tag{3.11}$$

where ξ^{Eqp}_{b} is the energy to transmit one bit independent of the distance involved and ξ^{Eqp}_{b-Tx}, ξ^{Eqp}_{b-Rx} constitute energy for transmitter and receiver device equipment.

Let $\mathcal{T}^{\texttt{p2p}}$ be the point-to-point time duration which also includes the Internet delay $\mathcal{T}^{\texttt{Net}}$ constricted by threshold $\mathcal{T}^{\texttt{Th}}$ for inter-GAP communications and the link queuing delay

$\mathcal{T}^{\text{Queue}}$. Factors such as current traffic, behavior of other nodes in IoT or its hardware status etc., determine the queuing delay and Internet delay. Thus $\mathcal{T}^{\texttt{p2p}}$ is defined as,

$$\begin{aligned} \mathcal{T}^{\texttt{p2p}} &= \sum_{\ell=1}^{\text{m Hops}} \left(\mathcal{T}_\ell^{\text{Queue}} + \mathcal{T}_\ell + \mathcal{T}_\ell^{\texttt{Net}} \right) \\ \iff \mathcal{T}_\ell &\leq (\mathsf{N}_\ell^{\text{DL}} - 1) \cdot (\mathcal{T}_\ell^{\text{hshake}} + \mathcal{T}_\ell^{\texttt{Data}} + \mathcal{T}_\ell^{\texttt{Time-out}}) \\ &+ \mathcal{T}^{\texttt{Sleep}} + \mathcal{T}^{\texttt{Signal}} \left(\mathcal{T}^{\text{hshake}} + \mathcal{T}^{\texttt{Data}} + \mathcal{T}^{\text{ack}} \right) \end{aligned} \tag{3.12}$$

Using Central Limit Theorem [68], the complete point-to-point delay $\mathcal{T}^{\texttt{p2p}}$ can be modeled as a Gaussian distributed random variable with variance given below,

$$\begin{aligned} &\mathsf{variance} = \texttt{var}(\mathcal{T}^{Net+Queue}) \text{ where} \\ &\mathcal{T}^{Net+Queue} = \sum_{\ell=1}^{\text{m Hops}} \mathcal{T}_\ell^{\text{Queue}} + \mathcal{T}^{\texttt{Net}} \end{aligned} \tag{3.13}$$

The target probability t_γ is obtained by Chebyshev's inequality[69] to decompose point-to-point delay constraint into,

$$\begin{aligned} &\mathbb{P}\left(\mathcal{T}^{\texttt{p2p}} \geq \mathcal{T}^{\texttt{Th}} \right) \\ &\leq \frac{\texttt{var}(\mathcal{T}^{Net+Queue})}{\texttt{var}(\mathcal{T}^{Net+Queue}) + (\mathcal{T}^{\texttt{Th}} - \sum_{\ell=1}^{\text{m Hops}} \mathcal{T}_\ell - \overline{\mathcal{T}^{Net+Queue}})^2} \\ &\leq 1 - t_\gamma \\ &\iff \mathbb{P}(\mathcal{T}^{\texttt{p2p}} \leq \mathcal{T}^{\texttt{Th}}) \geq t_\gamma \text{ satisfying} \\ &(\mathcal{T}^{\texttt{Th}} - \sum_{\ell=1}^{\text{m Hops}} \mathcal{T}_\ell - \overline{\mathcal{T}^{Net+Queue}}) > 0 \end{aligned} \tag{3.14}$$

The complete point-to-point throughput of IoT architecture is inversely proportional to the point-to-point delay ${}^{p2p}\mathcal{T}$ given by,

$$\rho^{\mathrm{p2p}} = \frac{\mathsf{N}^{\mathrm{bits}}_{\mathrm{Data}}}{\mathcal{T}^{\mathtt{p2p}}} \geq \rho^{\mathrm{Th}} \tag{3.15}$$

So far, we have explored the interrelations among the parameters at the physical layer (Section 3.6), the link layer (Section 3.7) and the network layer in Section 3.8. These parameters including the transmission power, modulation type, the FEC length, the number of retransmissions, the listen duration, the packet size and their interactions, are all optimized in our cross-layer framework, as described in the following section.

3.9 GWO Cross-layer Optimizer framework

3.9.1 Background on Grey Wolf Optimizer

We adopt the GWO algorithm framework discussed in Chapter. 2 Sec. 2.6.2 and apply for extracting a minimized triple specified in Eq. (3.1) by synthesizing the parameters discussed in delta diagram Figures 3.6, 3.10 and 3.9.

3.9.2 Algorithm Set-up

The GWO-Cross-layer framework records fitness values $\mathcal{F}^{fit}(\overrightarrow{X_i})$ using Eq. (3.1) for each candidate solution $\overrightarrow{X_i}$. The following notations and assumptions are to be conveyed before designing the algorithm for GWO-Cross-layer framework.

- Each and every candidate solution $\overrightarrow{X_i}$ is a triple defined in Eq. (3.1).
- The three most optimal candidate solutions are shortlisted as $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ respectively.

- Every fitness function call is subjected to the QoS constraints of Eqs. (3.2a), (3.2b) and (3.2c).
- The values for $\mathsf{PE}^{\mathtt{opt}}$, $\xi^{\mathtt{opt}}$, $\mathcal{T}^{\mathtt{Opt}}$, $\mathsf{PE}^{\mathtt{Th}}$, $\xi^{\mathtt{Th}}$, $\mathcal{T}^{\mathtt{Th}}$, $\mathsf{N}^{\text{bits}}_{\text{Header}}$, $\mathsf{PE}^{\mathtt{Net}}$, $\mathcal{T}^{\mathtt{Net}}$ are computed off line and not during the framework execution in real time.
- The timing parameters t_γ, $\mathcal{T}^{\text{hshake}}$, $\mathcal{T}^{\mathtt{Data}}$, $\mathcal{T}^{\mathtt{Time-out}}$, $\mathcal{T}^{\text{ack}}$, $\mathcal{T}^{\text{Queue}}$, μ_ℓ and the helper functions $\mathcal{K}_\psi$, $\mathcal{G}_\psi$, $\mathcal{S}_\psi$ are also independent of GWO and will be computed off line.
- The linear weights δ^{ξ}, $\delta^{\mathcal{T}}$, δ^{PE} and link traffic parameters $\Re_\ell$, N^{DL} are global and precomputed values independent of QoS framework.

The outcome of Alg. (2) will be near optimal set of triples which satisfy the minimization requirements of Eq. 3.1 subjected to QoS requirements of Eqs. (3.2a), (3.2b) and (3.2c).

ALGORITHM 2: GWO-Cross-layer Framework

Data: W_n–magnitude of the wolf pack, $Iter_{MAX}$ – total number of iterations more than 0, set of ordered triples $\mathtt{S_{triples}} \leftarrow \{ (\xi^{\mathtt{p2p}}, \mathcal{T}^{\mathtt{p2p}}, \mathsf{PE}^{\mathtt{p2p}}) \}$, user defined constraints specified in Sec 3.9.2

Result: Most Minimum Triple:$(\xi^{\mathtt{p2p}}, \mathcal{T}^{\mathtt{p2p}}, \mathsf{PE}^{\mathtt{p2p}}) \leftarrow \mathcal{F}^{fit}(\overrightarrow{X_i})$, Optimal Grey wolf position within given Cross-layer Architecture.

begin
  Randomly initialize W_n number of candidate $\overrightarrow{X_i}$ Grey wolves
  Using Eq. (4.31) identify the best three triples as X_α, X_β and X_δ wolves
  Set `t`:=0
  repeat
    foreach $\overrightarrow{X_i} \in \mathcal{S}_{triples}$ **do**
      Update $\overrightarrow{X_i}$ using Eq. (2.16)
    end
    Compute Eqs. (2.23, 2.15, 2.14) to update a, A, C
    forall *Wolves* $\overrightarrow{X_i}$ *of* $\mathcal{S}_{triples}$ **do**
      `newPositions` ← get positions using Eq. (2.10)
      if *newPositions* $\notin$ *Range* **then**
        `newPositions` ← get positions using Eq. (2.11)
      end
    end
    foreach $\overrightarrow{X_i} \in \mathcal{S}_{triples}$ **do**
      Deduce `Minimum-Triple` using fitness function $\mathcal{F}^{fit}(\overrightarrow{X_i})$ subject to QoS constraints of Eq. (3.2)
    end
    Rank α, β, and δ positions as premier 3 best solutions based on `Minimum Triples` from Previous Step.
    $t \leftarrow t + 1$
  until $t \leq Iter_{MAX}$
  Choose `Optimal Grey Wolf` position given by Eq. (3.1)
end

3.10 Real-time Protocol Testing

The intel-LEACH protocol [49] was incorporated to test our framework from synthesis to optimization steps in practical point-to-point IoT scenarios. The operation is briefly listed in following phases.

1. Transmission Phase
    - Check route validity from point to point nodes and following initialization routines of MAC operation described in Section 3.7.
    - Failure mitigation by generating Route Request (RR) packet containing the destination thing ID directed towards nearest GAP.
2. Service Phase
    - GAP transmits its ID in broadcast mode periodically
    - GAP has sufficiently large power to directly communicate with every thing in its domain (Section 3.4).
    - The things register themselves to GAP with Network Association (NAS) packet.
3. Messaging Phase
    - Upon receiving packet data, the thing sends a Route Acknowledge (RA) packet to the previous hop in the route to show its alive.
    - The above process is repeated in multi-hops scenario until source is reached.
    - Data is transmitted by following the optimal route with the chosen communication parameters defined by delta diagram in Figure 3.9 and according to description in Sec. 3.8.
    - Computation complexity is shifted from things to GAP which reduces multi-flow problems.

4. Routing Phase

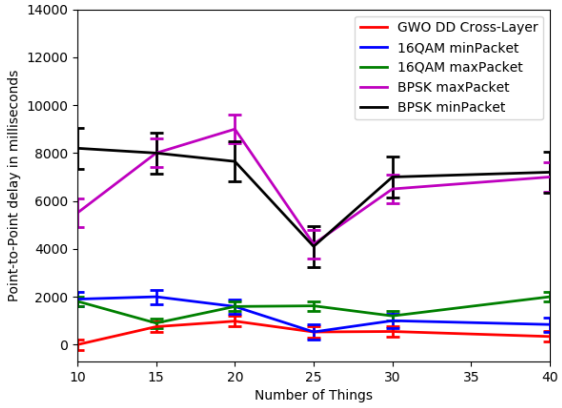
    - GAP receives the RR packet via several paths, the intermediate nodes are earmarked as priority candidates for data transmission
    - GAP initiates the GWO-Cross-layer framework for potential paths and QoS requirements to find the near optimal path and the associated communication parameters, as explained in Sec. 3.9.

The performance of the GWO-Cross-layer framework and traditional layered solutions was compared and analyzed by setting the network with following random variables for testing cross-layer functionalities.

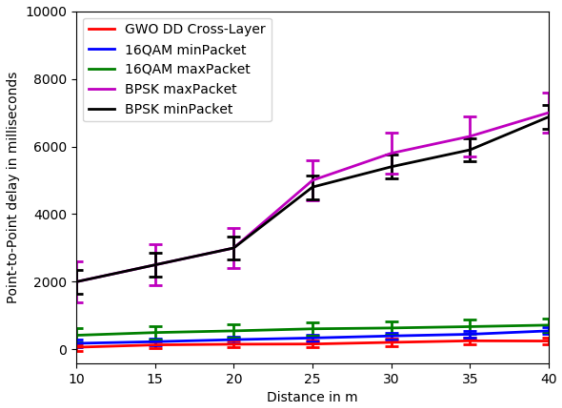
- Additive white Gaussian noise (AWGN) at each link ℓ as $\mathsf{N}_\ell \sim \mathcal{N}(\mu_{\text{noise}}, \mathbb{P}^{\text{noise}})$ where $\mu_{\text{noise}} = 0$, $10\log_{10}\mathbb{P}^{\text{noise}} = -86dB$
- Queuing Delay for things at each link ℓ as $\mathcal{T}_\ell^{\text{Queue}} \sim \mathcal{N}(10, 10^4)$ ms
- Delay due to Internet as $\mathcal{T}^{\texttt{Net}} \sim \mathcal{N}(10^2, 10^4)$ ms
- Packet drop-out rate at every link ℓ as $\mathbb{P}_\ell^{\text{drop}} \sim \mathcal{U}(0, 10^{-1})$
- Packet Error rate of internet as $\mathsf{PE}^{\texttt{Net}} \sim \mathcal{U}(0, 10^{-4})$

The traditional layered approach has each layer autonomously optimized based on Dynamic programming approach as described in [70, 71]. We consider the results only when the QoS is focused on either point-to-point delay minimization, energy consumption minimization, or a linear combination of both, while the threshold for PE is constricted to be less than $\mathsf{PE}^{\texttt{Th}} = 10^4$. However, for simulation purposes, the total amount of data to transmit per transmission is set to 105 bits and the possible packet sizes N^{bits} are 200, 500 or 2000 bits. It is important to note that for every transmission, the thing randomly selects its destination and there is an equal probability that links are either inter-GAP or intra-GAP. The things are randomly assigned diverse hardware capabilities, in terms of computing, memory, energy

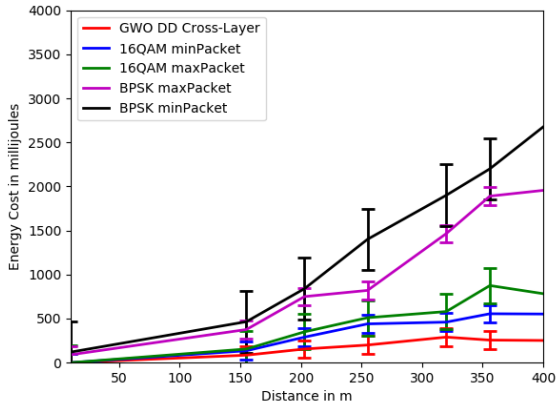
storage, power and communication. Consequently the link rate of things ranges among [250, 10^3 , 10^4] kbps and the power of things varies among [10, 30, 50, 80, 100] milliwatts.
Notice that the error bars in Figures. 3.11a, 3.11b and 3.11c represent the uncertainty interval at the 95% confidence level. The four layered solutions that are plotted for comparison differ in the modulation scheme and the packet size, and make use of the shortest path routing. The Figure 3.11a illustrates point-to-point delay in ms and the energy consumption in mJs (micros Joules) plotted against the number of things in an IoT network. The distance between the transmitter and the receiver is set to 40 m. Our GWO Delta diagram based cross-layer approach has at least 10% gain over other layered solutions. Consequently, we can observe that neither the point-to-point delay nor the energy consumption increases as the number of things increases. This happens when higher node density creates more optional paths for transmission, and contrastingly the point-to-point consumption does not have proportional relationship with it. The 95% confidence interval implies that our GWO Delta digram Cross-layer solution stabilizes than other layered solutions, and the point-to-point performance does not significantly deviate although the computation complexity at the GAP increases.
Another scenario was observed that by increasing the number of hops in the path implied the rise of the point-to-point delay since there are additional handshake, processing and queuing delay introduced into the transmission and the trend is seen in $\mathcal{T}^{\mathtt{p2p}}$ increases as distance increases (Fig. 3.11b). In addition, the energy consumed for the longer distance and by the additional nodes increase the overall $\xi^{\mathtt{p2p}}$ (Figure 3.11c). It can be inferred from Fig. 3.11d that the energy consumed per bit (Eq. 3.10,Eq. 3.11) is directly proportional to the distance between point to point and also evident that energy consumed by GWO Delta diagram cross-layer model is lower than referenced methods. Also PE improves significantly with increased SNR for our model compared to other benchmarks as depicted in Fig. 3.11e. The transmission power is very much lower compared to other layered approaches involving the distance covered as shown in Fig. 3.11f.



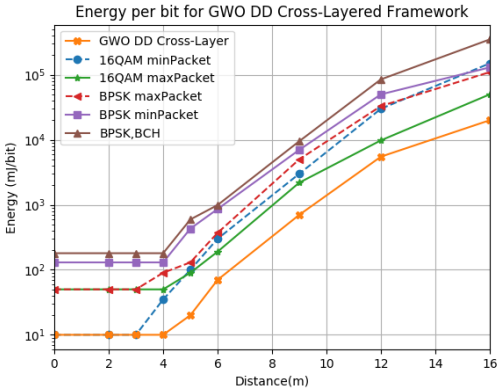
(a) Delay vs Things



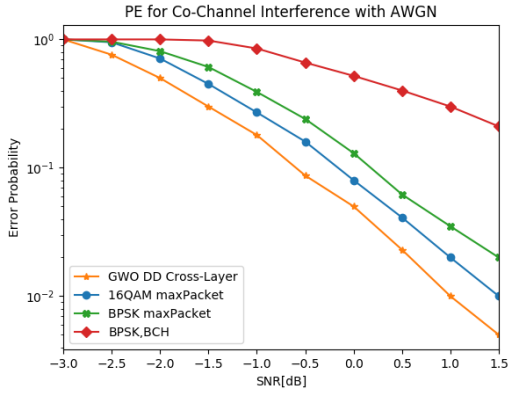
(b) Delay vs Distance



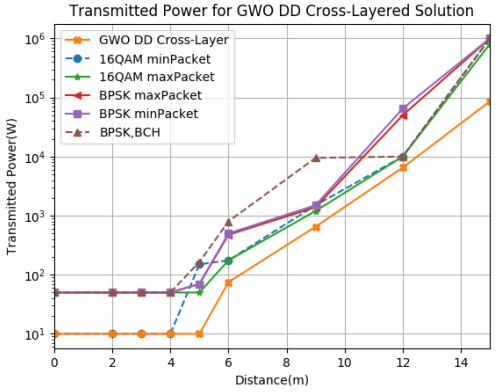
(c) Energy vs Things



(d) Bitwise Energy ξ_b vs Distance



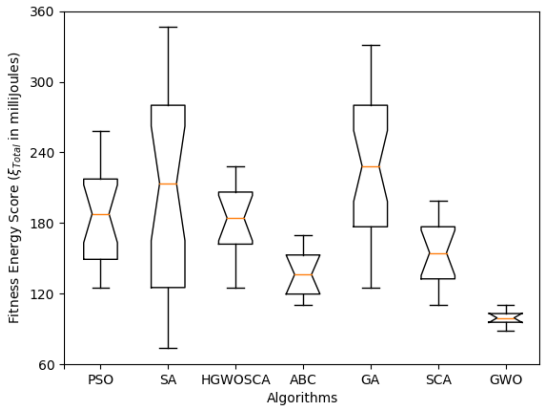
(e) Error Probability vs SNR



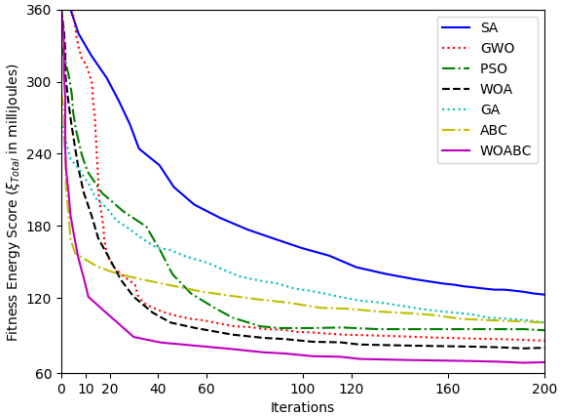
(f) Transmitted Power P^{Tx} vs Distance

Figure 3.11: GWO Delta diagram Synthesized Cross-layer solution compared with traditional layered solutions

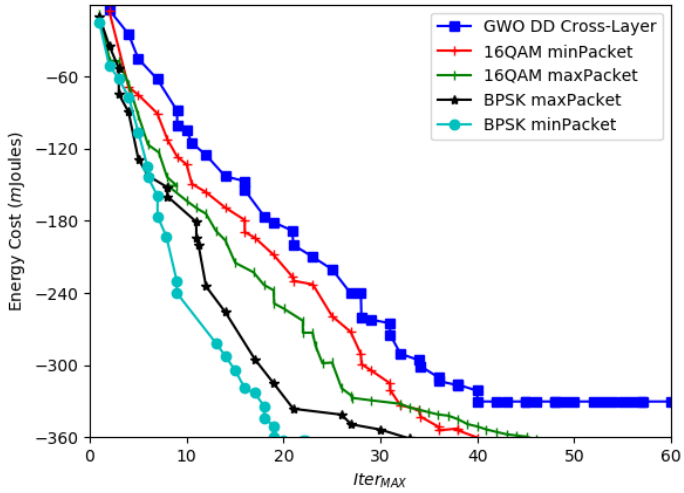
Energy Analysis: A critical cost analysis for energy benchmarking GWO-Crosslayer Framework with traditional crosslayer models having different modulation levels is depicted in Fig. 3.12c. It can clearly interpreted from Fig. 3.12c that our model based on Delta diagram outlasts all the other models in terms of network longevity and energy conservation. The comparison of energy dissipation among various optimization algorithms is shown in Fig. 3.12a. We begin the study of evolutionary algorithms by applying the multi-objective optimization (MOO) for energy factor described in fitness function of Eq. 3.1. Initially we choose MOO algorithms such as Simulated Annealing(SA) algorithm and the Genetic Algorithms(GA) for our model. However, SA algorithm suffers from extreme slowness thereby searching for an optimal solution is not very efficient and feasible. Thus as depicted in Fig 3.12a, the curve for SA predicates worst performance among all other comparative algorithms for energy model in Eq. 3.1. Although relatively the Genetic Algorithm (GA) performs better than SA, it encounters issues about not only termination time, but also with the convergence rate. This is due to the fact that GA has a major drawback of getting stuck in local minima, making it unsuitable for multi-objective based optimization problems. Particle Swarm Optimization (PSO) overcomes the above challenges faced by SA and GA, however it still falls short with problems related to high dimensional space. Furthermore, Hybrid Grey Wolf Optimizer Sine Cosine Algorithm (HGWOSCA) [72], Sine Cosine Algorithm (SCA) [73] and Artificial Bee Colony (ABC) [35] methods overcome most of the problems faced by other algorithms described earlier. But the GWO based framework described in Alg. 3 performs optimally for our the specific context and able to find the best possible fitness energy scores with least number of iterations. The Fig. 3.12b show that the box-plot of GWO is significantly lower and narrower than other algorithms. Fig. 15 shows that the box-plot of SA is super narrow implying worst suitability for our model; while GWO is under the minima of other algorithms. This means that GWO tends to find the global minimum and significantly outperforms other algorithms.



(a) Comparative tests of algorithms for Delta Diagram Synthesizer using Box-plot



(b) Comparative tests of algorithms for achieving convergence for our Delta Diagram model.



(c) Network Longevity Analysis

Figure 3.12: Network Lifetime and Convergence Scenario Analysis

3.11 Conclusion

In this Chapter, we presented the novel cross-layer design of the IoT protocol stack, which ranges bottom-up from the Physical to the MAC layer. The approach uses a Delta-diagram based synthesizer to identify the parameters to consider for optimization. The parameters chosen are completely dependent on the type of network and topology of IoT setup and

hence it is very flexible. Furthermore, synthesis process enables parsing differing levels of parameters and moving across different domains of Behavior, Structural and Optimizer requirements of the chosen protocol for IoT.

We constructed a model for deriving the fitness function which simultaneously minimizes Energy, Timing delays and Packet Error rate requirements of the IoT network protocol subjected to the QoS constraints.

We introduced a modified Grey Wolf Optimizer algorithm to optimize and search the near optimal minimized triple resulting from model search space. The results and analysis of the methodology shows that our approach using a test protocol outperforms other cross-layered approaches significantly and is highly flexible in nature so that it can adapted to any type QoS requirements of different IoT network protocols.

Chapter 4

HLS for Resource Constrained IoT Hardware

In this chapter a mathematical model to arrive at an optimal arrangement of resource utilizations within DSE is proposed. The proposed model resolves conflicting objectives of power-performance as well as area occupied by different types of critical hardware resources. We resolve for arriving at this type of arrangement with minimum completion time among all of the possible permuted arrangements. The problem is formulated in term of multi-objectives as in the case of DSE for ensuring reliable power-performance-area trade-off for application specific processors. For resolving such a trade–off problem many heuristic algorithms are considered a natural fit to derive an optimal solution in DSE procedure. Consequently, in this chapter a comprehensive mapping process and reliable solution evaluation approach called Grey Wolf Optimizer Driven DSE (GWO-DSE) is proposed with several advantages over other heuristics. The contributions of the chapter are outlined as: i) Introduction of a novel GWO-DSE methodology for power-performance-area trade-off ii) Unique solution evaluation methodology called Utility Coefficients and Utility Ranking iii) Novel fitness function evaluation technique for GWO iv) Comparative benchmarks using digital

filter transfer functions v) Substantial improvement in solution arrival within search space (average $>$ 51%) and reduction in exploration time ($>$ 89%) when compared to recent DSE approaches for the tested benchmarks.

