Volume 6, Issue 2, Aug. 01, 2021

TC-CPS Newsletter

IEEE
TC-CPS
Cyber-Physical Systems

An Overview of TSV Fault-Tolerance Design in 3D IC

Qi Xu, University of Science and Technology of China

Abstract

In three dimensional integrated circuits (3D-ICs), through silicon via (TSV) is a critical technique in providing vertical connections. However, the yield is one of the main obstacles to the adoption of the TSV-based 3D-ICs technology in industry. Various TSV fault-tolerance designs using redundant TSVs have been proposed in literature to improve yield and reliability. In this paper, we review some recent TSV fault-tolerance approaches. We hope to inspire more work and to see more talented methods in this field.

1 Introduction

As device feature sizes continue to rapidly decrease, interconnection delay is becoming a bottleneck limiting IC performance. Three dimensional integrated circuits (3D-ICs) technology involves the vertical stacking of multiple dies connected by through silicon vias (TSVs), providing a promising way to alleviate the interconnect problem and achieve a significant reduction in chip area, wire-length and interconnect power [1]. Research shows that the average wire-length of 3D-ICs varies with the square root of the number of layers [2]. In addition, 3D-ICs also offer the potential for heterogeneous integration, which is essential for More than Moore (MtM) technology. Figure 1 illustrates an example of a 3D-IC, where CPUs, memories, analog circuits and sensors are stacked together. Although 3D integration has already appeared in commercial applications in the form of 3D memory, there are still significant open problems in both research and implementation [3].

One fundamental problem in 3D-ICs is the TSV yield loss. General speaking, there are two types of yield losses in 3D-ICs, which are caused by defects in stacked dies or defects introduced during the assembly process [4]. In the former case, pre-bonding testing is essential to avoid the stacking of defective dies [5]. Several inter-die

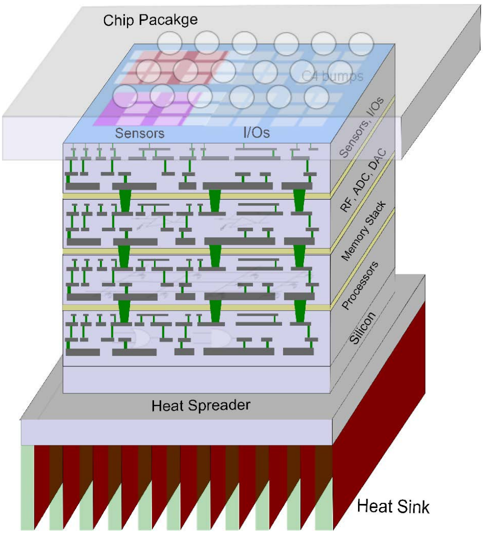


Figure 1: 3D-IC provides a solution for heterogeneous integration [3].

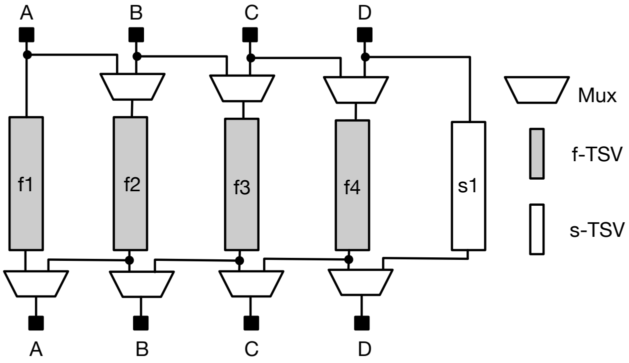


Figure 2: A regular chain structure.

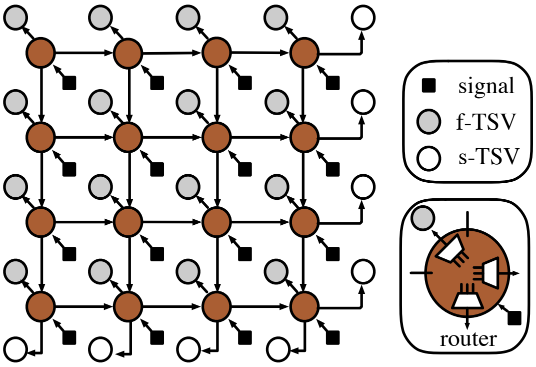


Figure 3: Router-based TSV redundancy structure.

repairing and die/wafer matching methods have also been proposed to enhance the stacking yield [6]. In the latter case, adding spare TSVs (denoted as **s-TSVs**) to repair faulty functional TSVs (denoted as **f-TSVs**) is an effective strategy to increase yield and ensure reliability. By adding the multiplexers (i.e., Muxes) and carefully designing the reconfigurable TSV replacing paths, a TSV fault-tolerance structure can be constructed, where the s-TSVs are used to transfer signals in the presence of faulty f-TSVs.

2 Related TSV Fault-Tolerance Works

In this section, we describe several TSV fault-tolerance structures, and discuss their pros and cons.

Hsieh *et al.* [7] proposed a regular TSV replacing chain structure, as shown in Figure 2. Each f-TSV is regularly connected to the right-hand side neighboring TSV, and the rightmost f-TSV is connected to an s-TSV. However, since only one s-TSV is inserted in each TSV group, the chain fault-tolerance structure cannot be repaired in case of more than one faulty TSVs. Similarly, Wang *et al.* [8] presented a redundant TSV allocation technique for reducing the yield loss. A greedy method is used to partition f-TSVs into groups and then an integer linear programming (ILP) formulation is adopted to allocate s-TSVs for each group with minimization of delay overhead. But the generation of fault-tolerance structure is not considered since they assume the regular chain structures always exist.

Jiang *et al.* [9] proposed a router-based TSV redundancy architecture to repair clustered TSV faults. As shown in Figure 3, the f-TSVs are regularly distributed in a uniform 4×4 grid structure, and the s-TSVs are placed on the right and bottom boundaries of the structure. Thus, the signals are re-routed from two directions (from left to right or from top to bottom). Besides, each f-TSV is connected to a router, which contains six ports and three 3-to-1 multiplexers. The signal and its corresponding f-TSV occupy two ports in the router, while the remaining four ports are linked to other routers in four different directions. Therefore, the signal port and two linking ports (left and

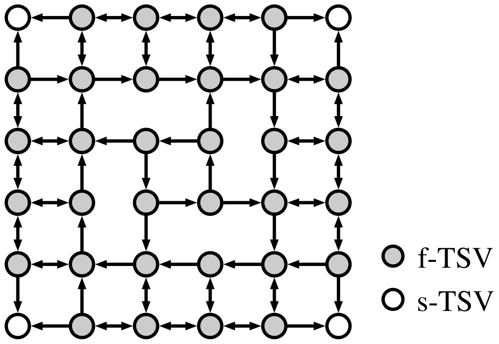


Figure 4: Ring-based TSV redundancy architecture.

top) can connect to the TSV port and the remaining linking ports (right and bottom) through the multiplexers. To minimize the delay overhead due to signal re-routing, a heuristic search algorithm is developed to generate replacing paths for each faulty TSV. As a result, the signals related to faulty TSVs can be re-routed to fault-free TSVs that are distant rather than to the neighboring TSVs. Thus, the router-based redundancy structure achieves a high TSV yield for the clustered fault. But the structure requires too many r-TSVs, and each TSV is equipped with three multiplexers, resulting in a high hardware overhead.