4.1 Background

High-level synthesis (HLS), also referred to as C synthesis, electronic system-level (ESL) synthesis, algorithmic synthesis, or behavioral synthesis, comprises of an automated design process that interprets an algorithmic description of a desired behavior and creates digital hardware that realizes that behavior[74]. Synthesis begins with a high-level specification of the problem, where behavior is generally decoupled from structure e.g. clock-level timing. Early HLS explored a variety of input specification languages[75], although recent research and commercial applications generally accept synthesizable subsets of ANSI C/C++/SystemC/MATLAB. The code is analyzed, architecturally constrained, and scheduled to create a register-transfer level (RTL) hardware description language (HDL), which is then in turn commonly synthesized to the gate level by using a logic synthesis tool. The goal of HLS is to let hardware designers efficiently build and verify hardware, by giving them better control over optimization of their design architecture, and through the nature of allowing the designer to describe the design at a higher level of abstraction while the tool does the RTL implementation. An algorithmic description specifies the inputs and outputs of the behavior of the algorithm in terms of operations to be preformed and data flow[75]. A description of the algorithm is usually represented in the form of an acyclic directed graph known as a sequencing graph[76]. These graphs specify the input/output relation of the algorithm and the data dependency present in the data flow. The graph is defined in terms of its vertices and edges, wherein the vertices signify the operations and the edges indicate the data dependency present in the function. Hence HLS is a conversion from the abstract behavioral description to its respective hardware description in the form of memory ele-

ments, storage units, multiplexers/de-multiplexers and the necessary interconnections. The transformed algorithm at the RT level is comprised of a data path unit and the control unit[77]. Multi-Objective Optimization techniques for very-large scale integrated (VLSI) is an area of multiple criteria decision making. Multi-objective VLSI designs are concerned with mathematical optimization problems involving more than one objective function such as cost of solving the scheduling, allocation, module selection since an exhaustive search has a prohibitive complexity. Techniques based on Multi-Objective Optimization are employed both in low end Application Specific Integrated Circuits (ASICs) with low power dissipation and acceptable performance and in high end ASICs with high performance requirements and optimal power expenditure. Therefore, it warrants efficient design space exploration (DSE) techniques to determine optimal usage of time as a consequence of time-to-market circumstance[78]. Design space exploration (DSE) refers to the activity of exploring design alternatives prior to implementation. The power to operate on the space of potential design candidates renders DSE useful for many engineering tasks, including rapid prototyping, optimization, and system integration. The main challenge in DSE arises from the sheer size of the design space that must be explored. Typically, a large system has millions, if not billions, of possibilities, and so enumerating every point in the design space is prohibitive[79, 80]. Typically DSE refers to the activity of discovering and evaluating design alternatives during system development using multi-objective techniques. The usage of DSE includes but not limited to (1) Rapid prototyping-to generate a set of prototypes prior to implementation. Simulating and profiling of these prototypes can increase understanding of the impact of design decisions while taking complex system dynamics into account. (2) Optimization-When metrics are available for comparing one design to another, DSE can be used to perform optimization, eliminating inferior designs and collecting a set of final candidates that are further studied.(3) System integration-requires the assembly and configuration of multiple components into a working whole. DSE can be used to find legal assemblies and configurations that satisfy a set of global design constraints[79, 81, 82].

4.2 Related Work

The area optimization, delay and power in behavioral synthesis was achieved in [83] albeit it does not emphasize high level design space exploration using swarm intelligence techniques. The model described in [33] solves DSE problem using a hybridized technique of GA and weighted sum particle swarm optimization (WSPSO). The WSPSO solved the DSE problem by performing a crossover between current position with global best position and local best position similar to GA while not considering velocity to update the position. However, WSPSO approach does not utilize multiple user constraints for power, execution time and area in cost function. Particle Swarm Optimization (PSO) is also a heuristic search methodology similar to Grey Wolf Optimizer (GWO)[3] algorithm which emulates the journey of a flock of birds aiming at finding food[84]. However, the performance of heuristic GWO-based algorithm[3] is far superior to well-known evolutionary trainers like GA, PSO, and Differential Evolution (DE). From our best knowledge, this chapter is the first to consider a IoT approach for parameter estimation in DSE based on the resource classification of design space.

4.3 Hypotheses and Problem Definition

In this section several definitions are postulated that will help to understand the flow of the design problem. A novel approach for establishing a design search space with defined boundaries and content format will eventually be discussed in this section to shortlist the ideal candidate micro-architecture from a collection of architecture design tuples. Therefore the proposed approach serves as a cornerstone for rapid high level synthesis design flow. Hypothetically the process of DSE is a protracted task for the designer since the exploration is a long-drawn-out procedure[76]. In order to arrive at an optimum design configuration, the DSE process warrants tremendous accuracy with extensive analysis. Ideally, it is advantageous to arrive at a best variant within large design space in shortest possible time and

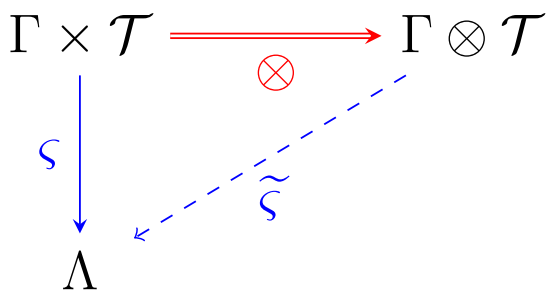


Figure 4.1: Universal Property for Tensor Product

having least complexity. Consequently, the impact a resource will have on each optimized parameter during its change must be determined and the following theory forms the basis to explain a real world example via HLS.

4.3.1 Execution Time Analysis

The Def. 1 introduces the term 'Reliability-Aware Workload' ($\mathcal{R}_{\mathtt{aw}}$) as described below.

Definition 1. The $\mathcal{R}_{\mathtt{aw}}$ of an i^{th} resource γ_i^j of type j is defined as time expended (or clock cycles processed) by a resource to complete its assigned operation during the process of scheduling. The type j of the resource γ_i^j represents the category to which it belongs among m categories e.g. adder, subtracter, multiplexer, alu, etc. and n number of total resources where $1 \leq i \leq n$, $1 \leq j \leq m$. ■

The maximum resources in a category is given by Def. 2 which follows from Def. 1.

Definition 2. Let η^j denote the maximum number of resources for any category j. Then η^j can be defined using a maximum function f for any given i^{th} resource as follows:

$$\eta^j := \max_{\gamma_i^j \in \Omega \subseteq \Gamma} f(\gamma_i^j) = \{h \mid \gamma_i^j \in \Omega \wedge \forall\ \kappa_h \in \Omega : f(\kappa_h) \leq f(\gamma_i^j)\} \tag{4.1}$$

■

The design space can be formulated as a plurality of vectors that represents total available resources as $\Gamma = \left[\ \gamma_1^1\ ,\ \gamma_2^1\ \ ,\ldots,\ \gamma_{\eta^1}^1,\ \gamma_1^2\ ,\ \gamma_2^2,\ \gamma_{\eta^j}^2\ \ ,\ldots,\ \gamma_1^m\ ,\ \gamma_2^m\ \ ,\ldots,\ \gamma_{n-1}^m,\ \gamma_{\eta^m}^m\ \right]$ and the number of clock cycles for respective resource as $\mathcal{T} = \left[\ \tau_1^1\ ,\ \tau_2^1\ \ ,\ldots,\ \tau_{\eta^1}^1,\ \tau_1^2\ ,\ \tau_2^2,\ \tau_{\eta^j}^2\ \ ,\ldots,\ \tau_1^m\ ,\ \tau_2^m\ \ ,\ldots,\ \tau_{n-1}^m,\ \tau_{\eta^m}^m\ \right]^T$. A *tensor product* of Γ and $\mathcal{T}$ homomorphically mapping to Λ is described in the Def. 3 below.

Definition 3. [85] Let Γ and $\mathcal{T}$ be two vector spaces in a given domain of design space. The tensor product of Γ and $\mathcal{T}$ denoted by $\Gamma \otimes \mathcal{T}$ is a vector space with a bilinear map

$$\otimes\ :\ \Gamma \times \mathcal{T} \rightarrow \Gamma \otimes \mathcal{T} \tag{4.2}$$

which has the universal property as described in Figure. 4.1. Iff $\varsigma\ :\ \Gamma \times \mathcal{T} \rightarrow \Lambda$, there exists a unique linear map up to isomorphism, $\widetilde{\varsigma}\ :\ \Gamma \otimes \mathcal{T} \mapsto \Lambda$ such that $\otimes\ \circ\ \widetilde{\varsigma} = \varsigma$ ■

Consequently $\mathcal{R}_{\texttt{aw}}$ from can be defined in terms of all available resources for finishing their respective operations by multiplying the *tensor product* Λ with ϱ_{data} – a constant which specifies the iterations an entire set of data elements needs to be processed and ϱ_t–the time period of the clock defined below.

$$\mathcal{R}_{\texttt{aw}}\ := \Lambda \cdot \varrho_{data} \cdot \varrho_t \tag{4.3}$$

It is observed that there are simultaneous infinitesimal changes $\delta\gamma$, $\delta\tau$ and $\delta\varrho_t$ to resources, resource clock cycles, and total period respectively during the exploration process. As a result, we can take a *total differential*[86] based on every category in Eq. (4.3) resulting in Eq. (4.4),

$$d\mathcal{R}_{\texttt{aw}} = \left[\frac{\partial \mathcal{R}_{\texttt{aw}}}{\partial \Gamma} \cdot \Delta\Gamma + \frac{\partial \mathcal{R}_{\texttt{aw}}}{\partial \varrho_t} \cdot \Delta\varrho_t\right] \cdot \varrho_{data} \tag{4.4}$$

The partial derivative applied to the Eq. (4.3) with respect to Γ is evaluated individually as follows,

$$\frac{\partial d\mathcal{R}_{\texttt{aw}}}{\partial \Gamma} = \frac{\partial\left(\Gamma \otimes \mathcal{T} \cdot \varrho_{data} \cdot \varrho_t\right)}{\partial \Gamma} \tag{4.5}$$

Proposition 1. *The number of clock cycles τ_i^j for an i^{th} resource γ_i^j is a constant value for j^{th} category if resources are monotonous[87] i.e. ${}^{j}\tau_i = 0, \forall j \neq i \implies \tau^j \leftarrow \mathcal{T}, \forall i.$*

As a consequence of Prop. (1), the tensor product in Eq. (4.2) contracts to product of summations[85] and partially differentiates with respect to individual j^{th} category subsets. Thereby, Eq. (4.5) reformulates to,

$$\begin{aligned}
\frac{\partial \mathcal{R}_{\texttt{aw}}}{\partial \Gamma} &\cong \frac{\partial \left(\prod_{j=1}^{m} (\eta^j + \tau^j) \right)}{\partial\ \eta^j} \cdot \varrho_{data} \cdot \varrho_t \\
&= \prod_{j=1}^{m} \left(1 + \frac{\partial \tau^j}{\partial\ \eta^j} \right) \cdot \varrho_{data} \cdot \varrho_t \\
&\quad \text{(by reciprocity transformation} \\
&\quad \text{and getting the dual using [88])} \\
&= \frac{\frac{\partial\ \tau^j}{\partial\ \eta^j}}{1 - \left(1 + \frac{\partial\ \tau^j}{\partial\ \eta^j} \right)^{-m}} \cdot \varrho_{data} \cdot \varrho_t \\
&= \frac{\frac{\partial\ \tau^j}{\partial\ \eta^j}}{1 - \zeta_r} \cdot \varrho_{data} \cdot \varrho_t \\
&\quad \text{where } \zeta_r = \frac{1}{\left(1 + \frac{\partial\ \tau^j}{\partial\ \eta^j} \right)^{m}} \\
&\quad \text{is the resource depletion ratio}
\end{aligned} \tag{4.6}$$

It is observed from Eq. (4.6) that resource depletion ratio ζ_r closely mimics present value analysis (PVA) applied to cost-benefit analysis [89] of asset management given by,

$$\begin{aligned}
\frac{\mathbb{P}_V}{\mathbb{F}_V} &= \frac{1}{\left(1 + \frac{\partial\ \tau^j}{\partial\ \eta^j} \right)^{m}} \\
&= \zeta_r
\end{aligned} \tag{4.7}$$

where $\mathbb{P}_V$ is Present Resource Value

and $\mathbb{F}_V$ is Future Resource Value.

From Eq. (4.7) it follows that the resource depletion ratio ζ_r can be equated as the rate of compounded depreciation over m number of resources.
Substituting Eq. (4.7) in Eq. (4.6) we get,

$$\frac{\partial \mathcal{R}_{\mathtt{aw}}}{\partial \Gamma} = \frac{1}{\left(1 - \frac{\mathbb{P}_V}{\mathbb{F}_V}\right)} \cdot \varrho_{data} \cdot \varrho_t \tag{4.8}$$

The *total differential* in Eq. (4.4) can be expressed in terms of Eq. (4.8) as,

$$\begin{aligned} d\mathcal{R}_{\mathtt{aw}} &= \left[\frac{1}{(1-\zeta_r)} \cdot \varrho_{data} \cdot \varrho_t \cdot \Delta\Gamma + \Lambda \cdot \varrho_{data} \cdot \Delta\varrho_t\right] \cdot \varrho_{data} \\ &\quad \text{using Prop. (1),} \\ &= \left[\frac{1}{(1-\zeta_r)} \cdot \varrho_t \cdot \Delta\Gamma + \prod_{j=1}^{m} \left(\eta^j + \tau^j\right) \cdot \Delta\varrho_t\right] \cdot \varrho_{data}^2 \end{aligned} \tag{4.9}$$

The individual terms of Eq. (4.9) can be interpreted as follows,

- Eq. (4.9) denotes that the effective change in $\mathcal{R}_{\mathtt{aw}}$ is a consequence of changes in total number of resources and the clock periodicity (frequency) respectively.

- The coefficient term $\frac{1}{(1-\zeta_r)} \cdot \varrho_t \cdot \Delta\Gamma$ in Eq. (4.9) indicates that rate of increase in $\mathcal{R}_{\mathtt{aw}}$ is proportionate to an increase in j categories and/or i number of resources for that particular resource $\Delta\Gamma$ (e.g. if the number of multiplexers in a specific resource category increases from one to ten, it results in a substantial increase in $\mathcal{R}_{\mathtt{aw}}$ due to resource constraints).

- The coefficient term $\prod_{j=1}^{m} (\eta^j + \tau^j) \cdot \Delta\varrho_t$ in Eq. (4.9) specifies rate of change in $\mathcal{R}_{\mathtt{aw}}$ contributed by the clock period (frequency) changes. Typically, the timing constraints[90]

occur due to propagation delays and net clock frequency changes. The effective correlation of synthesis and place-and-route timing delivers better timing quality-of-results (QoR) for resource clock[90] thus improving $\mathcal{R}_{\texttt{aw}}$.

- Based on the above analysis, we define the term *Utility Coefficient* (UC) during the DSE process of our proposed model. The UC underlines the influence of any given resource in net variability of optimization parameters such as area, time of execution and power consumption. The UC plays a vital role in organizing the architecture design space, which consists of variants arranged in either ascending or descending order of magnitude

Definition 4. The $\mathrm{UC_T}$ for parameter associated with time of execution for any given resource in $\Gamma = \left[\ \gamma_1^1\ ,\ \gamma_2^1\ \ ,\ldots,\ \gamma_{\eta^1}^1,\ \gamma_1^2\ ,\ \gamma_2^2,\ \gamma_{\eta^j}^2\ \ ,\ldots,\ \gamma_1^m\ ,\ \gamma_2^m\ \ ,\ldots,\ \gamma_{n-1}^m,\ \gamma_{\eta^m}^m\ \right]$ is defined as,

$$\begin{aligned}\mathrm{UC_T}(\gamma_i^j) &= \frac{\frac{1}{(1-\zeta_r)}\cdot \varrho_t \cdot \Delta\Gamma}{(\eta^j/\tau^j)} \\ &= \frac{\frac{1}{(1-\zeta_r)}\cdot \varrho_t \cdot \tau^j \cdot \Delta\Gamma}{\eta^j} \end{aligned} \tag{4.10}$$

$$\begin{aligned}\mathrm{UC_T}(\gamma_{clock}) &= \frac{\prod_{j=1}^{m}(\eta^j+\tau^j)\cdot\Delta\varrho_t}{({}^j\eta_{clock}/\tau^j)} \\ &= \frac{\prod_{j=1}^{m}(\eta^j+\tau^j)\cdot\tau^j\cdot\Delta\varrho_t}{{}^j\eta_{clock}} \end{aligned} \tag{4.11}$$

■

The $\mathrm{UC_T}$ defined in Def. 4 determines the rate of change of $\mathcal{R}_{\texttt{aw}}$ with respect to change in number of resources. The constant ϱ_{data} in Eq. (4.9) is ignored in evaluating UC since it does not have any impact on the *Utility Rank* (UR) calculations described in Def. 9 which contributes to design space variant evaluation.

A system with n functional resources of j type has its execution time T_{exe} related to cycle time τ_i^j of each resource i by,

$$T_{exe} = [\ \mathcal{L} + (n-1) \cdot \tau_i^j\ ] \tag{4.12}$$

where $\mathcal{L}$ represents latency of execution and n denotes the total resource count. As the number of resources to be processed is large for real life applications, $\mathcal{L}$ can be ignored while ${}^j\tau_i$ becomes a primary factor in evaluating UC and UR. However, $\mathcal{L}$ will be considered for design flow analysis discussed in Sec. (4.5).

4.3.2 Power Consumption Analysis

The Def. 5 introduces the term 'Reliability-Aware Power'($\mathcal{R}_{\mathtt{ap}}$) as described below.

Definition 5. The $\mathcal{R}_{\mathtt{ap}}$ for a system with n functional resources having m categories is obtained by multiplying ϱ_c–power expended per area unit resource and tensor product (Eq. (4.2)) of the resource $\Gamma = \big[\ \gamma_1^1\ ,\ \gamma_2^1\ \ ,\ldots,\ \gamma_{\eta^1}^1,\ \gamma_1^2\ ,\ \gamma_2^2,\ \gamma_{\eta^j}^2\ \ ,\ldots,\ \gamma_1^m\ ,\ \gamma_2^m\ \ ,\ldots,\ \gamma_{n-1}^m,\ \gamma_{\eta^m}^m\ \big]$ and the respective area $\Xi = \big[\ \xi_1^1\ ,\ \xi_2^1\ \ ,\ldots,\ \xi_{\eta^1}^1,\ \xi_1^2\ ,\ \xi_2^2,\ \xi_{\eta^j}^2\ \ ,\ldots,\ \xi_1^m\ ,\ \xi_2^m\ \ ,\ldots,\ \xi_{n-1}^m,\ \xi_{\eta^m}^m\ \big]^T$ occupied per unit resource at given operating frequency where $1\ \leq\ i\ \leq n$, $1\ \leq j\ \leq m$ defined as,

$$\mathcal{R}_{\mathtt{ap}} = (\Gamma \otimes \Xi) \cdot \varrho_c \tag{4.13}$$

■

Proposition 2. *The area occupied per unit resource ξ_i^j for an i^{th} resource γ_i^j is a fixed for each j^{th} category if resources are monotonous[87] i.e. $\xi_i^j = 0,\ \forall j \neq i\ \implies\ \xi^j \leftarrow \Xi,\ \forall i$.*

As a result of Prop. (2), the tensor product in Eq. (4.13) will be reduced as follows,

$$\mathcal{R}_{\mathtt{ap}} \cong \sum_{j=1}^{m} (\eta^j \cdot \xi^j) \cdot \varrho_c \tag{4.14}$$

Similar to the observation of Eq. (4.4), there will be simultaneous infinitesimal changes $\delta\gamma$, $\delta\xi$ and $\delta\varrho_c$ to resources, unit resource area, and power expended per area unit resource respectively during the exploration process.

By taking a *total differential*[86] based on every category in Eq. (4.14) gives Eq. (4.15) as,

$$d\mathcal{R}_{\mathtt{ap}} = \frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial\ \eta^j} \cdot \Delta\eta^j + \frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial \varrho_c} \cdot \Delta\varrho_c \tag{4.15}$$

The partial derivative applied to the Eq. (4.14) with respect to η^j is evaluated individually as follows,

$$\frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial\ \eta^j} = \frac{\partial}{\partial\ \eta^j}\left[\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) \cdot \varrho_c\right] \tag{4.16}$$

As ϱ_c and η^j are mutually independent.

Hence, $\dfrac{\partial \varrho_c}{\partial\ \eta^j} = 0$;

Applying chain rule [86] to Eq. (4.16),

$$\frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial\ \eta^j} = \left[\sum_{j=1}^{m}(\eta^j \cdot \frac{\partial\ \xi^j}{\partial\ \eta^j} + \xi^j \cdot \frac{\partial\ \eta^j}{\partial\ \eta^j})\right] \cdot \varrho_c$$

Also ξ^j and η^j are mutually independent,

so $\dfrac{\partial\ \xi^j}{\partial\ \eta^j} = 0$;

$$\frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial\ \eta^j} = \left(\sum_{j=1}^{m} \xi^j\right) \cdot \varrho_c$$

$$= \xi^{(1+2+\cdots+m)}$$

m^{th} term in arithmetic series summation 1 + 2 ... + m

$$\implies \frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial\ \eta^j} = \xi^m \cdot \varrho_c \tag{4.17}$$

Similarly, the partial derivative applied to the Eq. (4.14) with respect to η^j is evaluated

individually as follows,

$$\frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial \varrho_c} = \frac{\partial}{\partial \varrho_c}\left[\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) \cdot \varrho_c\right] \tag{4.18}$$

Applying chain rule[86] to Eq. (4.18), it reduces to

$$\frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial \varrho_c} = \frac{\partial}{\partial \varrho_c}\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) \cdot \varrho_c + \frac{\partial \varrho_c}{\partial \varrho_c} \cdot \left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right)$$

$$\text{where } \frac{\partial}{\partial \varrho_c}\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) = 0;$$

$$\implies \frac{\partial \mathcal{R}_{\mathtt{ap}}}{\partial \varrho_c} = \sum_{j=1}^{m} \eta^j \cdot \xi^j \tag{4.19}$$

By using Eqs. (4.17) and (4.19), the total differential in Eq. (4.15) reduces to Eq. (4.20).

$$d\mathcal{R}_{\mathtt{ap}} = \xi^m \cdot \varrho_c \cdot \Delta\eta^j + \left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) \cdot \Delta\varrho_c \tag{4.20}$$

The terms in Eq. (4.20) can be summarized as follows,

- Eq. (4.20) denotes that the effective change in $\mathcal{R}_{\mathtt{ap}}$ due to an alternation in total number of resources and the clock periodicity (frequency) respectively.
- The coefficient term $\xi^m \cdot \varrho_c \cdot \Delta\eta^j$ in Eq. (4.20) indicates that rate of increase in $\mathcal{R}_{\mathtt{ap}}$ is proportionate to an increase in j categories and/or i number of resources for that particular resource $\Delta\Gamma$.
- The coefficient term $\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) \cdot \Delta\varrho_c$ in Eq. (4.20) specifies rate of change in $\mathcal{R}_{\mathtt{ap}}$ contributed by the clock period (frequency) changes.

The factors mentioned above describe the influence on total power consumption when there is a change in number of resources at maximum clock frequency and specified using the Def. 6 of $\mathrm{UC_P}$ for power estimation below.

Definition 6. The *Utility Coefficient* $\mathrm{UC_P}$ for power consumption for any given resource in $\Gamma = \left[\gamma_1^1 \ , \gamma_2^1 \ \ , \ldots , \gamma_{\eta^1}^1, \gamma_1^2 \ , \gamma_2^2, \gamma_{\eta^j}^2 \ \ , \ldots , \gamma_1^m \ , \gamma_2^m \ \ , \ldots , \gamma_{n-1}^m, \gamma_n^m \ \right]$ is defined as,

$$\mathrm{UC_P}(\gamma_i^j) = \frac{\xi^m \cdot \varrho_c \cdot \Delta\eta^j}{\eta^j} \tag{4.21}$$

$$\mathrm{UC_P}(\gamma^{\texttt{Clock}}) = \frac{\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j \right) \cdot \Delta\varrho_c}{\eta^{\texttt{Clock}}} \tag{4.22}$$

■

4.3.3 Area Analysis

The Def. 7 introduces the term total 'Reliable-Area'($\mathcal{R}_{\texttt{area}}$) as described below.

Definition 7. For a system of n functional resources having m categories, the $\mathcal{R}_{\texttt{area}}$ is obtained as a summation of $\mathcal{R}_{\texttt{area}}^{\texttt{clock}}$– area of clock oscillator acting as resource providing the requisite clock frequency to the system, $\mathcal{R}_{\texttt{area}}^{\texttt{mem}}$– the storage elements'footprint in the context of overall area and tensor product (Eq. (4.2)) of the resource $\Gamma = \left[\gamma_1^1 \ , \gamma_2^1 \ \ , \ldots , \gamma_{\eta^1}^1, \gamma_1^2 \ , \gamma_2^2, \gamma_{\eta^j}^2 \ \ , \ldots , \gamma_1^m \ , \gamma_2^m \ \ , \ldots , \gamma_{n-1}^m, \gamma_n^m \ \right]$ and the respective area $\Xi = \left[\xi_1^1 \ , \xi_2^1 \ \ , \ldots , \xi_{\eta^1}^1, \xi_1^2 \ , \xi_2^2, \xi_{\eta^j}^2 \ \ , \ldots , \xi_1^m \ , \xi_2^m \ \ , \ldots , \xi_{n-1}^m, \xi_{\eta^m}^m \ \right]^T$ occupied per unit resource at given operating frequency where $1 \ \leq \ i \ \leq n$, $1 \ \leq j \ \leq m$ defined as,

$$\mathcal{R}_{\texttt{area}} = \mathcal{R}_{\texttt{area}}^{\texttt{clock}} + \mathcal{R}_{\texttt{area}}^{\texttt{mem}} + (\Gamma \otimes \Xi) \tag{4.23}$$

■

Using Prop. (2) in Eq. (4.23), it reduces to,

$$\mathcal{R}_{\texttt{area}} \cong \mathcal{R}_{\texttt{area}}^{\texttt{clock}} + \mathcal{R}_{\texttt{area}}^{\texttt{mem}} + \sum_{j=1}^{m} (\eta^j \cdot \xi^j) \tag{4.24}$$

As discussed before in Eq. (4.4), there will be simultaneous infinitesimal changes $\delta\gamma$, $\delta\xi$, $\delta\ \mathcal{R}_{\texttt{area}}^{\texttt{clock}}$ and $\delta\ \mathcal{R}_{\texttt{area}}^{\texttt{mem}}$ to resources, unit resource area, clock resource and storage resource per area unit resource respectively during the process of exploration.
By taking a *total differential*[86] based on every category in Eq. (4.24) gives Eq. (4.25) as,

$$d\mathcal{R}_{\texttt{area}} = \frac{\partial \mathcal{R}_{\texttt{area}}}{\partial\ \mathcal{R}_{\texttt{area}}^{\texttt{clock}}} \cdot \Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{clock}} + \frac{\partial\ \mathcal{R}_{\texttt{area}}}{\partial\ \mathcal{R}_{\texttt{area}}^{\texttt{mem}}} \cdot \Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{mem}} + \frac{\partial}{\partial\ \eta^j}\left(\sum_{j=1}^{m} \eta^j \cdot \xi^j\right) \cdot \Delta\eta^j \tag{4.25}$$

Reducing the above equation and substituting Eq. (4.17),

$$d\mathcal{R}_{\texttt{area}} = \Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{clock}} + \Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{mem}} + \xi^m \cdot \Delta\eta^j \tag{4.26}$$

Based on Eq. (4.26) the following points can be itemized,

- Eq. (4.26) implies that the rate of change of the total exploration area $\mathcal{R}_{\texttt{area}}$ is proportional to area fluctuations ξ^m of any m^{th} primitive resource(s), clock resources and memory elements.

- The coefficient term $\Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{clock}}$ in Eq. (4.26) indicates that rate of increase in $\mathcal{R}_{\texttt{area}}$ is proportionate to number of clock oscillator resources employed in exploration process.This is one of the factors impacting the total resource space. the hardware resources.

- The coefficient term $\Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{mem}}$ in Eq. (4.26) denotes the contribution of area taken up by memory storage elements (e.g. latches, registers,etc.,) to the overall resource space.

- The coefficient term $\xi^m \cdot \Delta\eta^j$ in Eq. (4.26) indicates that rate of increase in $\mathcal{R}_{\texttt{area}}$ is proportionate to a multiple of an increase in j categories for that particular resource η^j and the area occupied by j^{th} resource ξ^m.

Based on the summarization above, we define a Utility Coefficient $\mathrm{UC_A}$ using the Def. 8 specified below.

Definition 8. The *Utility Coefficient* $\mathrm{UC_A}$ for area calculation for any given resource in $\Gamma = \big[\ \gamma_1^1\ ,\ \gamma_2^1\ \ ,\ldots,\ \gamma_{\eta^1}^1,\ \gamma_1^2\ ,\ \gamma_2^2,\ \gamma_{\eta^j}^2\ \ ,\ldots,\ \gamma_1^m\ ,\ \gamma_2^m\ \ ,\ldots,\ \gamma_{n-1}^m,\ \gamma_{\eta^m}^m\ \big]$ is defined as,

$$\mathrm{UC_A}(\gamma^{\texttt{Clock}}) = \frac{\Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{clock}}}{\eta_i^{\texttt{clock}}} \tag{4.27}$$

$$\mathrm{UC_A}(\gamma^{\texttt{mem}}) = \frac{\Delta\ \mathcal{R}_{\texttt{area}}^{\texttt{mem}}}{\eta_i^{\texttt{mem}}} \tag{4.28}$$

$$\mathrm{UC_A}(\gamma_i^j) = \frac{\xi^m \cdot \Delta\eta^j}{\eta^j} \tag{4.29}$$

■

A *Utility Rank* (UR) can be defined subsequent to the calculation of Utility Coefficients defined using Defs. 8, 6, 4 for execution time, power and area.

Definition 9. The UR is a unique set of ordered m-tuples ($\mathfrak{a}^1$, $\mathfrak{a}^2\ \ ,\ldots,\ \mathfrak{a}^m$) generated by the n-fold Cartesian product of resource categories η^1, η^2, $\eta^3\ \ ,\ldots,\ \eta^m$ such that,

$$\eta^1 \times \eta^2 \times \cdots \times \eta^m = \big\{(\mathfrak{a}^1, \mathfrak{a}^2\ ,\ldots,\ \mathfrak{a}^m) : \mathfrak{a}^j\ \in\ \eta^j,\ \forall j = 1, 2 \ldots m\big\} \tag{4.30}$$

The order of values in the n-tuple UR must follow a strict ordering ($\prec$) proportional to descending order of their UCs respectively. Thus, $\mathrm{UR_k}$ = [($\mathfrak{a}^1$, $\mathfrak{a}^2\ \ ,\ldots,\ \mathfrak{a}^m$)] where k denotes either Area, Power or Execution Time and values in the n-tuple are strictly ordered according to $\mathrm{UC_k}(\gamma_i^m)\ \prec\ \mathrm{UC_k}(\gamma_i^{m-1})\ \ ,\ldots,\ \prec\ \mathrm{UC_k}(\gamma_i^2)\ \prec\ \mathrm{UC_k}(\gamma_i^1)$. ■

The UR in Def. 9 will be used to represent the architecture design space of our proposed approach. The terms in the n-tuple of an UR indicates the variant combinations of available resources viz. adders, subtracters, clocks, etc., during the process of system design.
For example, if there are three categories of resources: 2 multiplexers, 3 de-multiplexers and 1 clock resource having an area of 60 area units (a.u.), 75 a.u. and 85 a.u. on the chip respectively. Suppose it was determined that $\mathrm{UC_A}\ (\gamma^{\texttt{Clock}})\ >\ \mathrm{UC_A}\ (\gamma^{\texttt{De-Mux}})\ >\ \mathrm{UC_A}\ (\gamma^{\texttt{Mux}})$.

According to Def. 9, the UR will be a set of 3-tuples or triples given as: $\mathrm{UR_A}$ = [(1,1,1), (2,1,1), (1,2,1), (1,3,1), (2,2,1), (2,3,1)]. The example shows that the m-ary Cartesian power of resulting n-tuple in Def. 9 is isomorphic to the design space variants from m-categories using tensor product in Def. 3.

4.4 Deployment of GWO-DSE framework for IoT Synthesis

We adopt the GWO algorithm discussed in Chapter. 2 Sec. 2.6.2 and apply it for extracting a minimized tuple specified Def. 9. The application of GWO for DSE process is termed as the GWO-DSE framework.

4.4.1 Initialization of GWO-DSE

In the case of GWO-DSE, each candidate wolf $\overrightarrow{X_i}$ is treated as a $n \times 1$ vector where n matches the dimension of each tuple during UR set calculation using Def. 9. However each tuple continues to comply with the initialization Equations. 2.10 and 2.11 of GWO.

4.4.2 Fitness function Calculation

The GWO-DSE framework records fitness values $\mathcal{F}^{fit}$ using Eq. (4.31) for each and every candidate solution $\overrightarrow{X_i}$. The three most optimal candidate solutions are shortlisted as $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ respectively. The following notations are to be conveyed before defining a *fitness function* for GWO-DSE framework.

- f_{rand} is a uniformly distributed random variable between 0 and x, ub and lb represent the first and last tuples generated by Cartesian product as a consequence of Def. 9.

ALGORITHM 3: GWO-DSE Framework

Data: W_n–magnitude of the wolf pack, $Iter_{MAX}$ – total number of iterations more than 0, UR – set of ordered m-tuples (${}^{j}\mathfrak{a}_i$, ${}^{j}\mathfrak{a}_i$, . . . , ${}^{m}\mathfrak{a}_i$), $\mathfrak{C}_{\mathcal{R}_{\mathtt{aw}}}$, $\mathfrak{C}_{\mathcal{R}_{\mathtt{ap}}}$ and $\mathfrak{C}_{\mathcal{R}_{\mathtt{area}}}$ – user defined constraints

Result: Max +ve Difference Triple:(d_1^+, d_2^+, d_3^+) $\leftarrow$ $\mathcal{F}^{fit}(\overrightarrow{X_i})$, Optimal Grey wolf position within given Design Space.

```
1  begin
2      Randomly initialize W_n number of candidate X_i Grey wolves
3      Using Eq. (4.31 identify the best three n-tuples as X_α, X_β and X_δ wolves
4      Set t:=0
5      repeat
6          foreach X_i ∈ UR do
7              Update X_i using Eq. (2.16)
8          end
9          Compute Eqs. (2.23, 2.15, 2.14) to update a, A, C
10         forall Wolves X_i of UR do
11             newPositions ← get positions using Eq. (2.10)
12             if newPositions ∉ Range then
13                 newPositions ← get positions using Eq. (2.11)
14             end
15         end
16         foreach X_i ∈ UR do
17             Deduce Difference-Triple ( d_1, d_2, d_3 ) subject to C_{R_aw}, C_{R_ap} and C_{R_area}
                using Eq. (4.31)
18         end
19         Rank α, β, and δ positions as premier 3 best solutions based on Difference
            Triples from Previous Step.
20         t ← t + 1
21     until t ≤ Iter_MAX
22     Choose Optimal Grey Wolf position based on ( d_1^+, d_2^+, d_3^+ )
23 end
```

- $\mathfrak{C}_{\mathcal{R}_{\mathtt{aw}}}$–workload or execution timing constraint, $\mathfrak{C}_{\mathcal{R}_{\mathtt{ap}}}$–power constraint and $\mathfrak{C}_{\mathcal{R}_{\mathtt{area}}}$–area constraint values respectively set by the design specifications.

- $\mathfrak{R}^{\mathtt{wrk}}(\overrightarrow{X_i})$, $\mathfrak{R}^{\mathtt{pwr}}(\overrightarrow{X_i})$ and $\mathfrak{R}^{\mathtt{ara}}(\overrightarrow{X_i})$–fetch functions that take a ordered n-tuple ($\overrightarrow{X_i}$) as input and return the time of execution, power consumption and occupied area for that

particular variant branch respectively.

- Difference function defined by $\text{Diff}_{\text{k}}(\ \overrightarrow{X_i}\ ) = \mathfrak{C}_k - \mathfrak{R}^{\text{k}}(\overrightarrow{X_i})$ specifying either positive difference or negative difference where k denotes either Area, Power or Execution Time.

A fitness function for the GWO-DSE framework is defined using Eq. (4.31) based on the summarization above.

$$\begin{aligned} \mathcal{F}^{fit}(\overrightarrow{X_i}) = \{\ (d_1, d_2, d_3)\ \mid\ & \\ d_1 \leftarrow \text{Diff}_{\text{pwr}}(\overrightarrow{X_i}), & \\ d_2 \leftarrow \text{Diff}_{\text{wrk}}(\overrightarrow{X_i}), & \\ d_3 \leftarrow \text{Diff}_{\text{ara}}(\overrightarrow{X_i})\ \} & \end{aligned} \tag{4.31}$$

The steps for GWO-DSE algorithm is outlined using Alg. 3.

4.5 Design Flow Analysis

The theory specified for the proposed framework for DSE in the previous sections shall be employed using real world examples in the following sections. The UR defined in Def. 9 determines the design space organized as a set of n-tuples based on UC specification for each of the resource categories. In addition, the GWO-DSE algorithm proposed in Alg. 3 will be used as a tool for searching the best architecture after the design space is randomly organized as part of the initialization procedure. The design objectives were chosen based on the contemporary generation of VLSI computing systems such as smart-phones, MP3 players, smart-watches, etc., where reliability aware power consumption, area on chip and execution time are of paramount importance[91]. The flow chart in Figure. 4.2 shows the entire design flow for high level synthesis using the proposed GWO-DSE.

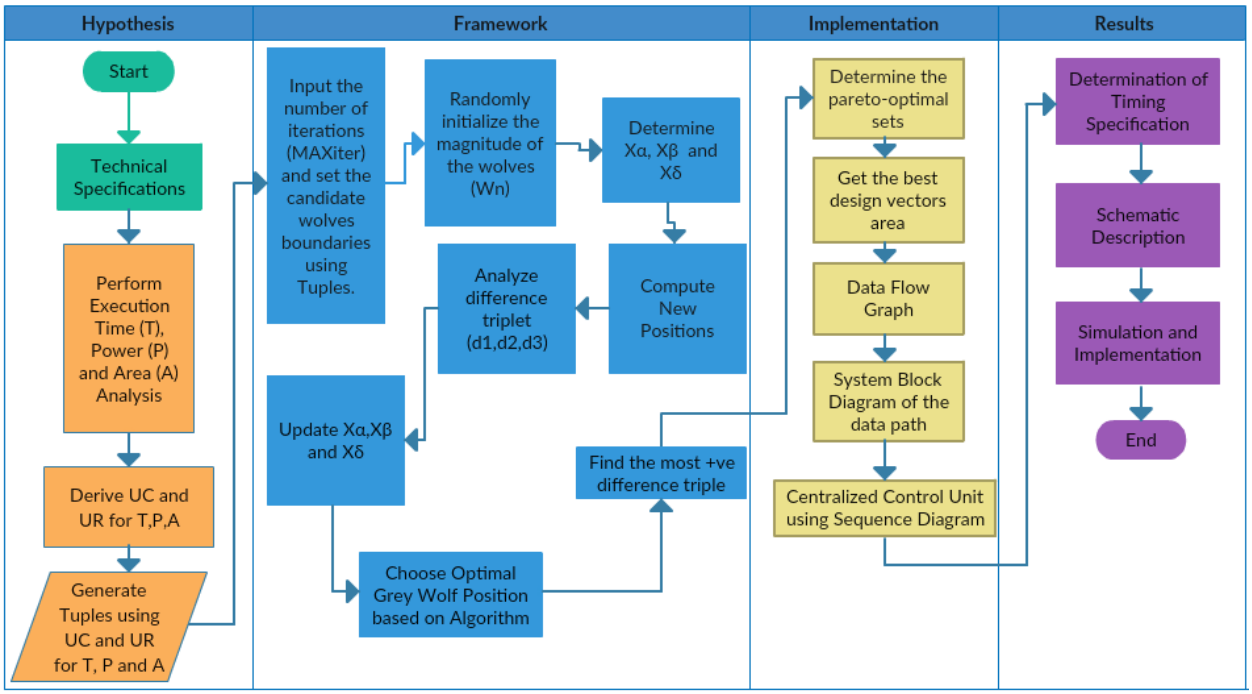


Figure 4.2: The proposed framework for high level design flow based on multi-parametric optimization.

4.5.1 Resource Libraries and Operational Constraints

For the purpose of demonstrating the design flow, the following specifications are assumed.

1. Maximum power consumption: 8 milliwatts (mW).
2. Maximum resources available for the system design:
    - Adder/Subtracter: 2 units
    - Multiplier: 4 units
    - Dividers: 2 units
    - Clock Frequency oscillators: 3 units.
    - Memory Element: 1 unit
3. Maximum Time of Execution: 140 μs (For 1000 sets of data).
4. Hardware Area of Resources: minimum while satisfying the above constraints.

Also, the following resource specifications are assumed as examples for system level design.

Clock Period for

- ✓ Divider/Operation : 4 clock cycles (cc)
- ✓ Multiplier/Operation : 4 cc
- ✓ Adder-Subtracter/Operation : 2 cc

Clock Frequency 100 MHz, 150 MHz and 200 MHz for 3 units respectively.

Area Occupied by

- ✓ Divider : 55 area units (a.u.) or 55 CLB on IoT device FPGA.
- ✓ Multiplier : 45 a.u. or 45 CLB on IoT device FPGA.
- ✓ Adder-Subtracter : 10 a.u.

- ✓ 100 MHz Clock Oscillator : 5 a.u.
- ✓ 150 MHz Clock Oscillator : 6 a.u.
- ✓ 200 MHz Clock Oscillator : 10 a.u.
- ✓ Memory Element : 3 a.u.

Power Consumed by

- ✓ 100 MHz Clock Oscillator : 10 milliwatts / area unit (mW/ a.u.)
- ✓ 200 MHz Clock Oscillator : 30 mW/ a.u.
- ✓ 150 MHz Clock Oscillator : 20 mW/ a.u.

In this chapter, the HLS design flow is demonstrated by utilizing the transfer function $H(s)$ of an Elliptic Wave filter. The example of Elliptic Wave filter was chosen arbitrarily and any other filter can be utilized for demonstration purpose. The transfer function in the analogue domain for an Elliptic Wave filter function is given by Eq. (4.33).

$$H(s) = \frac{0.01\,(s^2 + 7.25)\,(s + 1.576)}{(s^2 + 0.46s + 0.212)\,(01279s + 0.677)} \tag{4.32}$$

which translates to $H(z)$ in frequency domain denoted by,

$$H(z) = \frac{0.078 - 0.22z^{-1} + 0.22z^{-2}}{1 - 2.54z^{-1} + 2.11z} \tag{4.33}$$

Consequently, the input variables in time domain are $x(n)$, $x(n-1)$, $x(n-2)$, $x(n-3)$ for the Elliptic Wave filter and having an output as $y(n)$ with previous outputs being $y(n-1)$ and $y(n-2)$.

$$y(n) = 0.78\ x(n) - 0.22\ x(n-1) + 0.22\ x(n-2) + 2.54\ y(n-1) - 2.11\ y(n-2) \tag{4.34}$$

For the sake of simplicity in notations, the constants 0.78, 0.22, 2.54 and 2.11 shall be represented as $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ respectively.

4.5.2 Architecture Design Space development for Power Consumption

The architecture design space for proposed framework utilizes the tuples resulted from UR calculation defined in Def. 9. The design space organization manifests itself as strictly ordered tuples generated by Cartesian product of m sets of resources, where each set represents unique type of resource elements. Every tuple in UR represents the potential optimal solution within the design space. According to the specifications in Sec. (4.5.1), the $\mathrm{UC_P}$ defined in Def. 6 will be calculated for Reliability-aware power as follows. Multiplier ($*$) :

$$\begin{aligned} \mathrm{UC_P}(\gamma_i^*) &= \frac{(4-1)\cdot(45)\cdot(30)}{4} \\ &= 1012.5 \text{ mW} \end{aligned} \tag{4.35}$$

Adder/Subtracter ($\pm$) :

$$\begin{aligned} \mathrm{UC_P}(\gamma_i^{\pm}) &= \frac{(2-1)\cdot(10)\cdot(30)}{2} \\ &= 150 \text{ mW} \end{aligned} \tag{4.36}$$

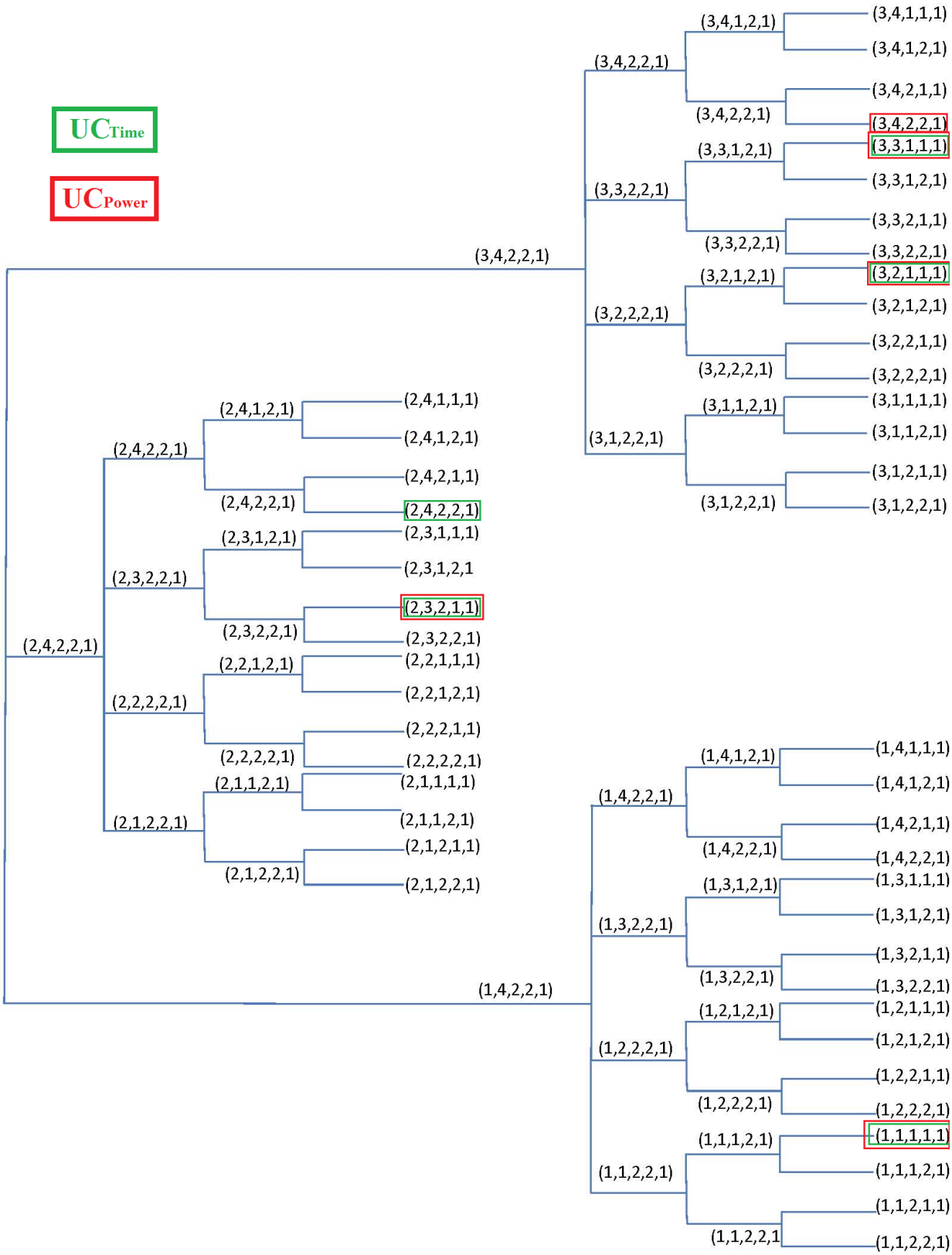


Figure 4.3: Phylogenetic tree indicating tuples arising due to UR_P evaluation in Eq. (4.40) and Eq. (4.46)

Divider ($\div$) :

$$\begin{aligned} \mathrm{UC_P}(\gamma_i^{\div}) &= \frac{(2-1)\cdot(55)\cdot(30)}{2} \\ &= 825 \text{ mW} \end{aligned} \tag{4.37}$$

Clock Oscillator (clock) :

$$\begin{aligned} \mathrm{UC_P}(\gamma_i^{\texttt{Clock}}) &= \frac{[2(10)+4(45)+2(55)]}{3}\cdot(30-10) \\ &= 2066.66 \text{ mW} \end{aligned} \tag{4.38}$$

Memory (mem) :

$$\begin{aligned} \mathrm{UC_P}(\gamma_i^{\texttt{mem}}) &= \frac{[(1)\ (3)]}{1}\cdot(30) \\ &= 90 \text{ mW} \end{aligned} \tag{4.39}$$

Based on the $\mathrm{UC_P}$ calculations above, we can determine the set of tuples for $\mathrm{UR_P}$ as follows

$$\begin{aligned} &\text{Using Eq. (4.30)} \qquad \mathrm{UR_P} = \eta^{\texttt{mem}} \times \eta^{\pm} \times \eta^{\div} \times \eta^{*} \times \eta^{\texttt{Clock}} \\ &\iff \mathrm{UC_P}(\gamma_i^{\pm}) \prec \mathrm{UC_P}(\gamma_i^{\div}) \prec \mathrm{UC_P}(\gamma_i^{*}) \prec \mathrm{UC_P}(\gamma_i^{\texttt{Clock}}) \\ &\qquad \prec \mathrm{UC_P}(\gamma_i^{\texttt{mem}}) \end{aligned} \tag{4.40}$$

A branching diagram (Figure. 4.3) in the form of Phylogenetic tree[92] is employed to visualize the n-tuples generated as a result of Eq. (4.40). In Figure. 4.3, the red boxes around certain tuples illustrates an instance of Grey Wolf Optimizer discussed in Alg. 3 which is set in motion during the process of pruning the DSE. This pruning process assists in finding the Pareto-optimized design tuple satisfying the power consumption within specified user-defined constraints $\mathfrak{C}_{\mathcal{R}_{\texttt{aw}}} = 80$ ms, $\mathfrak{C}_{\mathcal{R}_{\texttt{ap}}} = 6$ mW and $\mathfrak{C}_{\mathcal{R}_{\texttt{area}}} = 220$ a.u. The main objective

of the Alg. 3 during the process of DSE is to shortlist potential candidate tuples of $\mathrm{UR_P}$ that yield the maximum positive difference triple (d_1^+, d_2^+, d_3^+) in an increasing order of potential optimal solutions such as X_α, X_β and X_δ subject to $\mathfrak{C}_{\mathcal{R}_{\mathtt{aw}}}$, $\mathfrak{C}_{\mathcal{R}_{\mathtt{ap}}}$ and $\mathfrak{C}_{\mathcal{R}_{\mathtt{area}}}$.
For example, a snapshot of an execution instance of Alg. 3 during one of the iteration rounds will have the following values specified for different parameters specified in Table. 4.1. The input for initialization of the Alg. 3 during such an instance is the randomly generated tuples with boundaries setup using phylogenetic tree of Figure. 4.3. The final outcome in the example snapshot instantiation will choose *Optimal Grey Wolf* position based on (d_1^+, d_2^+, d_3^+) and it will be the tuple ($3, 3, 1, 1, 1$). Therefore, the X_α chosen will have corresponding power as 6 mW, 60.03 ms of execution time and an area of 210 a.u. respectively, which falls within user-defined constraints $\mathfrak{C}_{\mathcal{R}_{\mathtt{aw}}} = 80$ ms, $\mathfrak{C}_{\mathcal{R}_{\mathtt{ap}}} = 7$ mW and $\mathfrak{C}_{\mathcal{R}_{\mathtt{area}}} = 220$ a.u. The time of execution ($\mathrm{T_{exe}}$) and latency columns in Table. 4.1 are computed using Eq. (4.12).