Lo *et al.* [10] presented a ring-based TSV redundancy structure to repair faulty TSVs. As shown in Figure 4, the TSVs are distributed in a uniform grid structure, which are divided into multiple rings, and the s-TSVs are placed in four corners of the TSV grid or anywhere of the outermost ring. The signals can be shifted in the direction of their own ring and the outer ring through different types of multiplexers. When there exists a faulty f-TSV, the corresponding signal will be transferred to its neighboring f-TSV until an s-TSV is used. Although the ring-based architecture requires fewer s-TSVs, the structure is significantly affected by the clustered TSV faults.

Furthermore, Xu *et al.* [11] presented a switch-based TSV fault-tolerance structure during floorplanning, as illustrated in Figure 5. Based on the replaceable relations between f-TSVs, an ILP-based model is developed to form a fault-tolerance structure, with minimization of multiplexer delay overhead and hardware cost. However, the work [11] is under an assumption that a predetermined number of common s-TSVs is assigned to each TSV group, which causes overuse of s-TSVs. To overcome the issue, Chen *et al.* [12] develop an adaptive switch-based TSV fault-tolerance structure, in which the number of tolerant faults is adaptively determined by the distribution of the f-TSVs and their candidate s-TSVs. As a result, the number of s-TSVs is effectively reduced. Besides, Maity *et al.* [13] presented a tree-based TSV redundancy structure. In fact, the tree-based structure is still essentially a switch-based approach.

Lee *et al.* [14] developed a group-based TSV redundancy architecture, where all TSVs are partitioned into several groups. As illustrated in Figure 6, 12 f-TSVs in a TSV block are divided into four groups, and an s-TSV is assigned to each group. Each f-TSV node comprises a signal, a TSV, and a 2-to-1 multiplexer, while each s-TSV node contains a TSV and a multiplexer. Note that the type of the multiplexer connected with s-TSV depends on the group numbers and the number of f-TSVs in each group. For example in Figure 6, the inputs of the multiplexer of s-TSV s_2 include all signals of the B group and one signal of the other three groups. Thus, a 6-to-1 multiplexer is allocated to each s-TSV to re-route the signals. Due to the use of large multiplexers, the hardware cost of the group-based structure is very high.

Moreover, Wang *et al.* [15] proposed a cellular TSV fault-tolerance structure, as shown in Figure 7. To ensure the corresponding signal can be switched in three directions, each f-TSV node comprises a signal, a TSV, a 4-to-1 multiplexer and a 1-to-4 demultiplexer. The multiplexer selects the signals from the current node or three adjacent nodes, while the demultiplexer transfers the signal to its corresponding TSV or three neighboring TSVs. Besides, a min-cost max-flow based algorithm is presented to generate the TSV repair paths. However, each signal in the cellular structure can only be rerouted to TSVs within one hop. As a result, the structure is vulnerable to any faulty TSVs in close proximity, resulting in a low TSV yield under the clustered TSV fault distribution. As illustrated

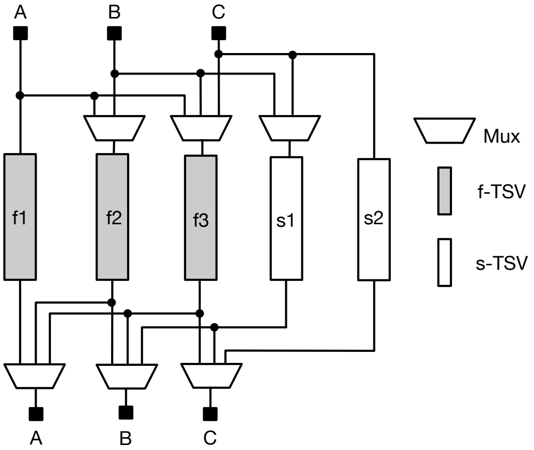


Figure 5: Switch-based TSV redundancy architecture.

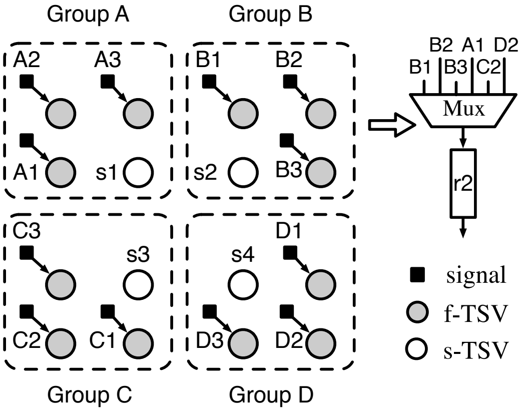


Figure 6: Group-based TSV redundancy structure.

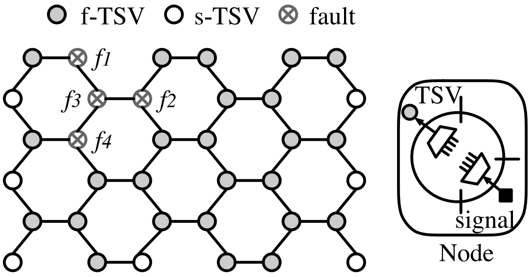


Figure 7: Cellular TSV redundancy architecture.

in Figure 7, since the three adjacent nodes connected with f-TSV f_3 are faulty, the repair path for f_3 cannot be generated.

To handle clustered TSV faults, a novel cellular TSV redundancy architecture is proposed in [16], with taking account of the delay overhead. As shown in Figure 8, each TSV is connected to a router, which contains five ports,

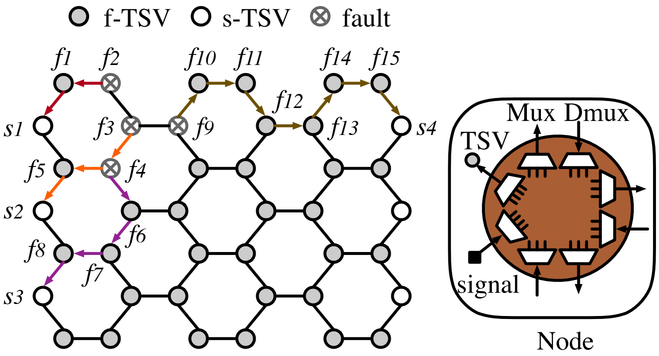


Figure 8: Example of the proposed cellular fault-tolerance structure, where the vertex-disjoint paths for the faulty f-TSVs are as follows: f_2: $\{f_2 \rightarrow f_1 \rightarrow r_1\}$ (red lines), f_3: $\{f_3 \rightarrow f_5 \rightarrow r_2\}$ (yellow lines), f_4: $\{f_4 \rightarrow f_6 \rightarrow f_7 \rightarrow f_8 \rightarrow r_3\}$ (purple lines), f_{10}: $\{f_{10} \rightarrow f_{11} \rightarrow f_{12} \rightarrow f_{13} \rightarrow f_{14} \rightarrow f_{15} \rightarrow f_{16} \rightarrow r_4\}$ (brown lines), while other fault-free f-TSVs transfer their corresponding signals.

one 4-to-1 Mux, three 3-to-1 Muxes, one 1-to-4 Dmux, and three 1-to-3 Dmuxes. The signal and its corresponding TSV occupy two ports in the router, while the remaining three ports are linked to other TSV routers in three different directions. Since a signal can be transferred to its corresponding TSV or three adjacent nodes, a 1-to-4 Dmux is allocated to the signal port. Similarly, a TSV may select the signal on the current node or the signals from the three adjacent nodes, the TSV port requires a 4-to-1 Mux. When all f-TSVs are fault-free, the Mux directly selects the signal on the current node for transmission. Once the current node lies on a replacing path for a faulty TSV, the Mux chooses the signal from one of the three adjacent nodes. In addition, when a linking port receives a signal from the adjacent node, the signal will output to the TSV port or the other two linking ports. Thus, a 1-to-3 Dmux is allocated to each linking port. In addition, each linking port also needs a 3-to-1 Mux to select a signal from the current node or from the other two linking ports. Because the signals related to faulty TSVs can be transferred to fault-free TSVs that are distant, the cellular TSV redundancy architecture can provide high TSV yield.