4.5.3 Architecture Design Space development for Execution Time

In this section the methodology for organizing the design space tuples for UR calculation follows similar steps outlined in Sec. (4.5.2). The resource depletion ratio ζ_r for UC estimation described in Def. 4 will be set at a decaying rate of 1% for given 12 type of resources under ideal conditions with specifications defined in Sec. (4.5.1). Thus, the $\mathrm{UC_T}$ defined in Def. 4 for Reliability-aware Workload can be deduced as, Multiplier ($*$) :

$$\begin{aligned} \mathrm{UC_T}(\gamma_i^*) &= \frac{(4-1)\cdot(4)\cdot(0.01)}{0.99\cdot(4)} \\ &= 30.3 \text{ ms} \end{aligned} \tag{4.41}$$

Adder/Subtracter ($\pm$) :

$$\mathrm{UC_T}(\gamma_i^{\pm}) = \frac{(2-1)\cdot(2)\cdot(0.01)}{0.99\cdot(2)}$$

Tuple	Power (mW)	Latency	T_{exe}	Area	$\mathcal{F}^{fit}(\overrightarrow{X_i})$
(1,1,1,1,1)	[(1 * 5) + (1 * 45) + (1 * 55) + (1 * 10) + (1 * 3)]/1000 * 10 = 1.18 mW	26	240.02	114	(-160.02, 5.82, 106)
(3,4,2,2,1)	[(3 * 5) + (4 * 45) + (2 * 55) + (2 * 10) + (1 * 3)]/1000 * 30 = 9.84 mW	14	60.01	320	(19.99, -2.84, -100)
(3,3,1,1,1)	[(3 * 5) + (3 * 45) + (1 * 55) + (1 * 10) + (1 * 3)]/1000 * 10 = 6.54 mW	18	60.03	210	(19.97, 0.46, 10)
(3,2,1,1,1)	[(3 * 5) + (2 * 45) + (1 * 55) + (1 * 10) + (1 * 3)]/1000 * 30 = 5.19 mW	20	80.02	165	(-0.02, 1.81, 55)
(2,3,2,1,1)	[(2 * 5) + (3 * 45) + (2 * 55) + (1 * 10) + (1 * 3)]/1000 * 20 = 5.36 mW	16	79.22	216	(0.78, 1.64, 4)

Table 4.1: An instance of GWO-DSE framework in action for power analysis. Note. Units for Latency is in c.c., T_{exe} is in ms and area in a.u. respectively.

$$= 10.1 \text{ ms} \tag{4.42}$$

Divider ($\div$) :

$$UC_T(\gamma_i^{\div}) = \frac{(2-1)\cdot(4)\cdot(0.01)}{0.99\cdot(2)}$$

$$= \qquad\qquad = 20.3 \text{ ms} \tag{4.43}$$

Clock Oscillator (clock) :

$$\begin{aligned} \mathrm{UC_T}(\gamma_i^{\texttt{Clock}}) &= \frac{[(2+2)\cdot(4+4)\cdot(2+4)\cdot(0.01-0.005)]}{3} \\ &= 320 \text{ ms} \end{aligned} \tag{4.44}$$

Memory (mem) :

$$\begin{aligned} \mathrm{UC_T}(\gamma_i^{\texttt{mem}}) &= \frac{[(1)\cdot(1)\cdot(0.01)]}{0.99\cdot(1)} \\ &= 10.1 \text{ ms} \end{aligned} \tag{4.45}$$

As per the $\mathrm{UC_T}$ calculations above, the set of tuples for $\mathrm{UR_T}$ can be determined as,

$$\begin{aligned} &\text{Using Eq. (4.30)} \qquad \mathrm{UR_T} = \eta^{\texttt{mem}} \times \eta^{\pm} \times \eta^{\div} \times \eta^{*} \times \eta^{\texttt{Clock}} \\ &\iff \mathrm{UC_T}(\gamma_i^{\pm}) \prec \mathrm{UC_T}(\gamma_i^{\div}) \prec \mathrm{UC_T}(\gamma_i^{*}) \prec \mathrm{UC_T}(\gamma_i^{\texttt{Clock}}) \\ &\quad \prec \mathrm{UC_T}(\gamma_i^{\texttt{mem}}) \end{aligned} \tag{4.46}$$

By observing the Phylogenetic tree in Figure. 4.3, the green boxes around certain tuples illustrates an instance of Grey Wolf Optimizer discussed in Alg. 3 which is a process state while pruning the DSE. The Pareto-optimized design tuple ($3, 3, 1, 1, 1$) is obtained during the pruning process and it is the final outcome result having optimal execution time based on user-defined constraints $\mathfrak{C}_{\mathcal{R}_{\texttt{aw}}} = 80$ ms, $\mathfrak{C}_{\mathcal{R}_{\texttt{ap}}} = 6$ mW and $\mathfrak{C}_{\mathcal{R}_{\texttt{area}}} = 220$ a.u.
The crux of the Alg. 3 is to determine potential tuples of $\mathrm{UR_T}$ that produce the maximum positive difference triple (d_1^+, d_2^+, d_3^+) in a strict increasing order to attain the objective X_α, X_β and X_δ subject to $\mathfrak{C}_{\mathcal{R}_{\texttt{aw}}}$, $\mathfrak{C}_{\mathcal{R}_{\texttt{ap}}}$ and $\mathfrak{C}_{\mathcal{R}_{\texttt{area}}}$.

Tuple	T_{exe} (ms)	Latency	Power	Area	$\mathcal{F}^{fit}(\overrightarrow{X_i})$
(1,1,1,1,1)	[(26) + (1000 - 1) * 24] * 0.01 = 240.02 ms	26	1.18	114	(-160.02, 5.82, 106)
(3,2,1,1,1)	[(20) + (1000 - 1) * 16] * 0.005 = 80.02 ms	14	5.19	320	(19.99, -2.84, -100)
(2,4,2,2,1)	[(14) + (1000 - 1) * 12] * 0.0066 = 79.21 ms	20	6.2	165	(-0.02, 1.81, 55)
(2,3,2,1,1)	[(16) + (1000 - 1) * 12] * 0.0066 = 79.22 ms	16	5.36	216	(0.78, 1.64, 4)
(3,3,1,1,1)	[(18 + (1000 - 1) * 12] * 0.005 = 60.03 ms	18	6.54	210	(19.97, 0.46, 10)

Table 4.2: An instance of GWO-DSE framework in action for execution time analysis. Note. Units for Latency is in c.c., T_{exe} is in ms and area in a.u. respectively.

4.5.4 Architecture Design Space development for Area Calculation

For an architecture design space resulting due to tuple set generation using Def. 9, each tuple in UR represents a potential optimal solution to represent the design space. As per the specifications defined in Sec. (4.5.1), the UC_A defined in Def. 4 will be calculated for Reliable-Area as follows.

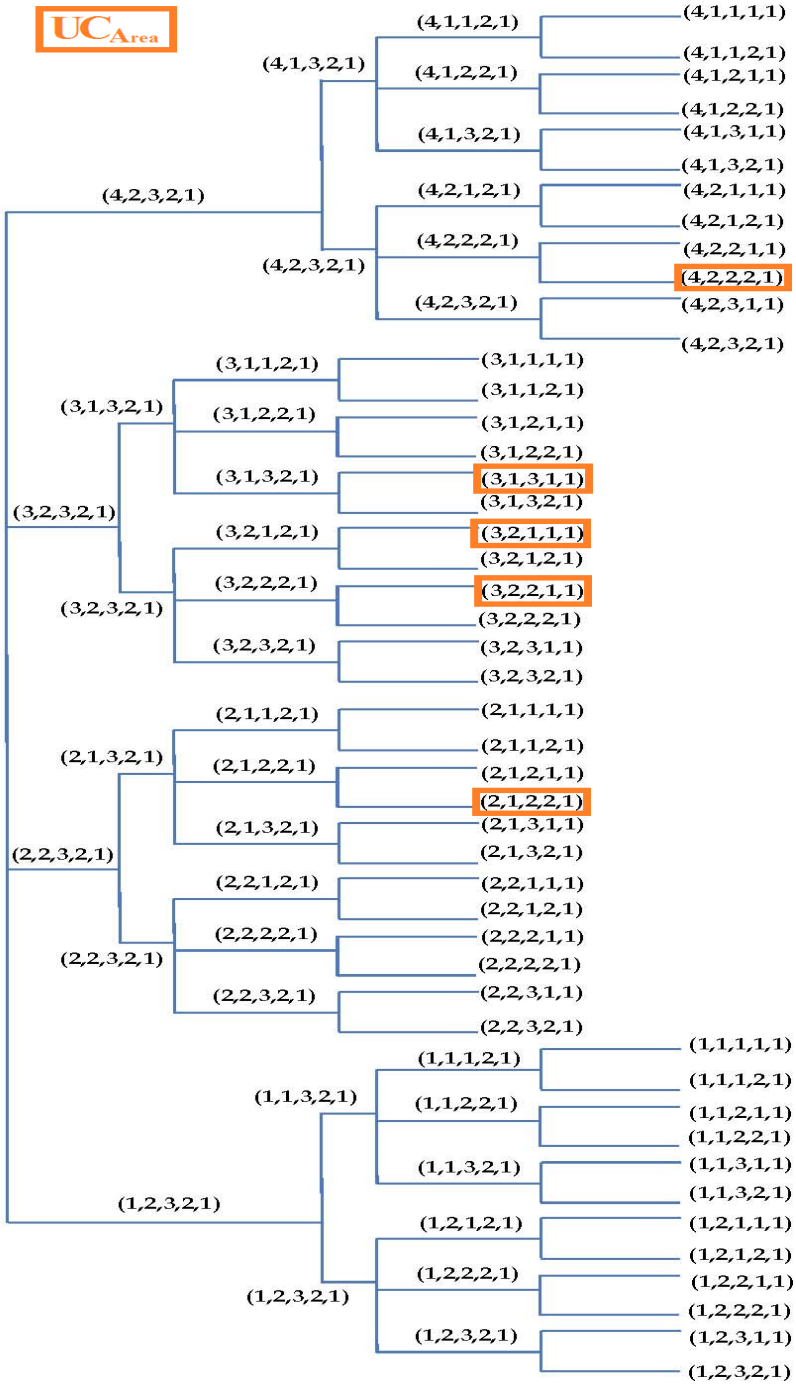


Figure 4.4: Phylogenetic tree indicating tuples arising due to $\mathrm{UR_A}$ evaluation in Eq. (4.54)

Multiplier ($*$) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{*}) &= \frac{(4-1)\cdot(45)}{4} \\ &= 33.75 \text{ a.u.} \end{aligned} \tag{4.47}$$

Adder/Subtracter ($\pm$) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{\pm}) &= \frac{(2-1)\cdot(10)}{2} \\ &= 5 \text{ a.u.} \end{aligned} \tag{4.48}$$

Divider ($\div$) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{\div}) &= \frac{(2-1)\cdot(55)}{2} \\ &= 27.5 \text{ a.u.} \end{aligned} \tag{4.49}$$

Clock Oscillator 1 (100 MHz) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{\texttt{Clock1}}) &= \frac{[(3-1)\cdot(5)]}{3} \\ &= 3.33 \text{ a.u.} \end{aligned} \tag{4.50}$$

Clock Oscillator 2 (150 MHz) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{\texttt{Clock2}}) &= \frac{[(3-1)\cdot(6)]}{3} \\ &= 4 \text{ a.u.} \end{aligned} \tag{4.51}$$

Clock Oscillator 3 (200 MHz) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{\texttt{Clock3}}) &= \frac{[(3-1)\cdot(10)]}{3} \\ &= 6.66 \text{ a.u.} \end{aligned} \tag{4.52}$$

Memory (mem) :

$$\begin{aligned} \mathrm{UC_A}(\gamma_i^{\texttt{mem}}) &= \frac{[(1)\ (3)]}{1} \\ &= 3 \text{ a.u.} \end{aligned} \tag{4.53}$$

Based on the $\mathrm{UC_A}$ calculations above, we can determine the set of tuples for $\mathrm{UR_A}$ as follows,

$$\begin{aligned} &\text{Using Eq. (4.30)} \qquad \mathrm{UR_A} = \eta^{\texttt{mem}} \times \eta^{\pm} \times \eta^{\texttt{Clock}} \times \eta^{\div} \times \eta^{*} \\ &\iff \mathrm{UC_A}({}^{*}\gamma_i) \prec \mathrm{UC_A}({}^{\div}\gamma_i) \prec \mathrm{UC_A}(\gamma_i^{\texttt{Clock}}) \prec \mathrm{UC_A}({}^{\pm}\gamma_i) \\ &\quad \prec \mathrm{UC_A}(\gamma_i^{\texttt{mem}}) \end{aligned} \tag{4.54}$$

The orange boxes in the Phylogenetic tree of Figure. 4.4 show certain tuples which are generated as a result of an instance of Grey Wolf Optimizer discussed in Alg. 3. Alg. 3 initiates the process of pruning the DSE to find the pareto-optimized design tuple satisfying the surface area occupies within specified user-defined constraints $\mathfrak{C}_{\mathcal{R}_{\texttt{aw}}} = 80$ ms, $\mathfrak{C}_{\mathcal{R}_{\texttt{ap}}} = 6$ mW and $\mathfrak{C}_{\mathcal{R}_{\texttt{area}}} = 220$ a.u. The Alg. 3 generates random tuples within the boundaries of DSE and identifies potential candidate tuples of $\mathrm{UR_A}$ that can produce maximum positive difference triple (d_1^+, d_2^+, d_3^+) in a strict increasing order beginning with the X_α, X_β and X_δ subject to $\mathfrak{C}_{\mathcal{R}_{\texttt{aw}}}$, $\mathfrak{C}_{\mathcal{R}_{\texttt{ap}}}$ and $\mathfrak{C}_{\mathcal{R}_{\texttt{area}}}$. An instance of Alg. 3 during the DSE process iteration rounds shall have the following tuples identified described in Table. 4.3. As visualized in the Table. 4.3, the final optimal solution deduced by the algorithm is shown by the corresponding row using *red* colored text. [h]

Tuple	Area (a.u.)	Latency	T_{exe}	Power	$\mathcal{F}^{fit}(\overrightarrow{X_i})$
(3,1,3,1,1)	[(3 * 45) + (1 * 55) + (1 * 10) + (1 * 10) + (1 * 3)] = 213 a.u.	20	60.02	6.54	(19.99, -2.84, -100)
(3,2,2,2,1)	[(3 * 45) + (2 * 55) + (6 * 1) + (1 * 10) + (1 * 3)] = 264 a.u.	14	260.01	6.84	(-160.02, 5.82, 106)
(4,2,2,2,1)	[(4 * 45) + (2 * 55) + (6 * 1) + (2 * 10) + (1 * 3)] = 319 a.u.	18	60.03	210	(19.97, 0.46, 10)
(3,2,1,1,1)	[(3 * 45) + (2 * 55) + (1 * 5) + (1 * 10) + (1 * 3)] = 263 a.u.	20	80.02	5.19	(-0.02, 1.81, 55)
(2,1,2,2,1)	[(2 * 45) + (1 * 55) + (1 * 6) + (2 * 10) + (1 * 3)] = 174 a.u.	16	79.22	3.5	(0.78, 1.64, 4)

Table 4.3: An instance of GWO-DSE framework in action for area analysis. Note. Units for Latency is in c.c., T_{exe} is in ms and area in a.u. respectively.

4.6 Results and analysis

4.6.1 Preprocessing for Data-path and Control Unit

When the optimal architecture configuration is processed by GWO-DSE framework as described in the previous sections, a scheduling procedure must be determined for the optimal tuple chosen by the DSE process. The scheduling must also adhere to the specifications in Sec. (4.5.1). There are a plethora of ways for implementing this scheduling procedure. However, in GWO-DSE model unconstrained scheduling[93] is chosen since the initial framework allocates starting time to components under the assumption that an unlimited amount of resources are available. Therefore in case of reconfigurable devices, this corresponds to placing

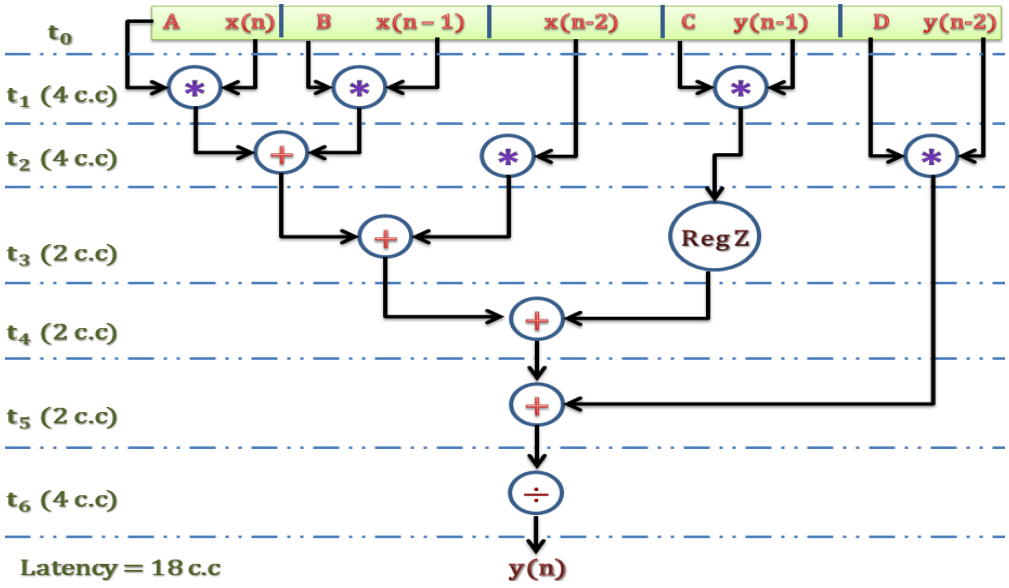


Figure 4.5: DFG Sequencing using ASAP algorithm for Elliptic Wave Filter of Eq. (4.33)

modules in a device with unlimited size that allows any partition to be implemented. Because the devices in reality have only a limited amount of resources, unconstrained scheduling cannot be used as such. Instead, it is usually used as pre-processing step for other algorithms. It can be used for instance for the computation of the upper and lower bounds on the starting time of operations in a dataflow graph. Whereas the *lower bound* provides the earliest time at which a module can be scheduled, the *upper bound* defines the latest time at which a module can be started. The difference between the upper and the lower bound of a module is its *mobility range*[93].

To compute the earliest starting time of each component, the *as soon as possible* (ASAP)-Algorithm defined in Alg. 4 is utilized for the GWO-DSE framework. The ASAP-algorithm traverses the dataflow in topological order starting from the primary input. Each primary input resource is assigned a starting time 0. The ASAP algorithm idealizes the binding process by assuming an unlimited amount of available resource, and assigns each operation as soon as it is ready to be executed, therefore providing the lowest starting time of the tasks in the graphs. The pseudo code for the ASAP-algorithm is described in Alg. 4 for the resulting

status_scheduling output. The corresponding time allocation using Alg. 4 for specifications in Sec. (4.5.1) is illustrated by Figure. 4.5.

As the scheduling procedure is completed a multiplexing scheme for each resource must be devised and such a multiplexing scheme is described in detail in [94, 76]. Using the multiplexing scheme, a data-path generation block diagram for case study of Elliptic Wave Filter (Eq. (4.33)) application is developed. This is followed by estimation of control and other synchronization signals which assist in design of a centralized control unit. Thus a complete schematic structure at Register Transfer (RT) level will become available subject to simultaneous development of data path unit and control path unit of Elliptic Wave filter ASP and subsequent deployment in the target IoT FPGA (Spartan 3E FPGA) for verification. The simulation of the block schematic performed in Xilinx ISE simulator[95] produced the expected outcome for the digital Elliptic Wave Filter design. Furthermore the hardware emulation of the design was performed to check the correct functionality of the design. The final results deduced that the device was realized successfully in IoT FPGA and is in compliance with all specifications in Sec. (4.5.1).

In the DFG described in Figure. 4.5 the Register Z represents the memory element which has been incorporated in time slot t_3 because the results of multiplier at time slot t_1 is not utilized until time slot t_4. As visualized from the Figure. 4.5, the total latency is 18 clock cycles and Figure. 4.6 shows the cycle time calculation using a Sequence Diagram for optimal tuple evaluation using Def. 9. A sequence diagram is an interaction diagram that shows how objects operate with one another and also the order of their interaction[96].

4.6.2 Comparative analysis with Contemporary Heuristic approach and ABF method

The GWO-DSE framework introduced in this chapter is streamlined in terms of efficiency with respect to architecture evaluation and its respective exploration time. The salient highlights of designing an optimized hardware is to choose a tuple of the design that best

reflects the algorithm in its micro-architecture. Based on the available hardware resources there can be a numerous ways of synthesizing a given logic at different abstraction levels like RTL. The GWO-DSE framework facilitates to determine and evaluate the suitability of a particular design tuple for a given task. The set of tuples realized using UR evaluation for power, execution time and area can have different characteristics and operational properties. The proposed framework has been verified on a number of popular high level synthesis benchmark digital signal processing (DSP) filters[97] both large and small, such as Auto Regressive Filter (ARF), Discrete Wavelet Transformation (DWT), Infinite impulse response (IIR) Digital Butterworth Filter and other different IIR digital filters. The framework also utilizes finite impulse response (FIR) filter as well as an Elliptic Wave Filter (Eq. (4.33)) which is case study filter in the proposed framework. The magnitude of the design space consisting of different tuple sets for respective DSP filters are indicated in Table. 4.4.

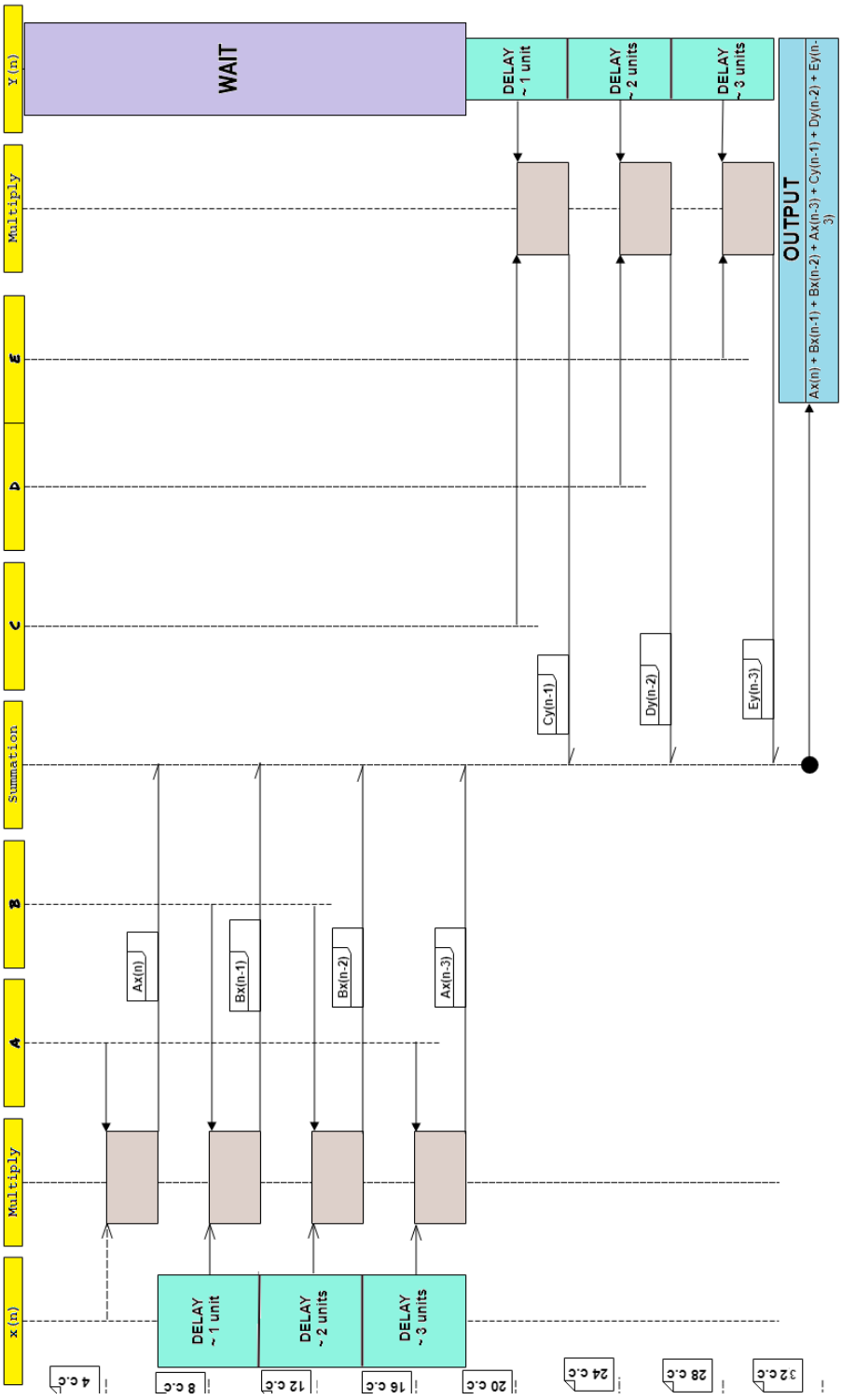


Figure 4.6: Cycle time diagram for finding the optimal tuple in UR based on DFG in Figure. 4.5

For example, 450 is the cardinality of the tuple set generated in design space for EWF, if 2 parameters were chosen; while on the other hand, the cardinality of set for design space with respect to Chebyshev and DWT are 188 and 266 respectively.

For larger sets, the exhaustive search process is extremely slow and takes a few weeks to return the possible solution. Also, an exhaustive search iterates over each possible tuple in the design space to find the most optimal solution. To speed up the exhaustive search, an attempt was made to arrive at an Pareto-optimal solution employing knapsack algorithm in [38]. However the method in [38] performs faster under select conditions of best and average cases whereas in worst case is same as exhaustive search due to exponential complexity of MDMKP[98]. Thus we discard the exhaustive search for comparison and choose the PSO[42] and ABF[1] for our comparative analysis in Table. 4.44.5 since they are the most recent DSE methodologies that closely compete with our domain in this article.

A summary of critical analysis benchmarking PSO[42] and ABF[1] with respect to GWO-DSE framework is outlined in Table. 4.4. As evident from the results indicated in in Table. 4.4, the proposed approach is capable of achieving tremendous speed boost compared to the PSO[42] and ABF[1] methodologies. The acceleration achieved is more than 90% while comparing IIR 3 variant, Auto regressive, DWT filters while it is more than 100% in case of Elliptic Wave and FIR filters.

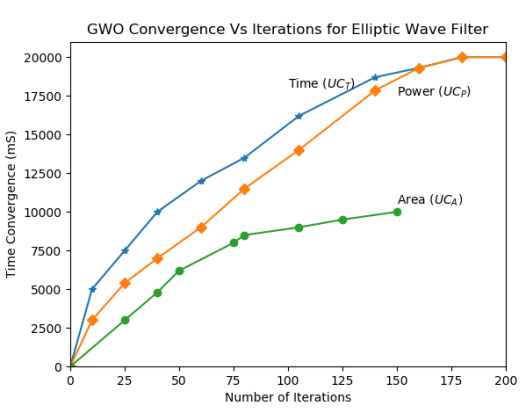
The Table. 4.5 outlines two aspects: (1) architectural evaluation methods using DSE based on different methodologies to find the best optimal resource arrangement for IoT device realization (2) the speed achieved due to respective methodologies. The benchmarks in Table. 4.4 utilizes the meta-heuristic optimizations based search like PSO and ABF which are the contemporary state of the art methodology for DSE. Further investigations reveal that the proposed approach is able to provide high acceleration for design space exploration while simultaneously maintaining the accuracy needed in architecture selection. It is evident from architecture evaluation section of the Table. 4.5 that GWO-DSE column numbers have higher magnitude than its counterpart heuristic method PSO, but substantially lesser magnitude

compared to search based on ABF[1]. This is due to the fact that GWO-DSE tends to cast wider net to accommodate as many unique tuples as possible due to Cartesian product falling short of repetitive arrangements as in ABF search model. Furthermore ABF suffers from adaptive mechanisms such as resource clamping and step size clamping to handle boundary outreach problem during exploration which limits its search space outreach. Comparatively in [42] the PSO model accommodates less number of search variants which includes best, worst and average case tuples for each iteration, although this might be efficient momentarily it leads to the issue of optimal solution stuck in local minima. As observed from the speed boost percentage indicators of column 3 of Table. 4.5, the proposed GWO-DSE framework significantly outpaces in terms of acceleration compared both to its counterpart heuristic PSO methodology as well as ABF methodologies all the while delivering accurate resource configurations during DSE.

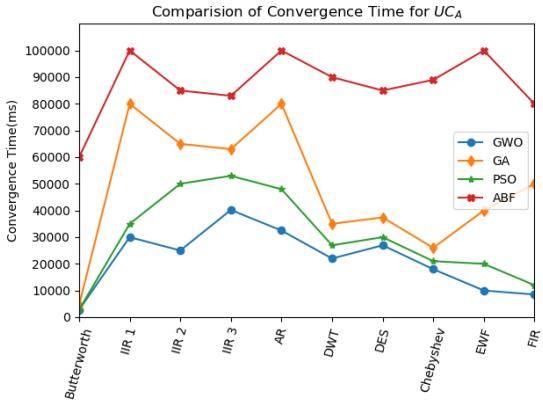
4.6.3 Convergence Analysis

Figures. 4.7b, 4.7c and 4.7d show the time taken to converge by the GWO algorithm for the Elliptic Wave Filter (EWF). The convergence time is plotted with respect to number of iterations for the architecture design space development in terms of execution time, power and area respectively. As depicted in Fig. 4.7a, it is evident that the development of the design space in terms of area reaches convergence before the development of the design space in terms of power and time can be reached. The convergence time for most optimal tuples found from the GWO algorithm for Area calculations is about 10 seconds whereas for time and power is close to 20 seconds.

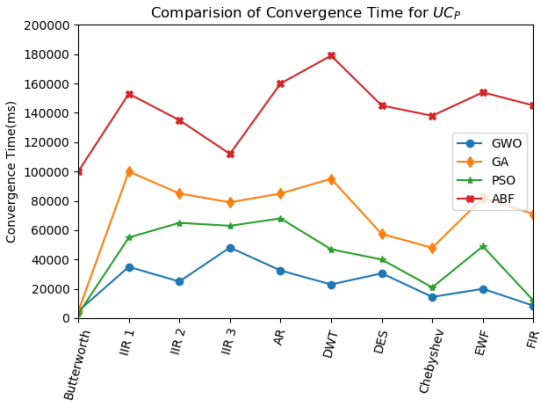
On comparing the GWO algorithm with other heuristic methods in Fig. 4.7b such as the ABF, GA and the PSO algorithms for different filters while analyzing the convergence time, it can be observed that ABF and the GA methods take the highest amount of time to converge. The ABF requires handling of constraints and parameters such as decoders, special operators, and penalty functions. There is a generalization difficulty as well as an acute



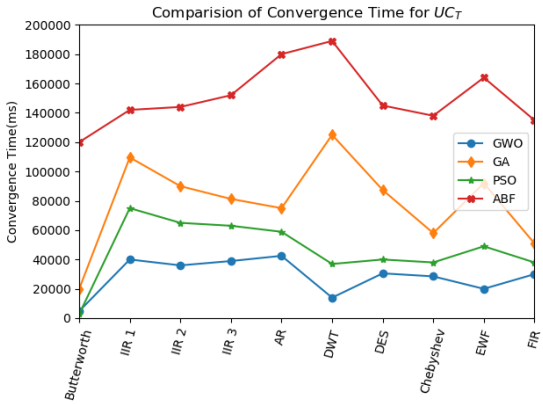
(a) Convergence vs Iterations



(b) UC_A convergence analysis



(c) UC_P convergence analysis



(d) UC_T convergence analysis

Figure 4.7: Convergence Scenario Analysis

hindrance in fine tuning of parameters when the number of iterations increase in the algorithm. This leads to a higher complexity cost and consequently the convergence time also increases. The GA requires a certain number of chromosomes for the mutation and crossover operations and also suffers from loss of diversity after a few iterations. As a result, design engineers will have to make intermittent tweaks in order to render the algorithm back into the system for it to consistently function and produce quality results. Consequently, this tweaking, and rendering after every few iterations impacts the time taken by the algorithm to converge with an exponential increase. For filters such as the Chebyshev filter, the GA converges faster but for filters that are more complex such as the Auto Regressive Filter or

the Elliptic Wave filter, the time taken by the algorithm to converge is significantly higher (Greater than 50seconds). The PSO algorithm does not have to go through the aforementioned adjustments that GAs will need to undergo, and it works far better. However, if space becomes highly complex and highly dimensional, it does not perform well and results in inconsistent output. Comparatively, GWO is much more robust meta-heuristic that overcomes the challenges faced by the previously mentioned algorithms since it has best ability to fully exploit and explore the search space despite the complex nature of the problem. As a result, the convergence time taken by the GWO in finding the most optimal tuples in terms of power area and time is far less than the other compared algorithms.

ALGORITHM 4: As Soon As Possible

Input: Operations O, Maximum number of control steps M.
Output: Control step for each operations, Status of scheduling

```
1  index ← 0
2  foreach o_i in O do
3  |  if TRUE = hasImmediatePredecessor(o_i) then
4  |  |  control_step(o_i) ← 1
   |  |  /* control_step (o_i) indicates control step into which operation o_i
   |  |     is scheduled.                                                    */
5  |  else
6  |  |  control_step (o_i) ← MAX (control_step (o_j)) + 1,
7  |  |  ∀ o_j ∈ {o' | o' ← immediate_predecessor(o_j)}.
8  |  end
9  end
10 foreach m ≤ M do
11 |  if control_step_m (o_i) == 1 then
12 |  |  Status_Scheduling ← Success
13 |  end
14 |
15 end
16 return Status_Scheduling
```

Total Tuples ∈ $UR_{\text{Architecture Design Space}}$			Tuples Analyzed by GWO-DSE			Net Estimate for GWO-DSE	
DSP Filters	1 Parameter	2 Parameters	Workload	Power	Area	Total tuples	Acceleration %
IIR Butterworth	24	48	4	4	5	18	79.6
IIR 1	32	64	4	4	5	18	77.3
IIR 2	36	72	4	5	7	19	82.7
IIR 3	48	96	5	6	4	21	94.23
Auto Regressive	98	172	6	9	12	29	95.06
Discrete Wavelet Transform	188	266	7	9	12	31	94.07
Differential Equation Solver	90	188	17	22	16	45	89.11
IIR Chebyshev	64	188	17	14	25	52	88.56
Elliptic Wave	220	450	7	27	32	52	102.08
Finite Impulse Response	550	1200	7	27	32	57	107.1

Table 4.4: Critical Analysis benchmarking ABF vs *GWO-DSE* framework

Total Tuples ∈ $UR_{\text{Architecture Design Space}}$			Architecture evaluation methods using DSE to find optimal arrangement			Speed Boost % of GWO-DSE	
DSP Filters	1 Parameter	2 Parameters	GWO-DSE	PSO[42]	ABF[1]	PSO[42]	ABF[1]
IIR Butterworth	24	48	8	4	5	65.34	79.6
IIR 1	32	64	8	4	5	45.67	77.3
IIR 2	36	72	7	3	5	62.88	82.7
IIR 3	48	96	8	3	5	57.08	94.23
Auto Regressive	98	172	6	4	9	67.54	95.06
Discrete Wavelet Transform	188	266	5	4	9	53.22	94.07
Differential Equation Solver	90	188	22	12	31	62.34	89.11
IIR Chebyshev	64	188	16	9	21	54.98	88.56
Elliptic Wave	220	450	39	27	62	72.12	102.08
Finite Impulse Response	550	1200	15	8	22	67.09	107.1

Table 4.5: Performance highlights of *GWO-DSE* framework vs ABF[1] and *State of the art* Heuristic method.

4.7 Conclusion

In this chapter, a structured approach is presented to synthesize a multi parametric optimized Application Specific Processor. A clear step by step flow from the inception of Utility Coefficients for Power, Area and Workload analysis to establish a framework to search the best optimal arrangement ranked using utility ranking concept is presented in this chapter. The results disseminated in this chapter and hardware synthesized on an IoT device verifies correctness of the design process. The logic synthesized can work as hardware accelerator for large systems or as a stand alone processing element as ASIC. The proposed GWO-DSE framework ensures that an efficient and multi-parametric optimized hardware can be synthesized accurately for any computational intensive application. Moreover it significantly outperforms the state of the art heuristic technique such as PSO as well as ABF methodologies in terms of speed and accuracy. We believe that this is the first work which shows how to optimize hardware resource configurations based on Grey Wolf optimization methodology for HLS.

Chapter 5

Security Enhancement in IoT device

In this chapter we introduce an adaptive scheme is to keep pace with the mobile security requirements of lightweight cipher systems of IoT devices. Adaptive ciphers are emerging to be an effective cryptographic architecture for the next generation security infrastructures. Such cipher designs are difficult to implement since they are prone to weaknesses based on usage, with properties being similar to one-time pad; keystream is also subjected to very strict requirements. A more practical solution is essentially not to develop an newer or stronger cipher scheme, but to devise ways to enhance cipher strength without undue changes to underlying VLSI architectures on a permanent basis. Recently, Modulo Addition 2^n has been suggested over logic XOR as a mixing operator to guard against such attacks. However, it has been observed that the complexity of Modulo Addition can be drastically decreased with the appropriate formulation of polynomial equations and probabilistic conditions. In this Chapter, we present a new cryptographic architecture for Modulo Addition. The framework for the new design is characterized by user-defined expandable security for stronger encryption and does not impose changes in existing layout of light weight cipher designs of IoT. The structure of the proposed design is highly scalable, which boosts the algebraic degree and thwarts the probabilistic conditions by maintaining the original hardware complexity

without changing the integrity of the Modulo Addition 2^n component.

5.1 Background

The security of data transmission is a vital aspect of IoT communication networks. An IoT communication system is only reliable as long as it provides a high level of security. Nowadays, millions of users exchange sensitive or classified information and documents via global system of interconnected computer networks that use the Internet. From e-mail to cellular communications secure web access to digital cash, cryptography is an essential part of today's information systems. Cryptography is the science of keeping transmitted data secure. Cryptography helps provide accountability, fairness, accuracy, and confidentiality. It can prevent fraud in electronic commerce and assure the validity of critical data transactions.

In this age of explosive worldwide growth of electronic data storage and mobile communications, many vital information exchange tasks require effective protection especially in the IoT domain. When used in conjunction with other approaches to information security, cryptography provides a powerful means for protecting information and data encryption for secure communication. Although cryptography improves the achievable communications confidentiality, it requires additional computational power and imposes latency [99], since a certain amount of time is required for both data encryption and decryption [100]. Recently, there has been an emergence of scalable security protocols using lightweight cryptography for IoT devices. These protocols target a wide variety of IoT devices that can be implemented on a broad spectrum of devices to establish secure communications. On the high end of the device spectrum are servers and desktop computers followed by tablets and smart-phones. Protocols using conventional cryptographic algorithms may perform well in these devices; therefore, these platforms may not require lightweight algorithms. However, on the lower end of the spectrum are IoT devices such as RFID tags, industrial controllers, WSN, smart cards, etc., which are ubiquitous in today's dynamic communications ecosystem. In fact,

lightweight cryptography is primarily focused on such resource constrained IoT devices that constitute the lower end of this spectrum [101].

5.2 An Adaptive Security Architecture for Lightweight IoT Ciphers

5.2.1 Algebraic Cryptanalysis

Algebraic cryptanalysis is an extremely successful technique for evaluating the security robustness of block and stream lightweight ciphers in Iot devices as well as a threat to the structures which are resistant to other types of attacks. Algebraic attacks have been proven to be effective against lightweight ciphers that include combining or filtering Boolean function along with linear part. Also, highly successful attacks have been staged against lightweight ciphers with non–linear components with or without memory[102]. Considering any secure lightweight IoT cipher, algebraic cryptanalysis consists of mainly two steps: First, one must convert the cipher and possibly some supplemental information (eg. file formats) into a system of polynomial equations, usually over $\mathbb{GF}(2)$, but occasionally over other group rings. Second, one must solve the system of equations and derive the secret key of the given cipher from the solution.

5.2.2 Algebraic Attack on IoT based Ciphers

Algebraic attack focuses on formulating multivariate polynomial equations between inputs and outputs with low algebraic degree. Algebraic attack against a function $\mathbb{F} : \{0,1\}^n \mapsto \{0,1\}^m$ start with an output $\{y_k\}_{k=0}^{m} = \{y_{m-1}, \ldots, y_1, y_0\}$ (presumably in the image of $\mathbb{F}$) and use it to initialize a system of polynomial equations over the hidden input variables $\{x_k\}_{k=0}^{n} = \{x_{n-1}, \ldots, x_1, x_0\}$. The system is further manipulated and extended (e.g.,by multiplying the polynomials by some low–degree polynomial) until a solution is found (e.g.,

linearization and Gaussian elimination or by computing a Gröbner basis of the expanded system[103], [102] and [104]). The significance of the attack is that the formulae exist with probability of 1 or close to 1, unlike traditional probabilistic attacks such as differential cryptanalysis[105] and linear cryptanalysis[106]. Subsequently, solving such equations successfully shall definitely yield the desired value of the targeted variable. Applications of this idea were first introduced in [107] and [108] to break public key scheme. Eventually, the attack was generalized and applied on stream ciphers and block ciphers[109] and [110]. One approach in algebraic cryptanalysis is to build a system of linear equations in the key variables using extensive preprocessing, such as cube attacks[111, 112, 113, 114] and timing attacks[115] and [116, 117]. Although it is evident that higher degree boosts the difficulty to solve the equations, it is quite common that the number of multivariate equations is usually less than the number of variables. This ensures that Algebraic attacks can solve large systems of low degree polynomial equations with surprisingly low complexity. For example, solving dense random-looking equations of degree 16 in several thousand variables over $\mathbb{GF}(2)$ (which correspond to many types of LFSR-based stream ciphers) can now be done in less than 2^{32} complexity[118].

5.2.3 Modular Addition

Stream ciphers such as SNOW 2.0[119], SOBER-t32[110], and ZUC[120], as well as block ciphers which include IDEA[121, 122], CAST[123], TWOFISH[108], and MARS[124] employ Modular Addition 2^n as an elementary cryptographic module. Typically, it is used for mixing, which combines two data sources to provide security. While the logic Exclusive–OR operation is also often used for mixing, Modular Addition offers better security against Algebraic attack[125] and side–channel attacks[126] because it is partly non-linear in $\mathbb{GF}(2)$. A linear operation in $\mathbb{GF}(2)$, such as Exclusive–OR, can be described by an equation of algebraic degree 1. In [125] it was shown that the algebraic degree of the formulae describing Modular Addition can be reduced to quadratic. At the same time, conditional properties

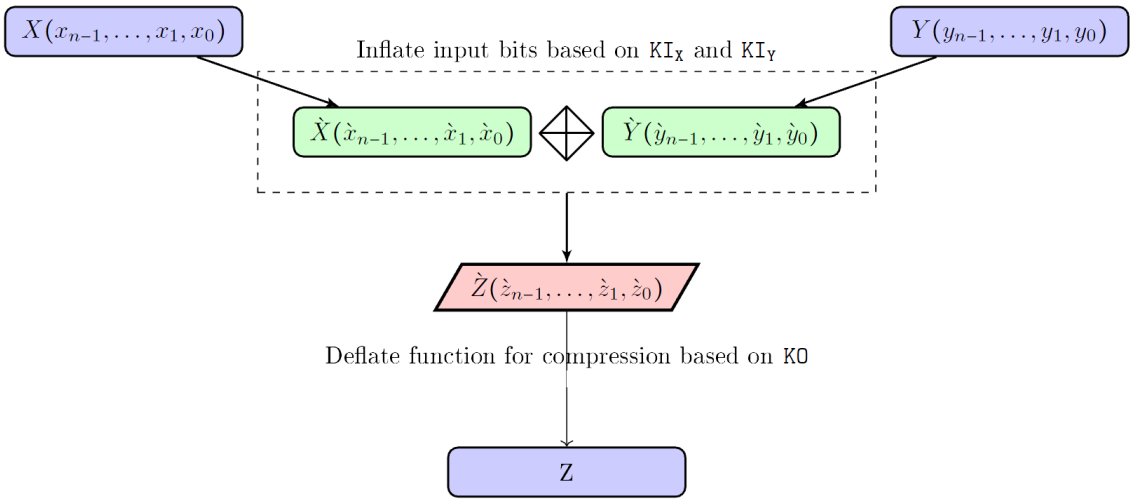


Figure 5.1: Block Schematic Diagram for New Design Framework of Modular Addition

of the Modular Addition have been found to lower the algebraic degree and create new independent equations. These techniques help reduce the complexity of solving Modular Addition significantly.

5.2.4 Security Enhancement for Ciphers in IoT

Our published research work [127, 128] and [129] devises a new design framework of Modular Addition that will increase the algebraic degree when compared to the general Modular Addition and increase the difficulty of using the conditional properties. However the size of the structure is user-defined and flexible, giving its users a scalable security against Algebraic attack (Figure. 5.1). Furthermore, the results in article [128] describes a framework for stream cipher systems only while [127, 129] deals for all types of cipher systems which includes both Stream and Block Ciphers in IoT devices. We briefly outline the major contribution in the following sections for enhancing the security of IoT based lightweight cipher systems which can be realized using both hardware and software[127, 128, 129].

5.3 Modular Addition Preliminaries

Most of the conventional IoT based lightweight ciphers are based on mixing of S–Boxes, arithmetic and Boolean operations. By being one of the fastest arithmetic operations, the Modular Addition as a power of 2 is handled very efficiently on IoT processors. Furthermore we adopt the notation $\boxplus$ for this operation in this article, to distinguish it from the Exclusive–OR operation denoted by $\oplus$.

By considering Modular Addition from the perspective of threats posed by an Algebraic attack, a derivation can be deduced using a set of equations which describes the relation between inputs and outputs. This is outlined in [130] for the n–bit Modular Addition of $Z = X \boxplus Y \bmod 2^n : (x_k, y_k) \mapsto z_k$, and shown in Eq. 5.1

$$\begin{cases} z_k = x_k + y_k & \text{if } k = 0 \\ z_k = x_k + y_k + c_{k-1} & \text{if } k = 1, \text{carry bit } c_k = x_k y_k \\ \vdots & \\ z_{k-1} = x_{k-1} + y_{k-1} + c_{k-2} & \text{if } 2 \leq k < n-1 \end{cases} \tag{5.1}$$

The addition operation is denoted by + sign in $\mathbb{GF}(2)$, which is the logic Exclusive–OR operation. The carry variable obtained above can be described by Eq. 5.2.

$$\begin{cases} c_k = x_k y_k & \text{if } k = 0 \\ c_k = x_k y_k + (x_k + y_k)(c_{k-1}) & \text{if } k = 1 \\ \vdots & \\ c_{k-1} = x_{k-2} y_{k-2} + (x_{k-2} + y_{k-2})(c_{k-2}) & \text{if } 2 \leq k < n-1 \\ \\ \text{In general,} & \\ c_k = x_k y_k + (x_k + y_k)(x_{k-1} y_{k-1}) + & \\ \quad \sum_{q=0}^{k-2} x_q y_q \prod_{r=q+1}^{k} (x_r + y_r) & \text{if } 2 \leq k < n-1 \end{cases} \tag{5.2}$$

By merging Eq. 5.1, Eq. 5.2, we observe that the Modular Addition is not completely non-linear since the least significant bit(lsb) of the resulting output always stays linear. Also the carry terms dominate all other terms of the resulting Algebraic degree. Consequently, the degree increases linearly with the carry terms as in Eq. 5.2. This is due to the fact that the more significantly positioned carry terms not only depend on their corresponding input variables, which have a degree 1, but also on the previous carry terms. As C_0 is generated by X_0 and Y_0, the degree of Z_1 becomes 2. Similarly, C_1 is generated by X_1 and Y_1, and the degree of Z_2 becomes 3. In general, for an n-bit output, we can define the Algebraic degree $\mathfrak{deg}$ for each output bit k as:

$$\mathfrak{deg}(k) = k + 1, \text{ where } 0 < k \leq n \tag{5.3}$$

Thus, the complexity of solving the equations is directly proportional to the Algebraic degree. In [125], a set of equations was devised that describes Modular Addition but limits the

Algebraic degree to 2. This property is described using Eq. 5.4. Moreover the methods in [125] produced $6n-3$ independent equations instead of the original n equations. This effectively reduces the complexity of Algebraic attack on Modular Addition even before the deployment of conditional properties.

$$\begin{cases} z_k = x_k + y_k & \text{if } k = 0 \\ z_k = x_k + y_k + x_{k-1}y_{k-1} & \text{if } k = 1 \\ z_2 = x_2 + y_2 + x_1y_1 + \\ \qquad (x_1 + y_1)(x_1 + y_1 + z_1) & \text{if } k = 2 \\ \vdots & \\ \text{In general,} & \\ z_n = x_{n-1} + y_{n-1} + x_{n-1}y_{n-1} + & \\ \qquad (x_{n-1} + y_{n-1})(x_{n-1} + y_{n-1} + z_{n-1}) & \end{cases} \tag{5.4}$$

5.4 The New Adaptive Lightweight Framework for IoT

In our proposed design[128, 129, 127], a new type of cryptographic model is devised that provides user-defined scalable security against Algebraic attack for Lightweight ciphers in IoT device nodes. Fig.5.1 clearly contrasts our new design framework with the general Modular Addition.

Definition 10 (Inflate Function $\mathfrak{J}_f$)**.** This component expands each single input bit into a $2m$–bit string based on an $n * m$–bit control string KI. The user–define parameter m is determined based on the user's requirement. The input control string KI typically can be generated within a cipher. The actual expansion function can be flexible; the user can substitute other expanding functions instead of the proposed one. For example, the expanding function can be an algebraic function or it can be an S-box (a basic component of symmetric key algorithms which performs substitution).

The proposed expansion function is an arithmetic relationship that is easily scalable. Also, each of its output bit is 0–1 balanced. The Inflate function $\mathfrak{J}_f$ can be described as follows: for a given n–bit word input $X = \{x_k\}_{k=0}^{n} = \{x_{n-1}, \ldots, x_1, x_0\}$ in $\mathbb{Z}$ with the corresponding key input KI_X defined as $KI_X = \{KI_{x_{n-1}}, KI_{x_{n-2}}, \ldots, KI_{x_1}, KI_{x_0} \mid KI_{x_k} \in \{0,1\}^m, 0 \leq k \leq n-1\}$. Consequently for each key input bit, we have $KI_{x_k} \triangleq \{KI_{x_{k,m-1}}, \ldots, KI_{x_{k,1}}, KI_{x_{k,0}}\}$ $\forall$ $KI_{x_{k,j}} \in \{0,1\}$ with $0 \leq k \leq n-1, 0 \leq j \leq m-1$. Furthermore, let $\grave{X}$ be the inflated input where $\grave{X} = \{\grave{x}_k\}_{k=0}^{n} = \{\grave{x}_{n-1}, \ldots, \grave{x}_1, \grave{x}_0\}$ and $\grave{x}_k \in \{\{0,1\}^w \mid w = 2^m\}$ and KI_{x_k} are considered as decimal numbers in Eq. 5.5 as follows. ■

$$\grave{x}_k = \mathfrak{J}_f(x_k,\ KI_{x_k}) \tag{5.5a}$$

$$\text{where, } \mathfrak{J}_f(x_k,\ KI_{x_k}) = \begin{cases} 2^w - 1 - 2^{KI_{x_k}} & \text{if } x_k = 0, \\ 2^{KI_{x_k}} & \text{if } x_k = 1 \end{cases} \tag{5.5b}$$

It is recommended to define an user-defined parameter $m \geq 2$ for Eq. 5.5 in order to avoid repeating values. Fig. 5.2 illustrates an example for $m = 3$.