Recently, Cheong *et al.* [17] proposed a 3D rotation-based TSV architecture by mimicking a Rubik's cube. But due to the large number of used multiplexers, the hardware cost of the 3D rotation-based structure is high. Park *et al.* [18] proposed a herringbone-based TSV repair architecture to simultaneously address both manufacturing TSV faults and aging-related problems. However, the TSV aging model is not accurate. Besides, a thermal-aware TSV recovery methodology is presented by Dang *et al.* [19] to deal with the clustered TSV faults. But the recovery architecture is not suitable to the irregular TSV placement.

3 Conclusion

In this paper, we summarize some state-of-the-art TSV fault-tolerant designs. We see that with the effective fault-tolerance structure and repair algorithm, the TSV yield can be enhanced. As continuing growth of technology node, 3D IC turns out to be a promising solution to further scaling, we believe this paper will stimulate more research on yield aware 3D IC design.

References

[1] Souri, Shukri J et al. Multiple Si layer ICs: Motivation, performance analysis, and design implications. In *Proc. of DAC*, pp. 213–220, 2000.

[2] Joyner, James W et al. A global interconnect design window for a three-dimensional system-on-a-chip. In *Proc. of IITC*, pp. 154–156, 2001.

[3] Lu, Tiantao et al. TSV-Based 3-D ICs: Design Methods and Tools. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 36, no. 10, pp. 1593–1619, 2017.

[4] Xu, Qiang et al. Yield enhancement for 3D-stacked ICs: Recent advances and challenges. In *Proc. of ASPDAC*, pp. 731–737, 2012.

[5] Lee, Hsien-Hsin S et al. Test challenges for 3D integrated circuits. IEEE Design & Test of Computers, vol. 26, no. 5, pp. 26–35, 2009.

[6] Ferri, Cesare et al. Strategies for improving the parametric yield and profits of 3D ICs. In *Proc. of ICCAD*, pp. 220–226, 2007.

[7] Hsieh, Ang-Chih et al. TSV redundancy: architecture and design issues in 3-D IC. IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 20, no. 4, pp. 711–722, 2012.

[8] Wang, Shengcheng et al. Defect Clustering-Aware Spare-TSV Allocation for 3D ICs. In *Proc. of ICCAD*, pp. 307–314, 2015.

[9] Jiang, Li et al. On effective through-silicon via repair for 3-D Stacked ICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 32, no. 4, pp. 559–571, 2013.

[10] Lo, Wei-Hen et al. Architecture of Ring-Based Redundant TSV for Clustered Faults. IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 24, no. 12, pp. 3437–3449, 2016.

[11] Xu, Qi et al. Clustered Fault Tolerance TSV Planning for 3D Integrated Circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 36, no. 8, pp. 1287–1300, 2017.

[12] Chen, Song et al. Adaptive 3D-IC TSV Fault Tolerance Structure Generation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 38, no. 5, pp. 949–960, 2019.

[13] Maity, Dilip Kumar et al. TSV-Cluster Defect Tolerance Using Tree-based Redundancy for Yield Improvement of 3D ICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), early access, 2020.

[14] Lee, Ingeol et al. Highly Reliable Redundant TSV Architecture for Clustered Faults. IEEE Transactions on Reliability, vol. 68, no. 1, pp. 237–247, 2019.

[15] Wang, Qin et al. A New Cellular-Based Redundant TSV Structure for Clustered Faults. IEEE Transactions on Very Large Scale Integration Systems (TVLSI), vol. 27, no. 2, pp. 458–467, 2019.

[16] Xu, Qi et al. Cellular Structure Based Fault-Tolerance TSV Configuration in 3D-IC. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), early access, 2021.

[17] Cheong, Minho et al. A 3-D Rotation-Based Through-Silicon via Redundancy Architecture for Clustering Faults. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 39, no. 9, pp. 1925–1934, 2020.

[18] Park, Sangmin et al. Herringbone Based TSV Architecture for Clustered Fault Repair and Aging Recovery. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), early access, 2021.

[19] Dang, Khanh N et al. HotCluster: A thermal-aware defect recovery method for Through-Silicon-Vias Towards Reliable 3-D ICs systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), early access, 2021.

Generating Random Keys for Cyber Physical System from Asynchronous Chaotic Topology

Yukui Luo, Shijin Duan, Xiaolin Xu
Northeastern University

Abstract

While the Cyber-Physical Systems (CPS) are emerging in the connection of networks and physical systems, the security concerns of CPS are drawing attentions. As a critical security primitive, true random number generator (TRNG) is utilized to generate one-time secret keys in CPS, which is the root of trust for the privacy of CPS. In this work, we propose a novel digital TRNG design based on a self-timed ring structure. The realization of this proposed TRNG is composed of an asynchronous chaotic cellular automata topology, which is purely digital with the ease of synthesis on standard all-digital components. We evaluate the performance of the proposed TRNG on three implementations, including HSpice simulation, ASIC test chip, and FPGA prototype. Further, the collected random numbers from test chips and FPGA are examined with three typical test suites, including NIST SP800–22, NIST SP800–90B, and AIS-31. The experimental result shows that the proposed TRNG can successfully pass all the test suites, achieving high throughput while only consuming much smaller hardware resources and energy, compared to other state-of-the-art TRNG designs. Moreover, the security validation demonstrates that the proposed TRNG is immune to frequency injection attacks, power attacks, and thermal attacks.

1 Introduction

Cyber-Physical Systems (CPS) applications are widely deployed in modern society, ranging from smart daily items to controllable industrial producing [1]. Due to the critical role of CPS, attacks on vital CPS implementation could cause immeasurable disasters. For example, the nation-wide smart power grid, a typical application on CPS, could suffer from significant economic loss and latent damage in many aspects if under attack [2]. Unlike software/firmware, the hardware in CPS cannot update patches with flexible configuration and negligible labor cost, making the hardware security in CPS more challenging and essential. As the root of trust for many cryptographic applications on hardware, secret key is utilized to certificate security-related operations in CPS [3].

For secret key generation, true random number generator (TRNG) is a promising solution in CPS secure applications [3]. TRNG is proposed to generate random numbers from physical characteristics that are non-deterministic and predictable in advance. Nevertheless, the embedding of TRNG is complicated because extra circuits are needed to transfer physical characteristic to digital output, such as the converters connecting the analog circuit-based TRNG and digital systems [4]. Further, most nonsymmetric TRNG designs must consider the bias of random numbers to logic 1 or 0 to ensure the randomness [5]. Several works have discussed the criteria for ideal TRNG designs, e.g., in [3], it is suggested that the TRNG structure is supposed to have high throughput, unbiased randomness, and minimal latency of cryptographic hash functions, for suiting numerous high-speed applications. Also, the consideration of the practical hardware implementation complexity for TRNG structure is addressed in [6].

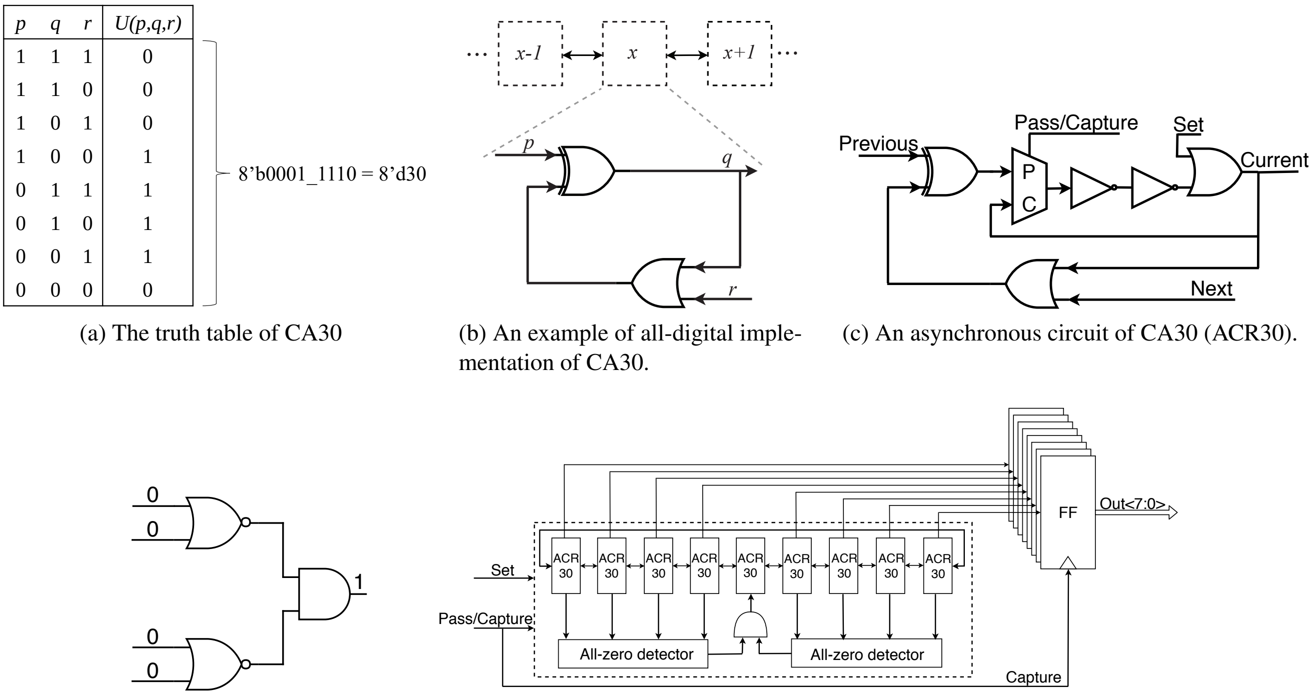
To eliminate the inconvenience and bias of transfer from the measurement of analog signals to digital systems, also easing the hardware implementation complexity, we present a high-performance and secure TRNG design composed of all-digital components. The proposed TRNG utilize the chaotic property of cellular automata (CA) topology [7] as the source of randomness, which is pure digital but still undeterministic. Based on digital CA30 component, a self-timed ring structure is formulated to generate random numbers. To explore the performance of the proposed TRNG, we evaluate it on HSpice simulation, ASIC test chips, and FPGA implementations. The results show the high-throughput of this proposed TRNG. Further, to validate the randomness and functionality, this TRNG passes all test statistics of three prevalent test suites. As for the security evaluation, we test the resistance of the proposed TRNG against three common attacks threatening most existing TRNGs: frequency injection attack, thermal attack and power attack. By evaluating the entropy model of the proposed TRNG, we show that it is immune to all these attacks with trivial entropy loss.

2 CA30 based TRNG Design

Compare with other self-timed based TRNG [8, 9, 10] and PRNG [11, 12] circuits, our proposed CA30 based self-timed ring structure can achieve a smaller area overhead and high-energy efficiency in random numbers generation. Moreover, we introduce the random entropy source while implementing the CA topology, which advances our scheme's robustness from statistic attacks.

2.1 Cellular Automata Principle and Self-time Ring Design

A CA can be defined as a deterministic system. It is constructed by a group of identical CA elements, where each element has k possible states. The 1-dimension CA structure is the simplest case, which consists of a series of CA cells. Each cell x has two possible states: 0 or 1 (i.e., k=2), and the current state is determined by its neighborhoods and a certainty update rule U. If defining the neighborhood range of x to be 1, in 1-dimensional space, the neighborhoods are $x-1$ and $x+1$. Therefore, we can use three parameters (p,q,r) to denote these three CA cells ($x-1$, x, $x+1$). The new state of x can be formulated with the state update rule $U(p,q,r)$, where the inputs are *(p, q, r)* and the rule is CA30. In addition, the paper [13] proves that the state evolution of the CA30 scheme is chaotic.



p	*q*	*r*	*U(p,q,r)*
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

(a) The truth table of CA30

(b) An example of all-digital implementation of CA30.

(c) An asynchronous circuit of CA30 (ACR30).

(d) The schematic of all-zero detector.

(e) The schematic of proposed all-digital TRNG design.

Figure 1: The schematics of the proposed TRNG [14]. (a) The truth table of the CA30 rule. (b) A feasible CA30 implementation circuit. (c) Using an asynchronous circuit to realize the CA30 (ACR30), which is the basic element in our proposed TRNG.(d) All-zero "stable" state detector. (e)Proposed all-digital TRNG, which is constructed by nine ACR30s. The middle one stands for breaking the "stable" state of the entire system and the rest eight ACR30s are used to generate true random number sequences.

In order to leverage such chaotic characteristic of CA30 and implement it in an all-digital circuit, we utilize an XOR gate and an OR gate, as shown in Figure 1(b), where $U(p,q,r) = p \oplus (q||r)$. However, the synchronous circuit implementation of CA30 has deterministic behavior, as shown in Figure 1(a). From the aspect of TRNG design, we make the circuit realization of CA30 with a self-timed ring architecture. It is an asynchronous circuit shown in Figure 1(c), and denoted as ACR30. Note that in this schematic, we use annotations `Previous` and `Next` to represent the inputs p and r. The `Current` stands for the input q, as well as the output of the ACR30 circuit. Different from the synchronous design, the ACR30 has two additional signals: `Set` and `Pass/Capture`. Those two signals can be leveraged to control a self-timed oscillation circuit and sample the current state/output. Specifically, the `Set` signal can force reset the ACR30 circuit to the entire "0" state. The `Pass/Capture` signal is applied as the select signal of a 2–1 MUX. The `Pass` controls the circuit working in self-oscillation mode, and the `Capture` latches the output.

Figure 1(e) shows the circuit schematic of the proposed TRNG, which consists of nine ACR30 elements. All elements build up a self-timed ring circuit by linking each in a series, and every element has two neighborhoods. The `Current` signal of the middle ACR30 is not sampled by the output register. The functionality of this component is to break the all-zero stable state with the help of an all-zero state detector, as shown in Figure 1(d). The `Current` signals of the other eight ACR30s are sampled by an 8-bit register. The clock of this 8-bit register is the same signal as `Pass/Capture` signal shared by all the ACR30 elements except the middle ACR30. They have been used to generate 8-bits true random numbers. Additionally, this TRNG design only consists of 75 NAND gates.