Definition 11 (Modular Addition)**.** This component of the new framework instantiates modular addition operation using the inputs $\grave{X} = \{\grave{x}_{n-1}, \ldots, \grave{x}_1, \grave{x}_0\} \triangleq \{\grave{x}_{(n-1)(w-1)}, \ldots, \grave{x}_{(n-1)1}, \grave{x}_{(n-1)0}, \ldots, \grave{x}_{1(w-1)}, \ldots, \grave{x}_{11}, \grave{x}_{10}, \grave{x}_{0(w-1)1}, \ldots, \grave{x}_{01}, \grave{x}_{00}\}$ and $\grave{Y} = \{\grave{y}_{n-1}, \ldots, \grave{y}_1, \grave{y}_0\} \triangleq \{\grave{y}_{(n-1)(w-1)}, \ldots, \grave{y}_{(n-1)1}, \grave{y}_{(n-1)0}, \ldots, \grave{y}_{1(w-1)}, \ldots, \grave{y}_{11}, \grave{y}_{10}, \grave{y}_{0(w-1)1}, \ldots, \grave{y}_{01}, \grave{y}_{00}\}$ to perform the operation. These inputs were generated as a result of previous Inflate function component during its expansion process. Consequently, the number of additions to be performed by this particular component has increased from 2^n to $2^{(n)(w)}$ with $w = 2^m$ for some user-defined $m \in \mathbb{Z}^*$. The output for the Modular Addition component is given by $\grave{Z} = \{\grave{z}_{n-1}, \ldots, \grave{z}_1, \grave{z}_0\} \triangleq \{\grave{z}_{(n-1)(w-1)}, \ldots, \grave{z}_{(n-1)1}, \grave{z}_{(n-1)0}, \ldots, \grave{z}_{1(w-1)}, \ldots, \grave{z}_{11}, \grave{z}_{10}, \grave{z}_{0(w-1)1}, \ldots, \grave{z}_{01}, \grave{z}_{00}\}$. In general, we can derive Eq. 5.6 using Eq. 5.4 or Eq. 5.1 and Eq. 5.2 as follows. ■

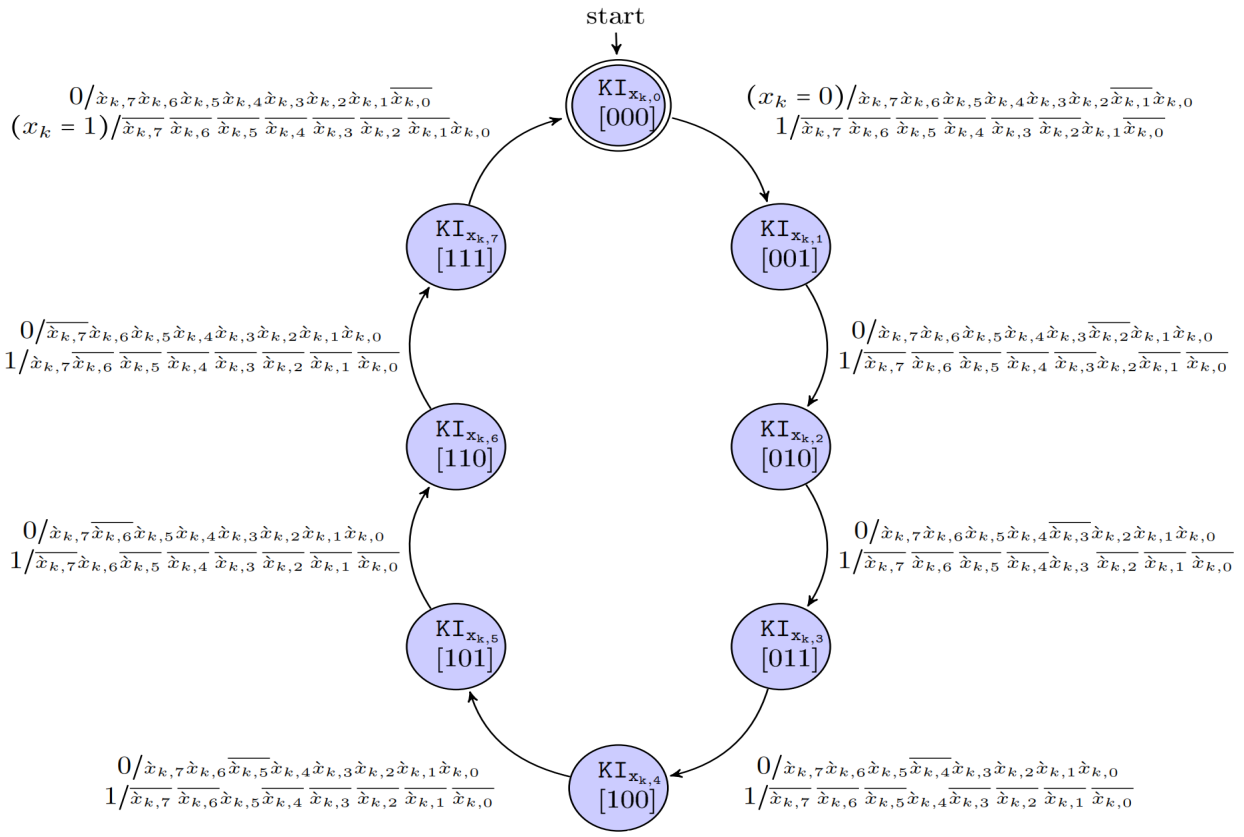


Figure 5.2: Inflate function output states $\dot{x}_{k,j}$ depending on control input $KI_{x_{k,j}}$ for $m = 3$

$$\dot{z}_{kj} = \begin{cases} \dot{x}_{kj} + \dot{y}_{kj} & \text{if } k{=}0,\ j{=}0\ ; \\ \dot{x}_{kj} + \dot{y}_{kj} + \dot{x}_{kj-1}\dot{y}_{kj-1} + & \\ (\dot{x}_{kj-1} + \dot{y}_{kj-1})(\dot{x}_{kj-1} + \dot{y}_{kj-1} + \dot{z}_{kj-1}) & \text{for } \begin{smallmatrix} 0<k\leq n-1 \\ 0<j\leq w-1 \end{smallmatrix} \end{cases} \tag{5.6}$$

Definition 12 (Deflate Function $\mathcal{D}_f$)**.** This function completes the last component of our proposed design. It is a compaction function that compresses $\{\dot{z}_k \mapsto z_k \mid \{0,1\}^{nw} \mapsto \{0,1\}^n \forall\ n, w \in \mathbb{Z}^*\}$ based on a $n*m$–bit control string KO. The Deflate function is highly flexile since the type of compaction method chosen to implement the task of compressing the summation output is solely user–dependent. In our design, we choose a $2^m : 1$ multiplexer (MUX) function. Let $KO = \{KO_{n-1}, \ldots, KO_k, \ldots, KO_1, KO_0 \mid KO_k \in \{0,1\}^m,\ 0 < k \leq n-1\}$.

Thereby we have, $Z = \{z_{n-1}, \ldots, z_1, z_0\} = \{\mathcal{D}_f(\grave{z}_{n-1}, KO_{n-1}), \ldots, \mathcal{D}_f(\grave{z}_1, KO_1), \mathcal{D}_f(\grave{z}_0, KO_0)\}$. Therefore the expression for $\mathcal{D}_f$ can be generalized as in Eq. 5.7a ■

$$z_k = \mathcal{D}_f(\grave{z}_k,\ KO_k) \tag{5.7a}$$

$$\text{where, } \mathcal{D}_f(\grave{z}_k,\ KO_k) = \left\{ \sum_{\wp=0}^{w-1} \grave{z}_{k\wp} \prod_{k=0}^{m-1} (-1)^{\frac{\wp}{2^k}+1} KO_{\wp k} \right. \tag{5.7b}$$

In the equation above, (-1) refers to the complement of $KO_{\wp b}$, summation refers to logic Exclusive–OR operation and multiplication implies logic AND operations.

Example An example is discussed below as in Eq. 5.8a for evaluating Deflate function with $m = 2$, where $\wp$ and k are index variables.

$$\begin{aligned}
\mathcal{D}_f(\grave{z}_k,\ KO_k) &= \sum_{\wp=0}^{w-1} \grave{z}_{k\wp} \prod_{k=0}^{m-1} (-1)^{\frac{\wp}{2^k}+1} KO_{\wp k} \\
&= \sum_{\wp=0}^{4-1} \grave{z}_{k\wp} \prod_{k=0}^{2-1} (-1)^{\frac{\wp}{2^k}+1} KO_{\wp k} \\
&= \grave{z}_{k0}(-1)^1 KO_{\wp 0}(-1)^1 KO_{\wp 1} \\
&\quad + \grave{z}_{k1}(-1)^2 KO_{\wp 0}(-1)^1 KO_{\wp 1} \\
&\quad + \grave{z}_{k2}(-1)^3 KO_{\wp 0}(-1)^2 KO_{\wp 1} \\
&\quad + \grave{z}_{k3}(-1)^4 KO_{\wp 0}(-1)^2 KO_{\wp 1}
\end{aligned} \tag{5.8a}$$

$$\begin{aligned}
\implies \mathcal{D}_f(\grave{z}_k,\ KO_k) &= \\
&= \grave{z}_{k0}\overline{KO_{\wp 0}}\ \overline{KO_{\wp 1}} + \grave{z}_{k1} KO_{\wp 0}\overline{KO_{\wp 1}} \\
&\quad + \grave{z}_{k2}\overline{KO_{\wp 0}} KO_{\wp 1} + \grave{z}_{k3} KO_{\wp 0} KO_{\wp 1}
\end{aligned} \tag{5.8b}$$

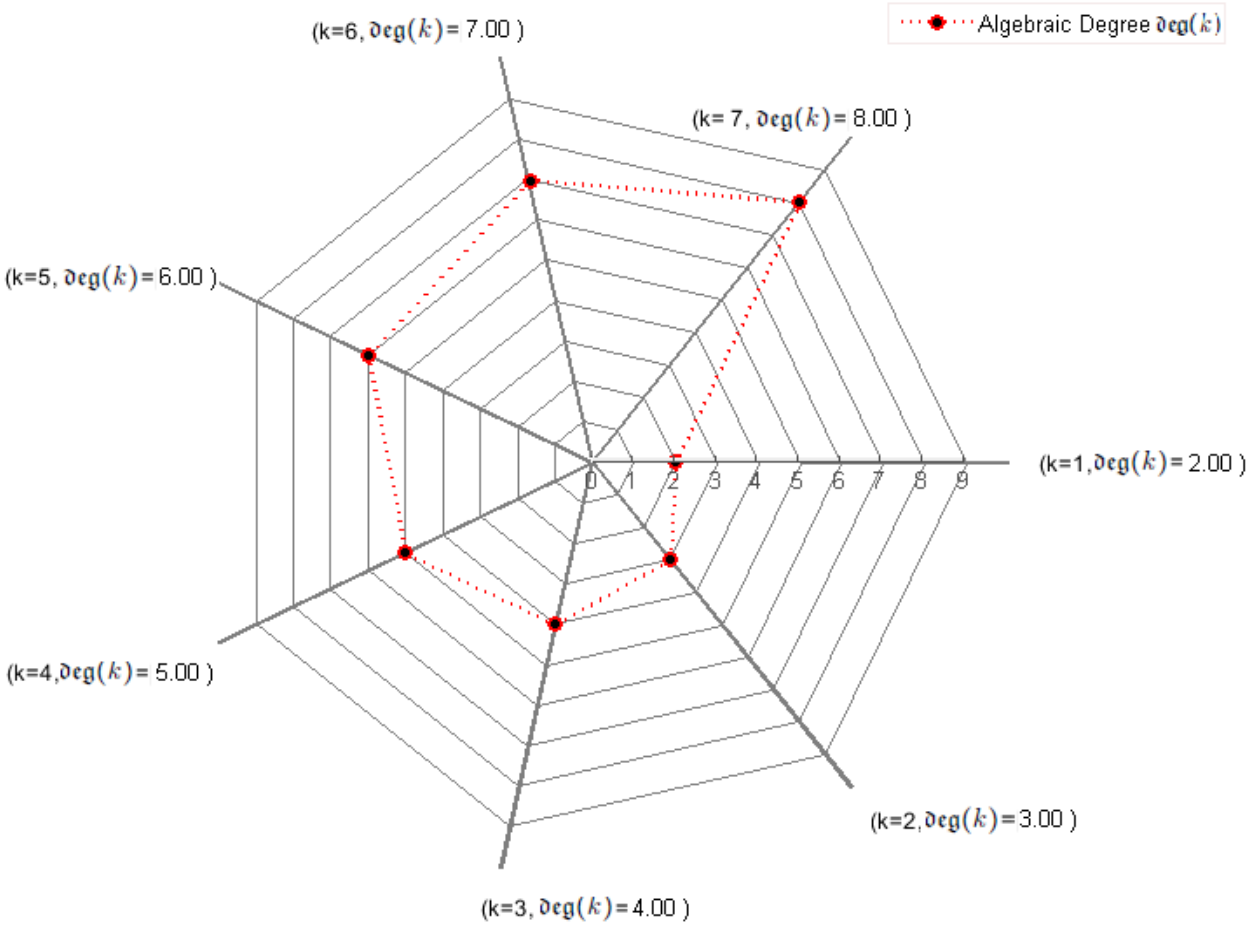


Figure 5.3: Algebraic degree $\mathfrak{deg}(k)$ for general Modular Addition corresponding to every bit k where $1 \leq k \leq 7$, $n = 8$.

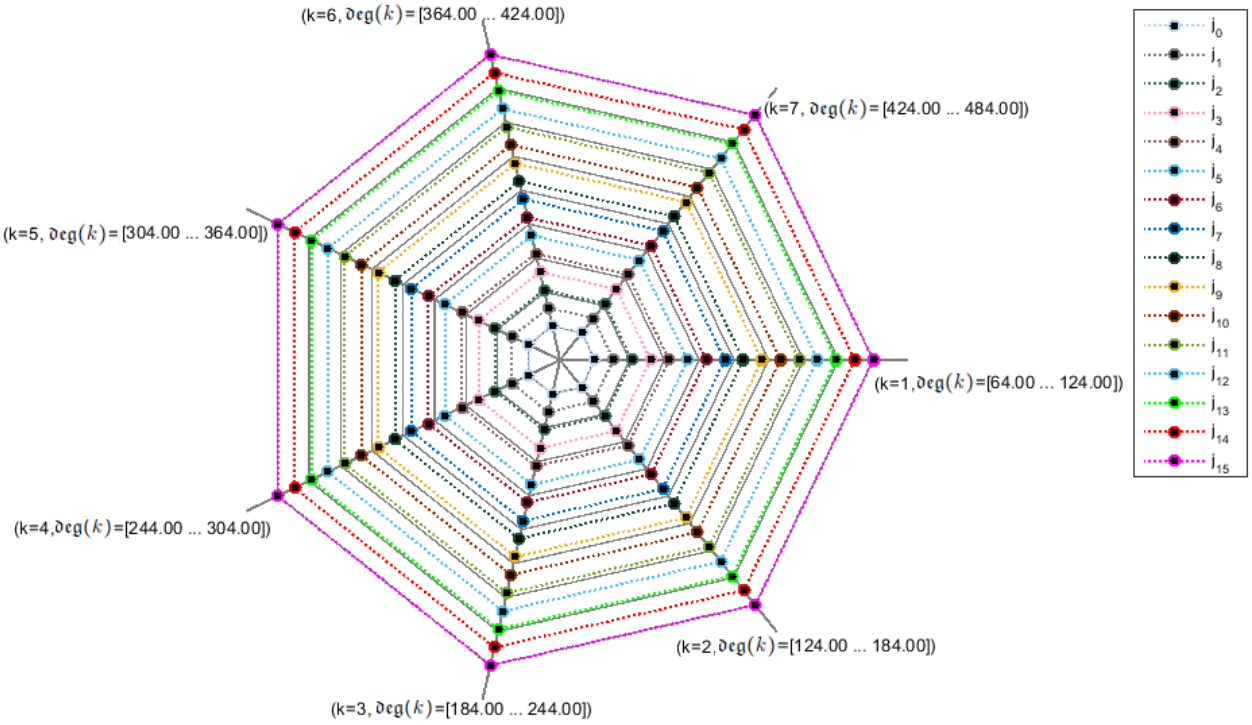


Figure 5.4: An outspread of Algebraic degree $\mathfrak{deg}(k)$ for New Design Framework of Modular Addition corresponding to every bit k where $1 \leq k \leq 7$, $n = 8$ and $0 < j \leq w - 1$, $w = 15$, $m = 4$.

5.5 Design Analysis for New Modular Addition Framework

In this section, we employ Algebraic cryptanalysis to analyze the design of the proposed model.

5.5.1 The Characteristics of the New Design Framework

Output characteristic can be defined for the new design framework by initially considering the general Modular Addition where the output bits can be used directly to derive potential carries and input pairings. In the new design framework however, the Deflate function is lossy; thus, the attacker can only obtain n bits out of 2^{nw} bits even if the output control string KO is known. Therefore, these n bits cannot provide enough information to derive the potential carries and input pairings. It is still possible to have all 1's in the sum of the Modular Addition component in the new design framework. This requires specific combinations of the two m-bit input control strings KI_{x_k} and KI_{y_k}. In particular, the two input control strings should be the same while the corresponding inputs should be a propagate pair. As discussed before, the probability of output being all 1's in a general Modular Addition is 2^{-n}. Thus the probability of this condition occurring in the new design framework is decreased to $(2^{-n})(2^{-mn})$. Apparently the difficulty has increased and the resulting derivation is shown in Section S.II (A) of the electronic appendix supplementary material for this article.

Input characteristic can be evaluated by employing a similar procedure while considering input characteristics of the general Modular Addition component. Initially, the expanded inputs will never be all 0's when using the Inflate function given in Eq. 5.5. Therefore, this characteristic becomes invalid. Nevertheless, it is possible for the expanded inputs to be the Two's Complement of one another. By observing Eq. 5.5 carefully, it is evident that

there are only 3 such cases given any m and $m \geq 2$. Thus, the probability is derived to be $(3/2^{2m+2})$, which is significantly less than 2^{-n}. A more elaborate derivation is shown in Section S.II (B) of the electronic appendix supplementary material for this article.

5.5.2 Carry Absence in Modular Addition

We observe that the probability of carry has decreased from $2^{-(n-1)}$ to $2^{-(wn-1)}$ for the same n-bit input pair as evident in Eq. 5.12. This results in an increase of difficulty for an attacker to create a scenario without any carry, as discussed in Sect. 5.5.1.

5.5.3 Carry Probability

We can estimate the probability of carry for general Modular Addition using Eq. 5.12. As discussed in preceding sections, each bit of the expanded input is $0-1$ balanced. We can view the Inflate function component as an amalgamation of Boolean functions wherein each output bit corresponds to a $\{0,1\}^{m+1} \mapsto \{0,1\}$ function. However in this case, each Boolean function is $0-1$ balanced since the output of the function has an equal chance of producing either a 0 or 1. Using this assumption, the probability of carry for the new design framework can be deduced as given below. The ensuing result produces an equation which is very similar to Eq. 5.12. Suppose the carry bits generated due to the summation of two expanded inputs is given by $\grave{C} = \{\grave{c}_{n-1},\ \ldots,\ \grave{c}_1,\ \grave{c}_0\} \triangleq \{\grave{c}_{(n-1)(w)},\ \ldots,\ \grave{c}_{(n-1)0},\ \ldots,\ \grave{c}_w,\ \ldots,\ \grave{c}_{11},\ \grave{c}_{10},\ \grave{c}_{0w},\ \ldots,\ \grave{c}_{01},\ \grave{c}_{00} \mid w = 2^m, m \in \mathbb{Z}^*\}$, then the probability can be deduced using the examples in (5.9), (5.10) and (5.11).

$$\Pr(\grave{c}_{01} = \zeta) = \left\{ \begin{matrix} \frac{1}{2} * \frac{1}{2} = \frac{1}{4} & \text{if } \zeta{=}1 \\ \\ \frac{3}{4} & \text{if } \zeta{=}0 \end{matrix} \right\} \text{ where } x'_{00} = y'_{00} = 1 \tag{5.9}$$

$$\left.\begin{aligned}\Pr(\grave{c}_{02}=\zeta) &= \tfrac{1}{2}*\tfrac{1}{2}*\tfrac{1}{4}+\tfrac{1}{2}*\tfrac{1}{2}*\tfrac{3}{4}\\ &\quad +\tfrac{1}{2}*\tfrac{1}{4}*\tfrac{1}{2}+\tfrac{1}{2}*\tfrac{1}{4}*\tfrac{1}{2}\\ &= \tfrac{3}{8}\end{aligned}\right\} \iff \begin{cases}\grave{x}_{01}=\grave{y}_{01}=\grave{c}_{01}=1;\zeta=1\\ \grave{x}_{01}=\grave{y}_{01}=1,\grave{c}_{01}=0\\ \grave{x}_{01}=\grave{c}_{01}=1,\grave{y}_{01}=0\\ \grave{y}_{01}=\grave{c}_{01}=1,\grave{x}_{01}=0\end{cases} \tag{5.10}$$

$$\Pr(\grave{c}_{02}=0)=1-\Pr(\grave{c}_{02}=1)=\frac{5}{8} \tag{5.11}$$

By utilizing Equations (5.9), (5.10) and (5.11), we can generalize the probability using Eq. 5.12 as follows.

$$\Pr(\grave{c}_{kj}=\zeta)=\left\{\begin{array}{ll}\frac{2^{k*w+j}-1}{2^{k*w+j+1}} & \text{if } \zeta{=}0\\ 1-\Pr(\grave{c}_{kj}=\overline{\zeta}) & \text{if } \zeta{=}1\end{array}\right\} \begin{array}{l}\text{where } 0\le k\le n-1,\\ 1\le j\le w\ \forall\ w=2^m\end{array} \tag{5.12}$$

5.5.4 The Complexity Analysis for the New Model

The Algebraic degree must be obtained in order to evaluate the complexity of solving the new design framework under an Algebraic attack. A cryptosystem can be seen as a set of Boolean functions, each of which is represented as a polynomial modulo 2 in what is known as its Algebriac Normal Form (ANF). Therefore, the new design framework is expressed using ANF, which describes a Boolean function using logic Exclusive–OR gates[103]. The Algebraic degree of each component is first studied before considering the degree of whole design.

Algebraic Degree of Inflate Function $\mathfrak{J}_f$ is a monomial with the largest degree in the Algebraic normal form. With regard to the Inflate function, each expanded variable can be expressed in the ANF by considering itself a Boolean function. Intuitively, the value of the expanded variable is a manipulation of the original input value based on the value of the

user-defined parameter m. By carefully observing the 2 examples outlined in the Table 5.1, it can be deduced that Algebraic degree relates directly to the value of user-defined parameter m.

Effective Algebraic Degree of Modular Addition can be evaluated for the new model using Eq. 5.4 or Eq. 5.1 and Eq. 5.2. We notice that Eq. 5.4 limits the algebraic degree to quadratic in the original Modular Addition by utilizing the output variables. This is under the assumption that output is observable. However, in the new design framework, the output variables of the Modular Addition component may not be observable. Moreover, it is possible to define them as additional variables so that the algebraic degree of the expression can be reduced. The drawback to this method is that the number of variables used to solve the set of equations has increased. Assuming that additional variables are used, the algebraic degree of the Modular Addition component is at most $2m$. This is due to the fact that each input variable now has a degree of m and the largest degree is quadratic using Eq. 5.4. At this point, it can be observed that the algebraic degree has already increased by the user-defined parameter m.

Subsequently, it is possible to express the Modular Addition using Eq. 5.1 and Eq. 5.2, with its algebraic degree being outlined by Eq. 5.3. As discussed before, currently each input variable has a degree of m. We outline the following notion while evaluating individual bits for any given n–bit word: the Algebraic degree of the least significant bit(lsb) is represented by $\text{lsb}_{\mathfrak{Deg}}$ and the rest of the bits by $\sim\text{lsb}_{\mathfrak{Deg}}$ where as the combined bits are denoted by a tuple as $\langle \text{lsb}_{\mathfrak{Deg}}, \sim\text{lsb}_{\mathfrak{Deg}} \rangle$. Considering addition operation we have, $\text{lsb}_{\mathfrak{Deg}} = m$ and for rest of the output bits, it is $\sim\text{lsb}_{\mathfrak{Deg}} = (k * w + j + 1)m$ where $0 \leq k \leq n-1, 0 \leq j \leq w-1, w = 2^m$. The derivation approach is similar to what is outlined in the preceding section. It can be noticed that the degree of the carry terms increases linearly according to their bit positions. However, the degree increases in multiples of m because the expanded input variables have a degree of m. As a result, the degree of $\grave{z}_{01}$ is generated by the multiplication of two degree-m variables $\grave{x}_{00}$ and $\grave{y}_{00}$. Similarly, the degree of $\grave{z}_{02}$ can be generated by the multiplication

of $\grave{x}_{01}$, $\grave{x}_{00}$, and $\grave{y}_{00}$, or the combination of $\grave{y}_{01}$, $\grave{x}_{00}$ and $\grave{y}_{00}$ whose degrees are $2m$ and $3m$ respectively. Therefore, each output variable of the Modular Addition, $\grave{z}_{kj}$, has a degree of $(k * w + j + 1)m$. A comparison summary of algebraic degree for the summation operation is illustrated in Table. 5.2. Therefore, by comparison the effective increase of algebraic is m with regard to the general Modular Addition.

Algebraic Degree of Deflate function $\mathfrak{D}_f$ As specified in preceding sections, this function is a 2m:1 logic Multiplexer (MUX) function defined by Eq. 5.7a. As this equation is itself in the Algebraic normal form (ANF), the degree can be determined by observing Eq. 5.7a. Consequently, the degree is $m + 1$ since the output of the MUX function depends on the values of all the select lines and the input. The 1 comes from the assumption that the degree of input to the MUX is 1. When the degree changes, it must be substituted accordingly.