2.2 Entropy Model

Entropy is the measurement value to determine the chaos of a system, for which a positive value means the system can generate unpredictable outputs. The entropy model of our proposed TRNG design has been discussed in [15]. Here we ignore the `Pass/Capture` and `Set` signals to simplify the analysis. Assume x_i, y_i, z_i and u_i $(i \in \{1,\dots 9\})$ are output of the XOR, inverter 1, inverter 2, and OR gates of each ACR30, respectively, shown in Equation 1. $s^+(x)$ is the standard step function for an inverter, which can be represented as $s^+(x) = 1 - s^-(x) = \begin{cases} 1, & if\ x > \theta \\ 0, & if\ x < \theta \end{cases}$. κ and γ denote the maximum voltage and propagation delay of each gate,

$$\begin{aligned}
\frac{dx_i}{dt} &= \kappa_{x_i}\left[s^+(z_{i-1})\,s^-(u_i) + s^-(z_{i-1})\,s^+(u_i)\right] - \gamma_{x_i}x_i \\
\frac{dy_i}{dt} &= \kappa_{y_i}s^-(x_i) - \gamma_{y_i}y_i \\
\frac{dz_i}{dt} &= \kappa_{z_i}s^-(y_i) - \gamma_{z_i}z_i \\
\frac{du_i}{dt} &= \kappa_{u_i}\left[1 - s^-(z_i)\,s^-(z_{i+1})\right] - \gamma_{u_i}u_i
\end{aligned} \tag{1}$$

We use an entropy model to calculate the information entropy. Assuming a random number set V collected at N timing points, where we have $\{V(t) : 1 \le t \le N\}$. $V(t)$ can form $N-m+1$ vectors denote as $v_m(i), (i \in \{1,\dots,N-m+1\})$ and each with length m, if we compare two vectors, r is the tolerance for accepting matches. We can denote $B^m(r)$ as the probability that two random vectors match for m points, and $A^m(r)$ as the probability that these two random vectors match for $m+1$ points. [14] provides more details about those two probabilities, following [16], we can use Sample Entropy ($SampEn$) to calculate the entropy of our proposed TRNG model. It can be represented as $SampEn(m,r,N) = -ln(\frac{A^m(r)}{B^m(r)})$. Here we have $SampEn \ge 0$, which concludes the TRNG system is able to generate entropy on its own with chaos [15]. In other words, it can generate random numbers based on its own chaotic dynamics.

3 Experimental Validation

We comprehensively evaluate the performance of the proposed TRNG design with the HSpice, ASIC, and FPGA implementations. For the HSpice simulation and ASIC test chip, 40nm TSMC technology nodes are used for the construction. To validate the feasibility of proposed scheme with reconfigurable devices, such as FPGA, we implement the proposed TRNG structure on an ML605 FPGA development tool kit embedded with a Virtex-6 FPGA. The FPGA is connected to a Dell precision 3630 tower with 16GB RAM through the PCIe port to collect the generated random numbers.

3.1 HSpice Simulation and Test Chip

We build the TRNG structure on HSpice to validate the functionality. For simulating the random behavior of the TRNG circuit, we provide the `Set` and `Pass/Capture` signals as external inputs, as shown in Figure 1. The results prove the randomness of the circuit output and the correct function of the control signals. To further investigate the sensitivity of the proposed TRNG circuit to the environmental conditions, such as the environment noise and the uncertainty of the rising/falling edge, we sample the outputs of the TRNG under different scenarios. The result demonstrates that even slightly various environmental conditions can flip the output of the TRNG circuit, which increases the unpredictability of the proposed TRNG.

Besides the simulation, we also validate the feasibility of the proposed TRNG on ASIC. We implement and fabricate a group of test chips with TSMC 40nm technology node. By exploring the comprehensive performance of the TRNG, we examine the TRNG on three typical test suites: NIST test suite SP800–22 [17], NIST test suite SP800–90B [18], and AIS31 test suite [19]. Further we compared the proposed TRNG with other state-of-the-art TRNG designs. The result is shown in Table 1. The TRNG proposed in this work outperforms other state-of-the-art designs: (1) it passes all the three test suites; (2) it consumes the lowest hardware footprint, only 270μm^2, but with high throughput, 1600MB/s; (3) it has the highest efficiency, 0.33pJ/bit, thanks to the pure digital circuits; (4) it can resist various attacks, which will be discussed in the next section.

Table 1: Performance of the proposed TRNG test chips and other state-of-the-art TRNG designs. Note that partial of the data in this table is summarized in [20].

	This Work	ISSCC'14 [20]	JSSC'12 [21]	VLSI'11 [22]	ISSCC'08 [23]	ISSCC'07 [24]	ISSCC'06 [25]	TC'03 [26]
Tech. Node	40nm	28/65nm	45nm	65nm	0.25μm	0.13μm	0.12μm	0.18μm
Noise Source	Chaotic CA Topo.	Jitter in RO	Meta-stability	Oxid Breakdown	Sin MOS-FET Noise	Meta-stability	Meta-stability	Jitter in RO
NIST SP800–22	Pass	Pass	Pass	Pass	Not Reported	5	Not Reported	Not Reported
NIST SP800–90B	Yes	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported
AIS-31	Yes	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported
Hardware footprint (μm^2)	270	375/960	4004	1200	1200	36300	9000	16000
Post Processing	Yes	No	No	No	Yes	No	Yes	No
Efficiency (pJ/bit))	0.33	23/57	2.9	181810	950	5000	250	230
Bit Rate (Mb/s)	1600	23.16/2.8	2400	0.011	2	0.2	0.2	10
Resis. to Attacks	Yes	Yes	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	No
FPGA Implementation	Yes	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported	Not Reported

3.2 FPGA Validation

The proposed TRNG can be implemented on reconfigurable devices since it only contains logic (digital) components. We construct the TRNG on a Virtex-6 FPGA to validate this feasibility, which only consumes 53 LUTs and 22 DFFs in total. This shows that similar to the scenario of test chips, the implementation of the proposed TRNG on FPGAs is still lightweight, costing a small circuit footprint.

As for the power consumption, the measurement from the Xilinx XPower Analyzer shows that the proposed TRNG circuit is executing with 2.05mW by clocking the `Pass/Capture` signal at 250MHz, i.e., the energy efficiency of the TRNG on the tested FPGA is $1pJ/bit$, which is slightly higher than the test chip implementation but still lower than other proposed TRNG designs, as shown in Table 1. Similarly, we examine the three test suites on the FPGA implementation of the proposed TRNG, and it passes all the tests successfully. For the resistance to existing TRNG attacks, we evaluate the proposed TRNG against the frequency injection attack, power attack, and thermal attack. By applying the frequency injection attack with several peak frequencies, it can be found that the entropy degradation caused by frequency injection is no more than 5%; the entropy even becomes higher in some injections. We test the TRNG implementation with various scales of power wasters and temperatures for the power and thermal attacks, respectively. The result demonstrates that the entropy of the proposed TRNG after attacks is not influenced obviously, and the TRNG can still pass all the test suites.

4 Conclusion

With the security issue of Cyber-Physical System (CPS) being paid attentions on, the effectiveness of TRNG is important as a critical security primitive, to ensure the reliability of the root-of-trust in CPS. In this work, we present a novel digital TRNG method, favoring implementations in a digital synthesis design-flow. Specifically, the TRNG is composed with a chaotic cellular automata topology with the CA30 rule, realizing an asynchronous self-timed ring circuit. The performance and the security of the proposed TRNG is comprehensively evaluated on both simulation and hardware platforms, with three state-of-the-art TRNG test suites. The results illustrate that the proposed TRNG can achieve high throughput with low hardware overhead and power consumption, compared to other state-of-the-art TRNGs. Moreover, the security of the proposed TRNG design is evaluated by under three different attacks, and the results demonstrate that the TRNG can resist these attacks with neglectable entropy loss.