5.5.5 Overall Algebraic degree of New Design Framework

This can be estimated by accruing the degrees of all the components for our new model. Table. 5.3 provides a summary of the algebraic degree of the new design framework and a comparison to the general Modular Addition. Here the algebraic degree of the general Modular Addition is calculated using Eq. 5.4. As summarized in Table. 5.4, the Algebraic Immunity has increased by $2m$, or at least 4 for $m = 2$. Moreover, the Describing Degree has increased from 2 to at least 66 for $m = 2$ and $n = 8$ bits. Also, it is worth noting that an attacker can seek to lower the degree of the new design framework by looking for additional independent equations with lower degree or by creating extra variables. The benefit of these methods is to be determined by the attacker. Considering this aspect, a corner case study is provided in Sect. 5.5.6 as a starting point.

An illustration of a comparative analysis of Algebraic degrees between the general Modular Addition and our New Design Framework is depicted using Figures. 5.3 and 5.4 for $m = 4,\ n = 8,\ w = 2^4 - 1$. These radar plots act as valuable tools in comparative analysis

of multivariate data. In the plot, each spoke denotes individual bits $k > 0$ or non–lsb bits whereas the data value points on each spoke represents the additional j component of inflate function. It is intuitively clear from Figure. 5.4, that the Algebraic degree obtained by drawing a unique colored line connecting the data values for each spoke has an outward spread due to its expansive property. Subsequently, the highest Algebraic degree corresponds to the point where the subscripts k, j of the expanded word are at their maximum values i.e., $i = 7$ and $j = 15$ thereby, completing the cycle in an ascending order.

5.5.6 A Corner Case Analysis for the New Design Framework

A corner case for the new design framework can be derived by carefully observing for a conditional property; a probabilistic condition can help reduce the Algebraic degree. As described in Sect. 5.5.4 and inferring example from Table. 5.1, it is clear that the Algebraic degree of the Inflate function depends on the multiplication of the input control string variables. Therefore, if the variables are all known, the degree falls to 1. Specifically, if the

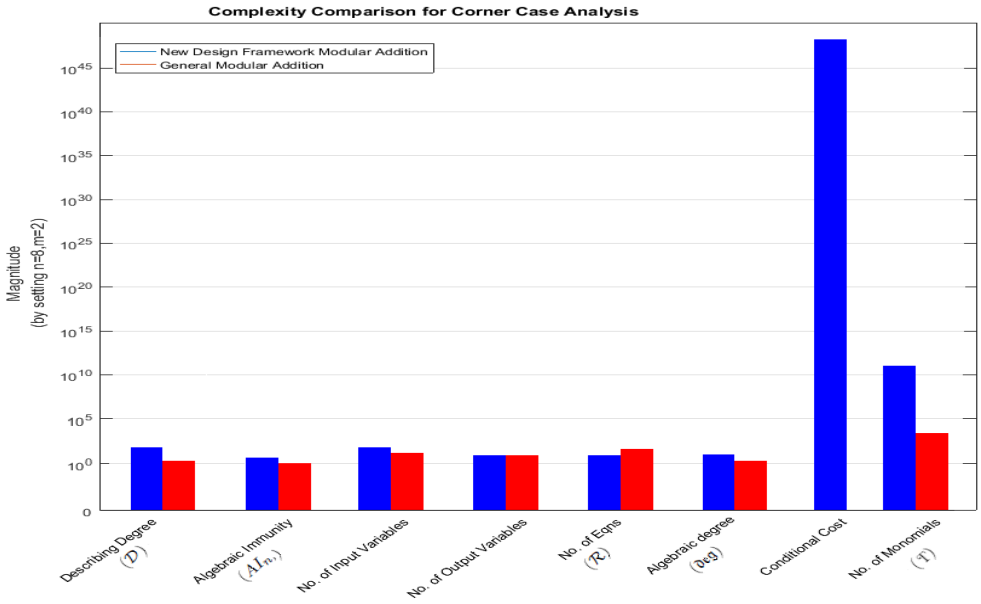


Figure 5.5: Differentiating various properties using Corner Case Analysis with $m = 2$ and $n = 8$.

input control string has all 0's, the expanded inputs are either the same as the inputs or the complement of the inputs. Under this condition, the degree of addition becomes at least 1, which is the same as the general Modular Addition. In each block of expanded inputs, the expression of the summation of expanded input variables can be reduced because many of the variables are the same. In fact, $\text{lsb}_{\mathfrak{Deg}}$ in each block of inflated inputs is $k+1$, for $0 \leq k \leq n-1$. The rest of the summation bits $\sim\text{lsb}_{\mathfrak{Deg}}$ obtained from adding each block of the inflated inputs have a degree of $k+2$. An example of the derivation is shown in Section S.I as part of the electronic appendix supplementary material for this article using $m = 2$. Furthermore, the attacker would notice that if the Deflate function is able to select the lsb in each block of the summation, i.e., $\grave{z}_{k0}$, the Algebraic degree is the lowest. In order to recreate this condition, the output control string needs to be all 0's. As a result, the degree of the new design framework becomes $k+1$ for $\grave{z}_k$ with $0 \leq k \leq n-1$. Incidentally, this is the same as the general Modular Addition as shown in Table. 5.2. The cost of this condition has a probability of 2^{-3mn} as all control bits need to be 0's. The general Modular Addition and our new design framework can be viewed as S-Boxes and their complexity against Algebraic attack can be approximated as S-Boxes. For the general Modular Addition, the required parameters have been studied in [125]. A comparison for the same has been listed in Table. 5.4. While referring to the corner case, the number of monomials is still larger because of the increased number of variables and Algebraic degree. Simultaneously, the complexity has also increased by attaching the conditional cost. A comparative bar graph is shown in Figure. 5.5 distinguishing various corner case properties between general Modular Addition and our proposed New Design Framework of Modular Addition.

Input ($\grave{x}_k$)	Algebraic Degree ($\mathfrak{deg}$)	Algebraic Normal Form (ANF)
	CASE m = 3	
$\grave{x}_{k0}$	3	$\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k1}$	3	$1 \oplus \mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k2}$	3	$1 \oplus \mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k3}$	3	$1 \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k4}$	3	$1 \oplus \mathrm{KI}_{x_2} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k5}$	3	$1 \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k6}$	3	$1 \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k7}$	3	$1 \oplus \mathrm{KI}_{x_2}\mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
	CASE m = 2	
$\grave{x}_{k0}$	2	$\mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k1}$	2	$1 \oplus \mathrm{KI}_{x_0} \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k2}$	2	$1 \oplus \mathrm{KI}_{x_1} \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$
$\grave{x}_{k3}$	2	$1 \oplus \mathrm{KI}_{x_1}\mathrm{KI}_{x_0} \oplus x_i$

Table 5.1: ANF Case Analysis for resultant $\mathfrak{J}_f$ Outputs.

Algebraic Degree	Modular Addition	Modular Addition based on New Design Framework
$\mathfrak{deg}$	$\langle \mathrm{lsb}_{\mathfrak{deg}}, \sim\mathrm{lsb}_{\mathfrak{deg}} \rangle$	$\langle \mathrm{lsb}_{\mathfrak{deg}}, \sim\mathrm{lsb}_{\mathfrak{deg}} \rangle$
Using Eq. 5.4	$\langle 1, 2 \rangle$	$\langle m, 2m \rangle$
Using Equations (5.1) and (5.2)	$\langle 1, k+1 \rangle$	$\langle m, (k*w+j+1)m \rangle$

Table 5.2: Comparative Summary of Algebraic Degrees for Summation Component.

Algebraic Degree	Modular Addition	Modular Addition based on New Design Framework
$\mathfrak{deg}$	$\langle \text{lsb}_{\mathfrak{deg}}, \sim\text{lsb}_{\mathfrak{deg}} \rangle$	$\langle \text{lsb}_{\mathfrak{deg}}, \sim\text{lsb}_{\mathfrak{deg}} \rangle$
Inflate Function $\mathfrak{J}_f$	$\langle 1, 1 \rangle$	$\langle m, m \rangle$
Modular Addition	$\langle 2, 2 \rangle$	$\langle m, m + (i * w + j + 1)m \rangle$
Deflate Function $\mathfrak{D}_f$	NA	$\langle m + 1, m + 1 \rangle$
Total	$\langle 1, 2 \rangle$	$\langle 2m, m + (i * w + j + 1)m \rangle$

Table 5.3: Exemplifying differences in Algebraic Degree for Generic and New Design Framework of Modular Addition.

Table 5.4: Complexity Evaluation using Corner Case Analysis

Type	Describing Degree ($\mathfrak{D}$)	Algebraic Immunity (AI_n)	Input Variables	Output Variables	Extra Variables	Equations ($\mathcal{R}$)	Algebraic Degree ($\mathfrak{deg}$)	Condition Cost	No. of Monomials ($\mathfrak{T}$)	Complexity (Γ)
Modular Addition	2	1	$2n$	n	0	$6n-3$	2	0	$\sum_{k=0}^{2} \binom{3*n}{k}$	$(\mathfrak{T}/2n)^{\lceil \mathfrak{T}/\mathcal{R} \rceil}$
New Design Framework Modular Addition	$m + ((n-1) * 2^m + (2^m - 1) + 1)m$	2m	$3mn + 2n$	n	0	n	$n+1$	2^{3mn}	$\sum_{k=0}^{n+1} \binom{3n(1+m)}{k}$	$(\mathfrak{T}/3mn + 2n)^{\lceil \mathfrak{T}/\mathcal{R} \rceil}$

5.6 Case Study

In this section, we demonstrate the application of our proposed new design framework using a stream cipher in case study. Further case studies including analysis on block ciphers have been elaborated in our articles [].

5.6.1 New SNOW 2.0

A stream cipher like SNOW 2.0, utilizes combiner with memory[131, 132] or formally, (k, l)–combiners in this example. A (k, l)–combiner $\mathcal{C} = (f, \phi)$ with k inputs and l memory bits is a finite state machine (FSM) which is defined by output function $f : \{0,1\}^l$ x $\{0,1\}^k \mapsto \{0,1\}$ and a memory update function $\phi : \{0,1\}^l$ x $\{0,1\}^k \mapsto \{0,1\}^l$. Given a stream $(x_0, x_2, \ldots x_{n-1})$ of inputs, $x_i \in \{0,1\}^k$, and an initial assignment $Q_1 \in \{0,1\}^l$ to the memory bits, the corresponding output bitstream $(z_0, z_1, \ldots z_{n-1})$ is defined according to z_k=$f(Q_k, X_t)$ and $Q_{k+1} = \phi(Q_k, X_k)$, where $0 \leq k \leq n-1$.

Overview of SNOW 2.0

This particular cipher uses a length of 16 LFSR over $\mathbb{GF}(2^{32})$. In other words, the LFSR has 16 elements, or states, but each state contains a 32-bit word. Let $S_0, S_1, \ldots, S_{15}$ denote the states of the LFSR. The feedback function is defined as the XOR combination of S_0 multiplied by α, S_2 and S_{11} divided by α. To produce the output key stream, a Finite State Machine (FSM) is used in conjunction with the LFSR. The FSM contains two 32–bit registers R1 and R2. The value of R2 is determined by feeding the value of R1 through a set of AES S–boxes and the AES Mix Column function. The value of R1 is determined by performing Modulo Addition 2^{32} between R2 and S_5. Finally, the output combiner function is defined as first performing Modulo Addition 2^{32} between R1 and S_{15}, then perform the Exclusive–OR operation on the result with R2, and finally XOR–ing the result of the former with S_0. It operates in two modes: Initialization and Key Stream Generation. Succinctly,

SNOW 2.0 uses a 128-bit secret key and a 128-bit initialization vector during Initialization Mode.

Application using the New Design Framework

The new design framework is used to replace the two Modulo Additions and the user-defined parameter m is chosen to be 3. There are 288 extra bits required to supply the input and output control strings of each addition, because for each input bit of the 32-bit addition, a 3-bit control string is needed. Therefore, 576 bits in total are required for two additions. The extra bits can be generated in my ways. In this case, S_{14} is used to generate 288 bits and the same set of bits is used for the two insertions of the new design framework. The logic behind this generation is as follows:

1. For each bit of the first input X, a total of 3 input control bits are needed. They will be the 3 LSBs of the 3-bit circular-left-shifted S_{14}. For example: $KI_{x_0} = (S_{14,2}, S_{14,1}, S_{14,0})$ and $KI_{x_1} = (S_{14,31}, S_{14,30}, S_{14,29})$.

2. For each bit of the second input Y, the 3 input control bits will propagate from the 3 LSBs of the 3-bit circular-right-shifted and inverted S_{14}. Let S'_{14} denote the bit-wise inverted S_{14}. Then, $KI_{y_0} = (S'_{14,2}, S'_{14,1}, S'_{14,0})$ and $KI_{y_1} = (S'_{14,5}, S'_{14,4}, S'_{14,3})$.

3. For the output control string, each 3 output control bits arrives from the 3 LSBs of the 3-bit circular-right-shifted S_{14}. For example: $KO_0 = (S_{14,2}, S_{14,1}, S_{14,0})$ and $KO_1 = (S_{14,5}, S_{14,4}, S_{14,3})$.

This setup can at least guarantee that the input control bits for the first input pair will not be all 0's simultaneously. The new SNOW 2.0 setup is shown in Figure. 5.6. An example output is shown in Section S.III of the electronic appendix supplementary material for this article to demonstrate the internal steps of applying the new design framework on S_{15} and R1 during the generation of the first output key stream. The secret key is defined as 0xAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA. The initialization vector(IV) is defined

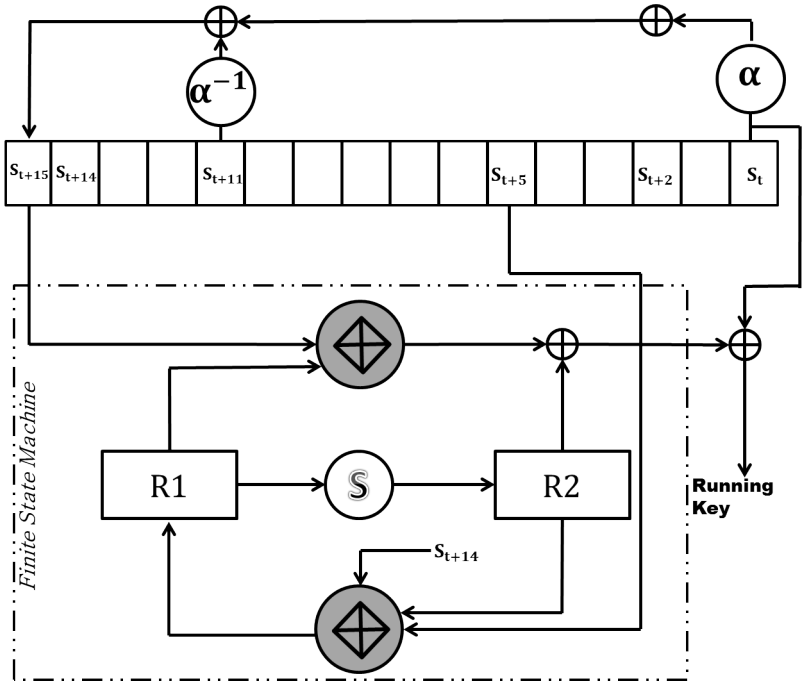


Figure 5.6: Schematic for the New SNOW 2.0 Design

as 0x00000004000000030000000200000001. This is one of the test vectors listed in [132]. At this specific timeframe, S_{15} = 0xCC15A50B, $R1$ = 0xAAB91A68, and S_{14} = 0x5164B6D 9. Consequently, the output of the new design is 0x37F7B4F7 while the original Modulo Addition gives 0x76CEBF73. Also, the first output key stream of new SNOW 2.0 is 0x91C C022F and the original key stream is 0xC355385D.

5.6.2 Analysis of New SNOW 2.0

The Algebraic Attack on SNOW 2.0 has been studied extensively in [133] and [125]. Two methods have been proposed to linearize the Modulo Addition 2^{32} in the stream cipher. The first method is relatively straightforward, as the Modulo Addition can be completely linearized when there are no carries. The probability of this occurring can be estimated using Equation. (5.12). The condition is satisfied as long as each input pair does not generate a carry. The probability of this happening is $(3/4)^{31}$ because the probability of an input pair

Type	By Method 1	By Method 2	Corner Case
	Fix Carries to 0	Fix Consecutive Outputs	
SNOW 2.0	2^{-248}	2^{-288}	--
New SNOW 2.0	2^{-4216}	2^{-1768}	NA

Table 5.5: Analysis of Results for the New SNOW 2.0.

generating no carries is (3/4). The author in [133] seeks to use this condition for both additions and for 17 consecutive cycles. The probability of this is $(3/4)^{31*2*17} \approx 2^{-438}$, which is close to exhaustive search 2^{-576}. In SNOW 2.0, the exhaustive search includes the search for 512 bits in the LFSR states and two 32-bit registers. In the new SNOW 2.0, the cost of having no carries has greatly increased. As $m = 3$ in this application, the length of the Modulo Addition component in the new design framework becomes $32 * 2^3 = 256$. To fix the carries for one Modulo Addition, the probability is estimated to be $2^{-(31*8*17)} = 2^{-4216}$ by using Equation. (5.12). This is much larger than exhaustive search. The second method sees the attacker trying to manipulate the output characteristics of the Modulo Addition to linearize the equations, as described in [125]. In particular, 9 consecutive values of the register R1 are fixed. The desired output values from the summation are $R1_1 = 0$, $R1_2 = 2^{32} - 1$, $R1_3 = 0$, $R1_4 = 0$, $R1_5 = 0$, $R1_6 = 0$, $R1_7 = 0$, $R1_8 = 0$, and $R1_9 = 0$. The value of R1 comes from summing R2 and S_5 but the value of R2 comes from feeding R1 through the ASE S-Boxes and Mix Column operation. Therefore, only S_5 needs to be fixed. Due to the nature of LFSR, 9 states need to be fixed, namely: S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}, S_{12}, and S_{13}. The associated probability is $2^{-32*9} = 2^{-288}$. With the new design framework applied; however, the output characteristic may not be applicable. As discussed in Section. 5.5, the probability of fixing all outputs to be 1 in the new design is $2^{-n(m+1)}$. In this scenario, the probability has become $2^{-32(3+1)} = 2^{-128}$. In addition, the probability of fixing all outputs to be 0 in the new design is $(3/2^{2m+2})^n$. Again, the probability becomes $(3/2^{2*3+2})^{32} \approx 2^{-205}$. For a total of 9 consecutive cycles, the probability has become $2^{-205*8} * 2^{-128} = 2^{-1768}$. In

essence, the adversary may want to utilize the corner case of the new design framework to lower the Algebraic degree. However, the control string generation logic, outlined in Section. 5.6, guarantees that the input control strings for the LSBs of the two inputs will not be 0 simultaneously. Therefore, the set of equations cannot be completely linearized, as illustrated in Table. 5.5.

5.7 Conclusion

In this chapter, a new type of model for Modulo Addition was proposed to defend against attacks involving multivariate polynomials for lightweight ciphers involving both Stream an Block type in IoT communication nodes. Our model features three components: an Expandable Input, Modulo Addition, and a compression module called Deflate function. Furthermore our new design framework utilizes an expanding and compacting structure that can fit into various lightweight cryptographic systems based on user–based requirements depending on the computational power of specific IoT nodes. Further case studies involving lightweight ciphers for IoT such as SNOW 2.0[119] and IDEA[121, 122] ciphers were elaborated in [128, 129, 127]. This concludes potential security enhancements for IoT in this chapter.

Chapter 6

Conclusion and Contributions

6.1 Summary

We can summarize from the research work described in previous chapters as follows. In Chapter 1, we discussed the gargantuan volume of data created by the IoT ranging from devices in home to city infrastructure will have an unfathomable impact on the networking systems used today, and managing them will be equally challenging. Furthermore in Chapter 2, MOO algorithms have been a subject of intense interest to IoT researchers for solving diverse multi-objective optimization problems, in which multiple objectives are treated simultaneously subject to a set of constraints. However, it is infeasible for multiple objectives to achieve their respective optima at the same time, thus there may not exist a single globally optimal solution, which is the best with respect to all objectives. We concluded that GWO is an algorithm which is best suited considering the diverse optimization requirements of an IoT. Also Chapter 5 discussed ways to enhance security of IoT nodes without compromising the performance requirements and stated cost overhead of replacements is a significant challenge in IoT domain. We proposed a new type of model for Modulo Addition was proposed to defend against attacks involving multivariate polynomials for IoT based lightweight ci-

phers involving both Stream an Block ciphers without added overheads. A novel High-level synthesis (HLS) is described in Chapter 4 that comprises of a reliability based design process that interprets an algorithmic description of a desired behavior and creates digital IoT specific hardware realizing the required behavior was formulated. The comparative analysis concluded the superiority of our methodology. Ultimately in Chapter 3, a novel delta diagram based parameter capture method to synthesize the network of resource deprived IoT communication systems was initiated using optimization framework described in Chapter 2. The final analysis concluded the point-to-point optimization benchmarks in comparison with state-of-the-art methods.

For a more comprehensive review of the contributions made towards IoT research areas in this thesis, we suggest the reader to follow through the main research highlights listed in the next section.

6.2 Contributions

This research makes the following contribution.

6.2.1 Security enhancement methods for IoT

Our research proposed a new type of model for Modulo Addition was proposed to defend against attacks involving multivariate polynomials for lightweight ciphers involving both Stream an Block type in IoT communication nodes. Our model features three components: an Expandable Input, Modulo Addition, and a compression module called Deflate function. Furthermore our new design framework utilizes an expanding and compacting structure that can fit into various lightweight cryptographic systems based on user–based requirements depending on the constrained computational power of specific IoT nodes.

6.2.2 Framework for Rapid DSE of IoT device

The thesis presented a new framework for an accelerated design space exploration of resource constrained IoT hardware. The approach was successful in laying the foundation for exploring the design points from the architecture design space according to the performance objective and intended functionality. The novel method determines the utility coefficient of each resource for designing system of IoT hardware. After the architectural design specification were organized in sorted order based on the utility coefficients calculated, the procedure for applying the optimizer algorithm became easier. As a result the proposed approach was able to drastically reduce the number of architectural variants to be analyzed for selection of the system architecture. The proposed mechanism for DSE was able to resolve conflicting objectives in DSE in HLS by utilizing a novel Grey Wolf Optimizer.

6.2.3 Grey Wolf Optimizer

We proposed a Grey Wolf Optimizer to solve the multi-objective problem with conflicting parameters for DSE approach with the architecture synthesis process useful for micro scale devices that are commonly employed by IoT based requirements. Thus it ensured that small, rugged, inexpensive and low powered IoT sensors will bring the IoT to even the smallest objects installed in any kind of environment, at substantially lower costs.

6.2.4 Automated Communication Synthesizer

We design and develop that uses a novel Delta Diagram based model for rapid choosing of IoT network layer parameters to initiate a cross-layer optimization. Furthermore, the optimization process utilizes our Grey wolf optimizer to optimize Quality of Service (QoS) requirements in terms of delay, energy consumption and reliability requirements of an IoT communication network.

6.3 Future Work

From the communication systems point of view, in this thesis, we have explored the interaction among functionalities across the different layers in protocol stack and developed a novel cross-layer optimization framework for the IoT using **Delta Diagram** synthesis. However, the proposed synthesis can be further expanded to any communication network (eg., Cellular GSM Networks, Bluetooth Personal Area Networks, etc.) to be optimized by simply incorporating **Delta Diagram** for other layers of the protocol stack and deriving mathematical models for them. The fundamental part of **Delta Diagram** based framework is to obtain the optimal routing paths and the communication parameters among things, by exploiting the interrelations among different layer functionalities in the IoT. This novel methodology of finding parameter paths while optimizing the layers in form of delta diagram defined in Figures 3.2, 3.3 and 3.4, can be exploited in next generation networks such as 5G cellular communications, WiMAX network, etc.. Furthermore the **Delta Diagram** approach can be a ideal optimizing framework for next generation intra-satellite networking paradigms where research is still in nascent stage.

From the perspective of IoT hardware optimization, there is further scope in the area of design space exploration and high level synthesis to improve the search time for finding the final design architecture, and thereby accelerate the speed of the exploration process. The developed design space exploration approach for high level synthesis can be improved further by decreasing the search time required during the exploration process. This can be realized by experimenting with parallel version of Grey wolf optimizer algorithms and its variants.

Additionally, the security enhancements described in this thesis for IoT ciphers can further be expanded to incorporate optimization of the user–defined parameter under different performance and resource constraints. In addition, the realization of the lightweight cipher design in software and hardware can be optimized as well. Other functions can be developed to substitute the existing functions in the Inflate function and deflate function to provide different or extra cryptographic features.

The next paradigm change at the door of Information and Communication Technology is IoT. IoT, as an evolving technology, is gaining rapid popularity because of its basic idea: accessing the physical objects through Internet. It has number of inbuilt benefits as discussed in Chapter 1, but there are numerous factors and bottlenecks discussed in this thesis which are delaying the worldwide implementation of this technology. There are many on-going world-wide research initiatives and standardization efforts, which aim at making the IoT a reality. Unfortunately, the nature of very high heterogeneity in hardware capabilities of things and QoS requirements for different applications throttles the performance of classical layered protocol solutions and the existing cross-layer solutions for wireless sensor networks or ad hoc networks.

Bibliography

[1] Saumya Bhadauria and A Gupta. Adaptive bacterial foraging driven datapath optimization: Exploring power-performance tradeoff in high level synthesis. *Applied Mathematics and Computation*, 269:265 – 278, 2015.

[2] KNect365. Internet of things conference for industrial markets. `https://tmt.knect365.com/iot-world-europe/`, January 2017.

[3] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Adv. Eng. Softw.*, 69:46–61, mar 2014.

[4] Javier Martinez. The internet of things will cause ip traffic to skyrocket 300% by 2018. `https://www.techradar.com/news/internet/broadband/the-internet-of-things-will-cause-ip-traffic-to-skyrocket-300-by-2018-1252812`, August 2017.

[5] Internet of things definition- postscapes. *Internet of Things Definition- Postscapes*, Apr 2015.

[6] Guido Noto La Diega and Ian Walden. Contracting for the internet of things: Looking into the nest. *Queen Mary School of Law Legal Studies Research Paper No. 219/2016*, Feb 2016.

[7] Nouha Baccour, Anis Koubâa, Luca Mottola, Marco Antonio Zúñiga, Habib Youssef, Carlo Alberto Boano, and Mário Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw.*, 8(4):34:1–34:33, September 2012.

[8] Jin-A Park, Seung-Keun Park, Dong-Ho Kim, Pyung-Dong Cho, and Kyoung-Rok Cho. Experiments on radio interference between wireless lan and other radio devices on a 2.4 ghz ism band. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, volume 3, pages 1798–1801 vol.3, April 2003.

[9] Luca Mottola, Gian Pietro Picco, Matteo Ceriotti, Ştefan Gună, and Amy L. Murphy. Not all wireless sensor networks are created equal: A comparative study on tunnels. *ACM Trans. Sen. Netw.*, 7(2):15:1–15:33, September 2010.

[10] O D Incel, S Dulman, P Jansen, and S Mullender. Multi-channel interference measurements for wireless sensor networks. In *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pages 694–701, Nov 2006.

[11] Sérgio Kostin, Ronaldo Moreira Salles, and Claudio Luis de Amorim. An approach for wireless sensor networks topology control in indoor scenarios. In *Proceedings of the 5th International Latin American Networking Conference*, LANC 09, pages 1–10, New York, NY, USA, 2009. ACM.

[12] Kenneth Bannister, Gianni Giorgetti, and Eep K. S. Gupta. Wireless sensor networking for ẗhot äpplications: Effects of temperature on signal strength, data collection and localization, 2015.

[13] John Thelen and Daan Goense. Radio wave propagation in potato fields. In *In 1st Workshop on Wireless Network Measurements*, 2005.

[14] Marco Zennaro, Hervé Ntareme, and Antoine Bagula. Experimental evaluation of temporal and energy characteristics of an outdoor sensor network. In *Proceedings of the*

International Conference on Mobile Technology, Applications, and Systems, Mobility '08, pages 99:1–99:5, New York, NY, USA, 2008. ACM.

[15] Qinghua Wang. Traffic analysis & modeling in wireless sensor networks and their applications on network optimization and anomaly detection. *Network Protocols & Algorithms*, 2(3):74–92, Sep 2010.

[16] J. Jeong, D. Culler, and J. H. Oh. Empirical analysis of transmission power control algorithms for wireless sensor networks. In *2007 Fourth International Conference on Networked Sensing Systems*, pages 27–34, June 2007.

[17] Dimitrios Lymberopoulos, Quentin Lindsey, and Andreas Savvides. An empirical characterization of radio signal strength variability in 3-d ieee 802.15.4 networks using monopole antennas. In Kay Römer, Holger Karl, and Friedemann Mattern, editors, *Wireless Sensor Networks*, pages 326–341, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[18] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[19] Hoang Anh Nguyen, A. Frster, D. Puccinelli, and S. Giordano. Sensor node lifetime: An experimental study. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 202–207, March 2011.

[20] A. Behzadan, A. Anpalagan, and B. Ma. Prolonging network lifetime via nodal energy balancing in heterogeneous wireless sensor networks. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5, June 2011.

[21] S. Okdem and D. Karaboga. Routing in wireless sensor networks using ant colony optimization. In *First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06)*, pages 401–404, June 2006.

[22] Tiago Camilo, Carlos Carreto, Jorge Sá Silva, and Fernando Boavida. *An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks*, pages 49–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[23] Anirban Sengupta, Reza Sedaghat, and Zhipeng Zeng. Rapid design space exploration by hybrid fuzzy search approach for optimal architecture determination of multi objective computing systems. *Microelectronics Reliability*, 51(2):502–512, 2011.

[24] Reza Sedaghat, Anirban Sengupta, and Pallabi Sarkar. A multi structure genetic algorithm for integrated design space exploration of scheduling and allocation in high level synthesis for DSP kernels. *Swarm and Evolutionary Computation*, 7:35–46, 2012.

[25] Anirban Sengupta and Reza Sedaghat. Swarm intelligence driven design space exploration of optimal k-cycle transient fault secured datapath during high level synthesis based on user power-delay budget. *Microelectronics Reliability*, 55(6):990–1004, 2015.

[26] Md. Akhtaruzzaman Adnan, Mohammd Abdur Razzaque, Ishtiaque Ahmed, and Ismail Fauzi Isnin. Bio-mimic optimization strategies in wireless sensor networks: A survey. *Sensors*, 14(1):299–345, 2014.

[27] Sohail Jabbar, Rabia Iram, Abid Ali Minhas, Imran Shafi, Shehzad Khalid, and Muqeet Ahmad. Intelligent optimization of wireless sensor networks through bio-inspired computing: Survey and future directions. *International Journal of Distributed Sensor Networks*, 9(2):421084, 2013.

[28] Jun Huang, Liqian Xu, Cong cong Xing, and Qiang Duan. A novel bioinspired multi-objective optimization algorithm for designing wireless sensor networks in the internet of things. *Journal of Sensors*, 192194:16, Mar 2015.

[29] C. A.C. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *Trans. Evol. Comp*, 8(3):256–279, June 2004.

[30] A. R. Rahimi-Vahed and S. M. Mirghorbani. A multi-objective particle swarm for a flow shop scheduling problem. *Journal of Combinatorial Optimization*, 13(1):79–102, 2007.

[31] Seyed Mahdi Jameii, Karim Faez, and Mehdi Dehghan. Multiobjective optimization for topology and coverage control in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 11(2):363815, 2015.

[32] O. De Weck. Multi-objective optimization: History and promise. In *The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems (Invited Keynote Paper, GL2-2) Kanazawa Japan*, volume 3, pages 1–4, Sep 2004.

[33] DS Ram, MC Bhuvaneswari, and Shanthi S Prabhu. A novel framework for applying multiobjective ga and pso based approaches for simultaneous area, delay, and power optimization in high level synthesis of datapaths. *VLSI design*, 2012:2, 2012.

[34] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51 – 67, 2016.

[35] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471, Nov 2007.

[36] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[37] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[38] P. Siddavaatam, R. Sedaghat, and A. T. Sharma. A novel multi-objective optimization approach for design flow in high level synthesis. volume 55, pages 990 – 1004, Jan 2018.

[39] Anirban Sengupta, Reza Sedaghat, and Pallabi Sarkar. Integrated design space exploration based on power-performance trade-off using genetic algorithm. In *International Conference on Advances in Computing and Artificial Intelligence, ACAI '11, Rajpura/Punjab, India - July 21 - 22, 2011*, pages 77–81, 2011.

[40] Anirban Sengupta, Reza Sedaghat, and Pallabi Sarkar. A multi structure genetic algorithm for integrated design space exploration of scheduling and allocation in high level synthesis for DSP kernels. *Swarm and Evolutionary Computation*, 7:35–46, 2012.

[41] Anirban Sengupta and Reza Sedaghat. Integrated scheduling, allocation and binding in high level synthesis using multi structure genetic algorithm based design space exploration. In *Proceedings of the 12th International Symposium on Quality Electronic Design, ISQED 2011, Santa Clara, California, USA, 14-16 March 2011*, pages 486–494, 2011.

[42] Anirban Sengupta and Deepak Kachave. Particle swarm optimisation driven low cost single event transient fault secured design during architectural synthesis. *The Journal of Engineering*, 1(1), 2017.

[43] Blackwell T. Poli R, Kennedy J. Particle swarm optimization an overview. *Swarm Intell.*, pages 1–25, Aug 2017.

[44] Hossam Faris, Ibrahim Aljarah, Mohammed Azmi Al-Betar, and Seyedali Mirjalili. Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications*, Nov 2017.

[45] Eric Post, Rolf O Peterson, Nils Chr Stenseth, and Brian E McLaren. Ecosystem consequences of wolf behavioural response to climate. *Nature*, 401(6756):905, 1999.

[46] M. J. Islam, M. S. R. Tanveer, and M. A. H. Akhand. A comparative study on prominent nature inspired algorithms for function optimization. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 803–808, May 2016.

[47] A. Ben Ammar, A. Dziri, M. Terre, and H. Youssef. Multi-hop leach based cross-layer design for large scale wireless sensor networks. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 763–768, Sept 2016.

[48] S. M. Abd El-atty. Efficient packet scheduling with pre-defined qos using cross-layer technique in wireless networks. In *11th IEEE Symposium on Computers and Communications (ISCC'06)*, pages 820–826, June 2006.

[49] P. Siddavaatam, R. Sedaghat, and A. T. Sharma. intel-leach: An optimal framework for node selection using dynamic clustering for wireless sensor networks. In *2017 12th IEEE International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 136–146, Dec 2017.

[50] I. F. Akyildiz and M. C. Vuran. Xlp: A cross-layer protocol for efficient communication in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9:1578–1591, 06 2010.

[51] D. Pompili and I. F. Akyildiz. A multimedia cross-layer protocol for underwater acoustic sensor networks. *IEEE Transactions on Wireless Communications*, 9(9):2924–2933, Sept 2010.

[52] I. F. Akyildiz and X. Wang. Cross-layer design in wireless mesh networks. *IEEE Transactions on Vehicular Technology*, 57(2):1061–1076, Mar 2008.

[53] M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross-layering in mobile ad hoc network design. *Computer*, 37(2):48–51, Feb 2004.

[54] IETF Routing Over Low power and Lossy networks (ROLL). Ietf routing over low power and lossy networks (roll). `https://datatracker.ietf.org/doc/charter-ietf-roll/`, January 2016.

[55] Novella Buonaccorsi, Claudio Cicconetti, Raffaella Mambrini, Nick Podias, and Paul Russell. Etsi m2m release 1 demonstration. *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, 2012.

[56] IEEE 802.15. Ieee 802.15 wireless personal area networks task group 4. `http://www.ieee802.org/15/pub/TG4.htm`, January 2016.

[57] Sunil Kumar, Vineet S. Raghavan, and Jing Deng. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Netw.*, 4(3):326–358, May 2006.

[58] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919 (Informational), August 2007.

[59] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349, 2005.

[60] M. C. Vuran and I. F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 14(2):316–329, April 2006.

[61] Eugene Shih, Seong-Hwan Cho, Nathan Ickes, Rex Min, Amit Sinha, Alice Wang, and Anantha Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 272–287, New York, NY, USA, 2001.

[62] Wenyu Jiang and Henning Schulzrinne. Modeling of packet loss and delay and their effect on real-time multimedia service quality. In *PROCEEDINGS OF NOSSDAV '2000*, 2000.

[63] Parametric survey on cross-layer designs for wireless sensor networks. *Computer Science Review*, 27:112 – 134, 2018.

[64] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

[65] R. Almesaeed, A. S. Ameen, A. Doufexi, N. Dahnoun, and A. R. Nix. A comparison study of 2d and 3d itu channel model. In *2013 IFIP Wireless Days (WD)*, pages 1–7, Nov 2013.

[66] R. Sassioui, M. Jabi, L. Szczecinski, L. B. Le, M. Benjillali, and B. Pelletier. Harq and amc: Friends or foes? *IEEE Transactions on Communications*, 65(2):635–650, Feb 2017.

[67] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, Dec 2004.

[68] L. Le Cam. The central limit theorem around 1935. *Statist. Sci.*, 1(1):78–91, 02 1986.

[69] Peter J. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 221–233, Berkeley, Calif., 1967. University of California Press.

[70] F. Fu and M. van der Schaar. A new systematic framework for autonomous cross-layer optimization. *IEEE Transactions on Vehicular Technology*, 58(4):1887–1903, May 2009.

[71] Y. Dong and C. H. Chang. An improved autonomous cross-layer optimization framework for wireless multimedia communication. In *2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)*, pages 53–58, June 2014.

[72] N. Singh and S.B. Singh. A novel hybrid gwo-sca approach for optimization problems. *Engineering Science and Technology, an International Journal*, 20(6):1586 – 1601, 2017.

[73] Seyedali Mirjalili. Sca: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96:120 – 133, 2016.

[74] Philippe Coussy and Adam Morawiec. *High-level synthesis: from algorithm to digital circuit*. Springer Science & Business Media, 2008.

[75] I. XILINX. High-level synthesis design suite guide(vivado). `https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug902-vivado-high-level-synthesis.pdf`, 2017.

[76] Anirban Sengupta and Deepak Kachave. Low cost fault tolerance against kc-cycle and km-unit transient for loop based control data flow graphs during physically aware high level synthesis. *Microelectronics Reliability*, 74:88–99, 2017.

[77] Keith Campbell, Wei Zuo, and Deming Chen. New advances of high-level synthesis for efficient and reliable hardware design. *Integration, the VLSI Journal*, 58:189–214, 2017.

[78] Hongwei Wang, Ziyuan Zhu, Jinglin Shi, and Yongtao Su. An accurate acosso metamodeling technique for processor architecture design space exploration. In *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pages 689–694. IEEE, 2015.

[79] Eunsuk Kang, Ethan Jackson, and Wolfram Schulte. An approach for effective design space exploration. In *Monterey Workshop*, pages 33–54. Springer, 2010.

[80] Virginie Fresse, Catherine Combes, Matthieu Payet, and Frédéric Rousseau. Methodological framework for noc resources dimensioning on fpgas. *Procedia Computer Science*, 56:391–396, 2015.

[81] Bernhard Schätz, Sebastian Voss, and Sergey Zverlov. Automating design-space exploration: optimal deployment of automotive sw-components in an iso26262 context. In *Proceedings of the 52nd Annual Design Automation Conference*, page 99. ACM, 2015.

[82] Bruno Bodin, Luigi Nardi, M Zeeshan Zia, Harry Wagstaff, Govind Sreekar Shenoy, Murali Emani, John Mawer, Christos Kotselidis, Andy Nisbet, Mikel Lujan, et al. Integrating algorithmic parameters into benchmarking and design space exploration in 3d scene understanding. In *Proceedings of the 2016 International Conference on Parallel Architectures and Compilation*, pages 57–69. ACM, 2016.

[83] AC Williams, AD Brown, and M Zwolinski. Simultaneous optimisation of dynamic power, area and delay in behavioural synthesis. *IEE Proceedings-Computers and Digital Techniques*, 147(6):383–390, 2000.

[84] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[85] Jan-Philip Schmidt Hung Nguyen-Schfer. *Tensor Analysis and Elementary Differential Geometry for Physicists and Engineers*. Springer-Verlag Berlin Heidelberg, Germany, 2017.

[86] K.F. Riley, M.P. Hobson, and S.J. Bence. Mathematical methods for physics and engineering: A comprehensive guide. chapter 5. Cambridge University Press, 2016.

[87] Saumya Bhadauria and Anirban Sengupta. Adaptive bacterial foraging driven datapath optimization: Exploring power-performance tradeoff in high level synthesis. *Applied Mathematics and Computation*, 269:265 – 278, 2015.

[88] David Ellerman. Series-parallel duality in the mathematics of appraisal, 2008.

[89] F. Melese, A. Richter, and B. Solomon. Military cost–benefit analysis: Theory and practice. Routledge Studies in Defence and Peace Economics, chapter 17. Taylor & Francis, 2015.

[90] David C Ku and Giovanni DeMicheli. *High level synthesis of ASICs under timing and synchronization constraints*, volume 177. Springer Science & Business Media, 2013.

[91] de Vegt Rolf, Cherian George, Barriac Gwen, and Tian Qingjiang. A quantification of 5 ghz unlicensed band spectrum needs. White paper, Qualcomm Technologies, Inc, JUL 2016.

[92] Charles Darwin and Mortimer J Adler. *On the origin of species by means of natural selection*. Broadview Press Peterborough, Ontario, Canada, 2003.

[93] Christophe Bobda. *Introduction to reconfigurable computing: architectures, algorithms, and applications*. Springer Science & Business Media, 2007.

[94] Giovanni De Micheli. *Synthesis and optimization of digital circuits*. McGraw-Hill Higher Education, 1994.

[95] Incorporated XILINX. *ISE Simulator, Version 2017.3*. Vivado Design Suite.

[96] Design Patterns RHRJJV Erich Gamma. Elements of reusable object-oriented software, 1994.

[97] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[98] Summit Sehgal. Performance driven design space exploration for multi-objective optimization using stability in competition, Aug 2011.

[99] M. Valdez-Vivas and N. Bambos. Latency and energy in quality-driven applications for networked wireless devices. In *2013 IEEE International Conference on Communications (ICC)*, pages 6019–6024, June 2013.

[100] V. Venugopal and D. M. Shila. High throughput implementations of cryptography algorithms on gpu and fpga. In *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 723–727, May 2013.

[101] Charalampos Manifavas, George Hatzivasilis, Konstantinos Fysarakis, and Konstantinos Rantos. *Lightweight Cryptography for Embedded Systems – A Comparative Analysis*, pages 333–349. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[102] Nicolas T. Courtois. *Information Security and Cryptology — ICISC 2002: 5th International Conference Seoul, Korea, November 28–29, 2002 Revised Papers*, chapter Higher Order Correlation Attacks, XL Algorithm and Cryptanalysis of Toyocrypt, pages 182–199. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[103] Nicolas T. Courtois and Jacques Patarin. *Topics in Cryptology — CT-RSA 2003: The Cryptographers' Track at the RSA Conference 2003 San Francisco, CA, USA, April 13–17, 2003 Proceedings*, chapter About the XL Algorithm over GF(2), pages 141–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[104] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. *Advances in Cryptology — EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques Bruges, Belgium, May 14–18, 2000 Proceedings*, chapter Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations, pages 392–407. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[105] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[106] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, EUROCRYPT '93, pages 386–397, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[107] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials :two new families of asymmetric algorithms. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, pages 33–48, 1996.

[108] Jacques Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt 98. *Des. Codes Cryptography*, 20(2):175–209, 2000.

[109] Nicolas T. Courtois. Algebraic attacks on combiners with memory and several outputs. In *Proceedings of the 7th International Conference on Information Security and Cryptology*, ICISC'04, pages 3–20, Berlin, Heidelberg, 2005. Springer-Verlag.

[110] Nicolas T. Courtois and Josef Pieprzyk. *Advances in Cryptology — ASIACRYPT 2002: 8th International Conference on the Theory and Application of Cryptology and Information Security Queenstown, New Zealand, December 1–5, 2002 Proceedings*, chapter Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[111] Itai Dinur and Adi Shamir. Applying cube attacks to stream ciphers in realistic scenarios. *Cryptography Commun.*, 4(3-4):217–232, dec 2012.

[112] Itai Dinur and Adi Shamir. *Advances in Cryptology - EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, chapter Cube Attacks on Tweakable Black Box Polynomials. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[113] Itai Dinur and Adi Shamir. Breaking grain–128 with dynamic cube attacks. In *Proceedings of the 18th International Conference on Fast Software Encryption*, FSE'11, pages 167–187, Berlin, Heidelberg, 2011. Springer-Verlag.

[114] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. *Fast Software Encryption: 16th International Workshop, FSE 2009 Leuven, Belgium, February 22-25, 2009 Revised Selected Papers*, chapter Cube Testers and Key Recovery Attacks on Reduced-Round MD6 and Trivium. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[115] Alejandro Hevia and Marcos Kiwi. Strength of two data encryption standard implementations under timing attacks. *ACM Trans. Inf. Syst. Secur.*, 2(4):416–437, November 1999.

[116] Leo Dorrendorf, Zvi Gutterman, and Benny Pinkas. Cryptanalysis of the random number generator of the windows operating system. *ACM Trans. Inf. Syst. Secur.*, 13(1):10:1–10:32, November 2009.

[117] Scott A. Crosby, Dan S. Wallach, and Rudolf H. Riedi. Opportunities and limits of remote timing attacks. *ACM Trans. Inf. Syst. Secur.*, 12(3):17:1–17:29, January 2009.

[118] Jonathan J. Hoch and Adi Shamir. *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, chapter Fault Analysis of Stream Ciphers, pages 240–253. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[119] Patrik Ekdahl and Thomas Johansson. A new version of the stream cipher snow. In Kaisa Nyberg and Howard Heys, editors, *Selected Areas in Cryptography*, pages 47–61, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[120] 3rd Generation Partnership Project. Specification of the 3GPP Confidentiality and Integrity Algorithms - Document 2: KASUMI Specification (Release 6). Technical Report 3GPP TS 35.202 V6.1.0 (2005-09), sep 2005.

[121] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, EUROCRYPT '90, pages 389–404, New York, NY, USA, 1991. Springer-Verlag New York, Inc.

[122] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'91, pages 17–38, Berlin, Heidelberg, 1991. Springer-Verlag.

[123] Sandy Harris and Carlisle M. Adams. Key-dependent s-box manipulations. In *Proceedings of the Selected Areas in Cryptography*, SAC '98, pages 15–26, London, UK, 1999. Springer-Verlag.

[124] Carolynn Burwick, Don Coppersmith, Edward D'Avignon, Rosario Gennaro, Shai Halevi, Charanjit Jutla, Stephen M. Matyas Jr, Luke O'Connor, Mohammad Peyravian, Jr. Luke, O'connor Mohammad Peyravian, David Stafford, and Nevenko Zunic. Mars - a candidate cipher for aes. *NIST AES Proposal*, 1999.

[125] Nicolas T. Courtois and Blandine Debraize. Algebraic description and simultaneous linear approximations of addition in snow 2.0. In *Proceedings of the 10th International Conference on Information and Communications Security*, ICICS '08, pages 328–344, Berlin, Heidelberg, 2008. Springer-Verlag.

[126] William E. Cobb, Rusty O. Baldwin, and Eric D. Laspe. Leakage mapping: A systematic methodology for assessing the side-channel information leakage of cryptographic implementations. *ACM Trans. Inf. Syst. Secur.*, 16(1):2:1–2:29, June 2013.

[127] Prathap Siddavaatam, Reza Sedaghat, and Min Hsuan Cheng. An adaptive security framework with extensible computational complexity for cipher systems. In *2016 International Conference for Internet Technology and Secured Transactions*, pages 27–31, Dec 2016.

[128] Min Hsuan Cheng, Reza Sedaghat, and Prathap Siddavaatam. A new adaptable construction of modulo addition with scalable security for stream ciphers. In *Network and System Security - 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings*, pages 383–397, 2016.

[129] Prathap Siddavaatam and Reza Sedaghat. A novel architecture with scalable security having expandable computational complexity for stream ciphers. *Facta Universitatis, Series: Electronics and Energetics*, 30(1):459–475, Sept 2017.

[130] Simon Fischer and Willi Meier. *Fast Software Encryption: 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, chapter Algebraic Immunity of S-Boxes and Augmented Functions, pages 366–381. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[131] Frederik Armknecht and Willi Meier. *Fault Attacks on Combiners with Memory*, pages 36–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[132] Patrik Ekdahl and Thomas Johansson. A new version of the stream cipher snow. In *Selected Areas in Cryptography, 9th Annual International Workshop*, SAC '02, pages 47–61, London, UK, UK, 2003. Springer-Verlag.

[133] Olivier Billet and Henri Gilbert. *Topics in Cryptology – CT-RSA 2005: The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005. Proceedings*, chapter Resistance of SNOW 2.0 Against Algebraic Attacks, pages 19–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

Glossary

5G 5th Generation Wireless Systems. 5, 9

ABC Artificial Bee Colony. 30

ABF Adaptive Bacterial Foraging. viii, xi, 100, 103, 104, 108, 109

AHN Ad-Hoc Networks. 34, 41

ARQ Adaptive Repeat Request. 53

ASAP As Soon As Possible. xiii, 99

ASIC Application Specific Integrated Circuit. 69, 109

BPSK Binary Phase Shift Keying. 48

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 53

DE Differential Evolution. 30

DFG Data Flow Graph. xiii, 102

DSE Design Space Exploration. 6, 8–10, 67–70, 82, 84, 91, 93, 97, 98, 103, 104

DSP Digital Signals Processing. 101

DWT Discrete Wavelet Transform. 103

FEC Forward Error Correction. 57

FPGA Field Programmable Gate Array. 100

GA Genetic Algorithm. 21, 22, 30, 70, 104, 105

GAP Gateway Access Points. 41–43, 53

GE Gate Equivalents. 9

GWO Grey Wolf Optimizer. vii, 7, 22, 26, 27, 29–31, 33, 36, 37, 48, 57, 64, 67, 70, 82, 104, 106

GWO-DSE Grey Wolf Optimizer Driven DSE. viii, xi, 31, 67, 82–84, 92, 94, 98–101, 103, 104, 108, 109

HLS High-Level Synthesis. 10, 68, 87, 109

IoE Internet of Everything. 5

IoT Internet of Things. vii, ix, xii, 1–11, 13–15, 18, 20–22, 30–34, 36, 39–47, 49, 52–57, 60, 62, 65, 66, 70, 103, 109–111, 114, 115, 138–140

ISM Industrial, Scientific and Medical. 3

ITU-T International Telecommunication Union. 2

KP Knapsack Problem. 20, 21

MAC Medium Access Control. 4, 39, 40, 53

MKP Multidimensional Knapsack Problem. 21

MoA Multi-objective Algorithms. 6, 15, 21

MOO Multi-Objective Optimization. 6, 10, 13, 15, 21

MoP Multi-objective Optimization Problem. 17, 18

PF Priority Factor. xii, 17, 18

PSO Particle Swarm Optimization. 18, 22–24, 30, 70, 103, 104, 106, 109

QoS Quality-of-Service. 32, 34, 36, 39, 44, 53, 59

QPSK Quadrature Phase Shift Keying. 48

RFID Radio Frequency Identification. 5, 111

RTL Register Transfer Level. 68, 101

S-box Substitution-Box. 117

SMAC Sleep MAC. 52, 53

SNR Signal to Noise Ratio. 48, 53

UC Utility Coefficient. 84, 91

UR Utility Rank. xiii, 8, 84, 91, 94, 101, 102

VLSI Very-Large-Scale Integration. 69, 84

WMN Wireless Mesh Networks. 34, 41

WSN Wireless Sensor Network. 3, 5, 6, 15, 34, 41, 111

WSPSO Weighted Sum Particle Swarm Optimization. 70