References

[1] M. Broy, M. V. Cengarle, and E. Geisberger, “Cyber-physical systems: imminent challenges,” in *Monterey workshop*. Springer, 2012, pp. 1–28.

[2] C. Konstantinou, M. Maniatakos, F. Saqib, S. Hu, J. Plusquellic, and Y. Jin, “Cyber-physical systems: A security perspective,” in *2015 20th IEEE European Test Symposium (ETS)*. IEEE, 2015, pp. 1–8.

[3] J. Wurm, Y. Jin, Y. Liu, S. Hu, K. Heffner, F. Rahman, and M. Tehranipoor, “Introduction to cyber-physical system security: A cross-layer perspective,” *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 3, pp. 215–227, 2016.

[4] C. S. Petrie and J. A. Connelly, “A noise-based ic random number generator for applications in cryptography,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 615–621, 2000.

[5] D. E. Holcomb, W. P. Burleson, K. Fu *et al.*, “Initial sram state as a fingerprint and source of true random numbers for rfid tags,” in *Proceedings of the Conference on RFID Security*, vol. 7, no. 2, 2007, p. 01.

[6] B. Sunar, W. J. Martin, and D. R. Stinson, “A provably secure true random number generator with built-in tolerance to active attacks,” *IEEE Transactions on computers*, vol. 56, no. 1, 2007.

[7] S. Best and X. Xu, “An all-digital true random number generator based on chaotic cellular automata topology,” in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[8] R. A. Schulz, “Random number generator circuit,” Feb. 27 1990, uS Patent 4,905,176.

[9] A. Cherkaoui, V. Fischer, L. Fesquet, and A. Aubert, “A very high speed true random number generator with entropy assessment,” in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2013, pp. 179–196.

[10] M. Dichtl and J. D. Golić, “High-speed true random number generation with logic gates only,” in *Cryptographic Hardware and Embedded Systems-CHES 2007*. Springer, 2007, pp. 45–62.

[11] P. D. Hortensius, R. D. McLeod, and H. C. Card, “Parallel random number generation for vlsi systems using cellular automata,” *IEEE Transactions on Computers*, no. 10, pp. 1466–1473, 1989.

[12] L. Petrica, “Fpga optimized cellular automaton random number generator,” *Journal of Parallel and Distributed Computing*, vol. 111, pp. 251–259, 2018.

[13] E. Jen, “Aperiodicity in one-dimensional cellular automata,” *Physica D: Nonlinear Phenomena*, vol. 45, no. 1-3, pp. 3–18, 1990.

[14] Y. Luo, W. Wang, S. Best, Y. Wang, and X. Xu, “A high-performance and secure trng based on chaotic cellular automata topology,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4970–4983, 2020.

[15] E. Farcot, S. Best, R. Edwards, I. Belgacem, X. Xu, and P. Gill, “Chaos in a ring circuit,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 4, p. 043103, 2019.

[16] J. S. Richman and J. R. Moorman., *Physiological time-series analysis using approximate entropy and sample entropy*. American Journal of Physiology-Heart and Circulatory Physiology, 278.6 (2000): H2039-H2049. [Online]. Available: http://books.google.com/books?id=W-xMPgAACAAJ

[17] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” Booz-allen and hamilton inc mclean va, Tech. Rep. 800-22 Rev 1a, 2001.

[18] K. McKay, “User‘s guide to running the draft nist sp 800-90b entropy estimation suite,” Tech. Rep., 2016.

[19] W. Killmann and W. Schindler, “Ais 31: Functionality classes and evaluation methodology for true (physical) random number generators, version 3.1,” *Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn*, 2001.

[20] K. Yang, D. Fick, M. B. Henry, Y. Lee, D. Blaauw, and D. Sylvester, “16.3 a 23mb/s 23pj/b fully synthesized true-random-number generator in 28nm and 65nm cmos,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*. IEEE, 2014, pp. 280–281.

[21] S. K. Mathew, S. Srinivasan, M. A. Anders, H. Kaul, S. K. Hsu, F. Sheikh, A. Agarwal, S. Satpathy, and R. K. Krishnamurthy, “2.4 gbps, 7 mw all-digital pvt-variation tolerant true random number generator for 45 nm cmos high-performance microprocessors,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, 2012.

[22] N. Liu, N. Pinckney, S. Hanson, D. Sylvester, and D. Blaauw, “A true random number generator using time-dependent dielectric breakdown,” in *VLSI Circuits (VLSIC), 2011 Symposium on*. IEEE, 2011, pp. 216–217.

[23] M. Matsumoto, S. Yasuda, R. Ohba, K. Ikegami, T. Tanamoto, and S. Fujita, “1200μm 2 physical random-number generators based on sin mosfet for secure smart-card application,” in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*. IEEE, 2008, pp. 414–624.

[24] C. Tokunaga, D. Blaauw, and T. Mudge, “True random number generator with a metastability-based quality control,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 78–85, 2008.

[25] R. Brederlow, R. Prakash, C. Paulus, and R. Thewes, “A low-power true random number generator using random telegraph noise of single oxide-traps,” in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*. IEEE, 2006, pp. 1666–1675.

[26] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, “A high-speed oscillator-based truly random number source for cryptographic applications on a smart card ic,” *IEEE transactions on computers*, vol. 52, no. 4, pp. 403–409, 2003.

Bayesian Sharing Grouped Convolution

Tinghuan Chen

CSE Department, The Chinese University of Hong Kong

Abstract

Compared with traditional convolutions, grouped convolutional neural networks are promising for both model performance and network parameters. However, existing models with the grouped convolution still have parameter redundancy. In this paper, concerning the grouped convolution, we propose a sharing grouped convolution structure to reduce parameters. To efficiently eliminate parameter redundancy and improve model performance, we propose a Bayesian sharing framework to transfer the vanilla grouped convolution to be the sharing structure. Intra-group correlation and inter-group importance are introduced into the prior of the parameters. We handle the Maximum Type II likelihood estimation problem of the intra-group correlation and inter-group importance by a group LASSO type algorithm. The prior mean of the sharing kernels is iteratively updated. Extensive experiments are conducted to demonstrate that on different grouped convolutional neural networks, the proposed sharing grouped convolution structure with the Bayesian sharing framework can reduce parameters and improve prediction accuracy.

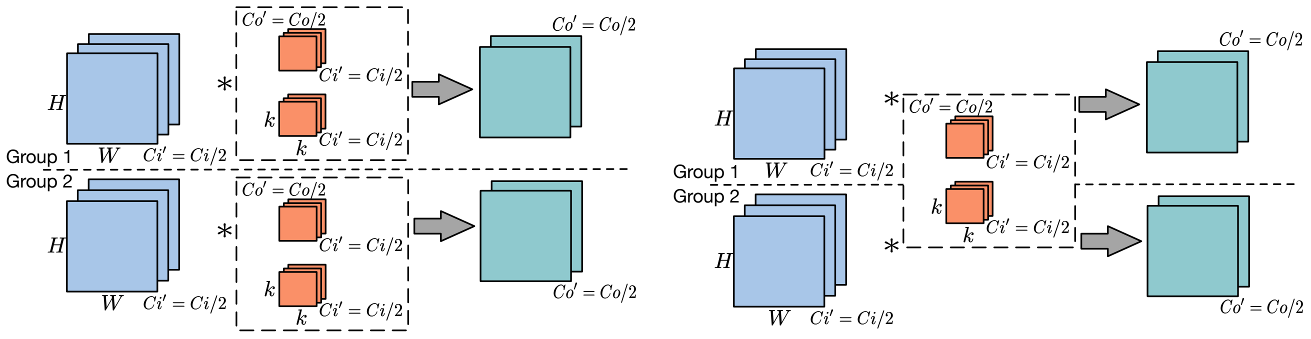
1 Introduction

Convolutional neural networks (CNNs) have achieved impressive successes in various applications of computer vision, such as object recognition [1, 2], object detection [3, 4, 5], and autonomous driving [6]. To handle complicated applications, CNN models become deeper and wider, which causes massive network parameters. The massive network parameters, however, bring huge challenges to model storage, data transfer, computation overhead, and energy consumption [7, 8]. Besides, the massive network parameters may contain redundancy, which causes overfitting and performance degradation.

The grouped convolution has been adopted to decrease parameter redundancy and improve accuracy in popular CNNs, such as AlexNet [9] and ResNeXt [10]. The vanilla grouped convolution is shown in Figure 1(a), where the inputs, the weights, and the outputs are divided into several groups to perform the convolution operation. In practice, the grouped convolution is proven to be able to alleviate overfitting and improve the model accuracy.

Although the grouped convolution has the aforementioned advantages, the network parameters may still have redundancy. Various arts are proposed to reduce parameter redundancy [11, 12]. Although existing compression methods have good compression performance in the traditional convolution models, they may lead to performance degradations while being applied to grouped convolutions since they ignore the diversities of importances and correlations (*i.e.*, inter-group importance and intra-group correlation) among the different parameter groups.

To eliminate parameter redundancy and improve efficiency of the grouped convolution, in this paper, we propose a sharing grouped convolution structure, a novel and simple architecture, to reduce parameters as shown in Figure 1(b). A Bayesian grouped convolution sharing framework is proposed to transfer the vanilla grouped convolution to be the sharing structure. Intra-group correlation and inter-group importance are introduced into the prior of network parameters. We handle the Maximum Type II likelihood estimation problem of the intra-group correlation and inter-group importance by a group LASSO type algorithm [13]. The prior mean is iteratively updated with the posterior mean and the inter-group importance learned in the previous iteration. We conduct experiments on CIFAR-10 and CIFAR-100 [14], to validate our proposed sharing grouped convolution structure with Bayesian sharing framework. Experiments demonstrate that our framework can reduce parameters significantly and improve model accuracies.



(a)

(b)

Figure 1: The vanilla grouped convolution and our proposed sharing grouped convolution (2 groups, the blue boxes are the input features, the orange boxes are the kernels, the green boxes are the output features. H and W are height and weight of the input features. Ci and Co are the numbers of the input and output channels. Ci' and Co' are the numbers of the input and output channels in each group. k is kernel size): (a) The vanilla grouped convolution. Each group has its own weights. (b) Our proposed sharing grouped convolution. All of these groups share the same weights.

2 Sharing Grouped Convolution

In this section, a sharing grouped convolution structure is proposed to reduce parameter redundancy, improve parameter efficiency. Then the number of parameters in it is analyzed to illustrate the compression performance.

To demonstrate the vanilla grouped convolution, the variation from ResNet to ResNeXt [15, 10] is taken as an example. Figure 2 shows their basic block, which is repeatedly stacked with different configurations to the whole model. The basic block contains a shortcut and three convolutional layers, whose all kernels are represented by a box in each layer. In ResNet, the basic block is shown in Figure 2(a), where there are three traditional convolutional layers. In order to transfer ResNet to ResNeXt, in each block, the second convolutional layer is transferred as the vanilla grouped convolution by dividing the 64 channels into 16 groups, and each group has 4 channels for this example as shown in Figure 2(b). Compared with the traditional convolution, the vanilla grouped convolution adopts the sparse convolution connections between input and output channels, by dividing the input channels, output channels, and their connections into several groups. According to Figure 2, the parameter number for the second convolutional layer is reduced from $64 \times 3 \times 3 \times 64 = 36864$ of ResNet to $16 \times 4 \times 3 \times 3 \times 4 = 2304$ of ResNeXt in the convolutional layer.

In order to further reduce the parameter number and improve parameter efficiency in the vanilla grouped convolution, we propose a sharing grouped convolution structure. Specifically, all groups share the same parameters so that the same parameters can be used to extract features and pass information among different groups. It has the same manner with [16] to improve parameter efficiency. Then in each basic block, the vanilla grouped convolution (the second layer) as shown in Figure 2(b) will be transferred as the sharing grouped convolution as shown in Figure 2(c). The parameter number for the second convolutional layer is reduced from $16 \times 4 \times 3 \times 3 \times 4 = 2304$ to $4 \times 3 \times 3 \times 4 = 144$. Note that compared with the vanilla grouped convolution, our proposed sharing grouped convolution does not reduce computational complexity. However, as shown in Figure 2(c), the parameters are shared among different groups. Therefore, the efficiency of parameters is improved and the parameter redundancy can be reduced. Besides, the sharing grouped convolution can facilitate the weights reusing strategy in the hardware level implementations so that the actual number of memory accesses decreases significantly and the inference runtime reduces. As comparison, we show the numbers of parameters of basic blocks in ResNet, ResNext and the sharing ResNeXt in Table 2. Compared with ResNet, ResNeXt has fewer parameters, and the proposed sharing ResNeXt can further reduce the number of parameters.

Although the proposed sharing grouped convolution structure can reduce parameters and improve the efficiency of parameters, directly training models with the group convolution structure may cause performance degradation

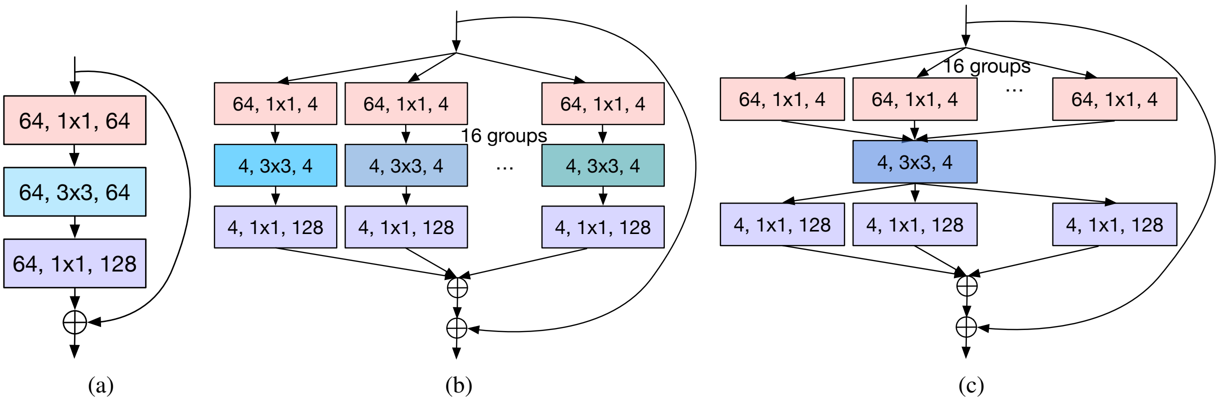


Figure 2: The basic block contains a shortcut and three convolutional layers (the boxes indicate the convolutional kernels [#input channel, kernel size, #output channel] for each layer): (a) the three convolutional layers in ResNet; (b) the two convolutional layers and one vanilla grouped convolutional layer with 16 groups in ResNeXt; (c) the two convolutional layers and one sharing grouped convolutional layer with 16 groups in the sharing ResNeXt.

Table 2: The numbers of parameters of basic blocks in ResNet, ResNext and the sharing ResNeXt with $g = 16$.

Type	ResNet	ResNeXt	Sharing ResNeXt
conv	$64 \times 1 \times 1 \times 64$	$64 \times 1 \times 1 \times 64$	$64 \times 1 \times 1 \times 64$
(g)conv	$64 \times 3 \times 3 \times 64$	$16 \times 4 \times 3 \times 3 \times 4$	$4 \times 3 \times 3 \times 4$
conv	$64 \times 1 \times 1 \times 128$	$64 \times 1 \times 1 \times 128$	$64 \times 1 \times 1 \times 128$
shortcut	$64 \times 1 \times 1 \times 128$	$64 \times 1 \times 1 \times 128$	$64 \times 1 \times 1 \times 128$
total	57344	22784	**20624**

since the correlation among parameters and groups does not be considered.

3 Bayesian Sharing Framework

To transfer the vanilla grouped convolution into the sharing structure, a naïve method is directly constructing a network with the proposed sharing grouped convolution structure then training it. However, this method may cause performance degradation. To avoid performance degradation, we adopt a separate-merge methodology [17], that is updating independently parameters among all groups in the back-propagation stage and computing loss function value by the shared parameters in the forward propagation stage. Based on the separate-merge methodology, a typical method is indiscriminately averaging the parameters among different groups in the forward propagation stage. This is quite straightforward but it ignores the diversities of different groups.

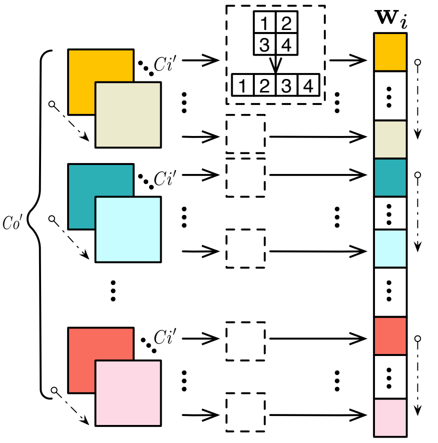
In this section, to efficiently eliminate parameter redundancy and improve model performance, we introduce the **intra-group correlation** and **inter-group importance** of parameters. Then we propose a Bayesian sharing framework. Some notations used in this paper are listed in Figure 3(a) and visualized in Figure 1.

3.1 Intra-group Correlation and Inter-group Importance

To introduce **intra-group correlation** and **inter-group importance**, a prior distribution on model parameters $\mathscr{P}(\boldsymbol{w})$ is firstly introduced in our framework. Following previous arts [18, 19, 20], $\mathscr{P}(\boldsymbol{w})$ is defined to be a multivariate Gaussian distribution. For the convenience of expressions, network parameters are reshaped to be vectors. As shown in Figure 3(b), in each group, the kernels in each channel are flattened to be a vector. Some notations are explained in Figure 3(a). Then they are concatenated sequentially to be a vector. Considering that the features are extracted independently from different groups in the vanilla grouped convolution as shown in Figure 1(a), we assume any two network parameters from different groups are independent, *i.e.*, $\mathscr{P}(\boldsymbol{w}) = \prod_i^g \mathscr{P}(\boldsymbol{w}_i)$. Therefore, the prior distribution

Name	Definition
$\boldsymbol{w}$	parameters in one grouped convolutional layer
g	# of groups in one grouped convolutional layer
$\boldsymbol{B}_i$	intra-group correlation of the group i
γ_i	inter-group importance of the group i
$\boldsymbol{w}_i$	parameters of the group i
$\boldsymbol{w}_b$	the sharing parameters of g groups
k	kernel size
Ci'	# of input channels in each group
Co'	# of output channels in each group
H	height of input feature
W	width of input feature

(a)



(b)

Figure 3: (a) List of Notations; (b) Reshape the parameter tensor of one group as a vector, with Ci' input channels and Co' output channels. The kernel size is 2×2. The arrows show the flattening order.

of network parameters in the group i is defined as $\mathscr{P}(\boldsymbol{w}_i;\gamma_i,\boldsymbol{B}_i) \sim \mathscr{N}(\boldsymbol{\mu}_{\boldsymbol{w}_i},\boldsymbol{\Sigma}_{\boldsymbol{w}_i})$, where $\boldsymbol{\Sigma}_{\boldsymbol{w}_i} \triangleq \gamma_i\boldsymbol{B}_i$, $\boldsymbol{w}_i \in \mathbb{R}^{NCo'}$, and $\boldsymbol{B}_i \in \mathbb{R}^{NCo' \times NCo'}$ with $N \triangleq k^2Ci'$. $\boldsymbol{B}_i$ is a positive definite matrix which captures the correlations of the parameters in group i, termed as **intra-group correlation**. γ_i is a coefficient reflecting the relative importance of group i in comparison with other groups, termed as **inter-group importance**. γ_i also indicates the importance of the group i while passing messages or knowledges in the model during inference. $\boldsymbol{\mu}_{\boldsymbol{w}_i}$ is the mean vector of the network parameters $\boldsymbol{w}_i$ in the group i. And $\boldsymbol{\Sigma}_{\boldsymbol{w}_i}$ is the covariance matrix of the network parameters $\boldsymbol{w}_i$ in the group i. For the convolutional layer with g groups, the prior distribution of network parameters is

$$\mathscr{P}(\boldsymbol{w};\boldsymbol{B},\boldsymbol{\gamma}) \sim \mathscr{N}(\boldsymbol{\mu}_{\boldsymbol{w}},\boldsymbol{\Sigma}_{\boldsymbol{w}}), \tag{2}$$

where $\boldsymbol{\mu}_{\boldsymbol{w}} = [\boldsymbol{\mu}_{\boldsymbol{w}_1}^\top,\boldsymbol{\mu}_{\boldsymbol{w}_2}^\top,\cdots,\boldsymbol{\mu}_{\boldsymbol{w}_g}^\top]^\top$ is the mean vector of the network parameters $\boldsymbol{w}$. $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \text{diag}[\boldsymbol{\Sigma}_{\boldsymbol{w}_1},\boldsymbol{\Sigma}_{\boldsymbol{w}_2},\cdots,\boldsymbol{\Sigma}_{\boldsymbol{w}_g}]$ is the covariance matrix of $\boldsymbol{w}$, which is a block diagonal matrix with principal diagonal blocks being $\boldsymbol{\Sigma}_{\boldsymbol{w}_1},\boldsymbol{\Sigma}_{\boldsymbol{w}_2},\cdots,\boldsymbol{\Sigma}_{\boldsymbol{w}_g}$. $\boldsymbol{B} \triangleq \{\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_g\}$ and $\boldsymbol{\gamma} \triangleq [\gamma_1,\gamma_2,\cdots,\gamma_g]^\top$. The intra-group correlation $\boldsymbol{B}_i$ and the inter-group importance γ_i are determined by maximizing Type II likelihood [13] as shown in Formulation (3).

$$\max_{\boldsymbol{B},\boldsymbol{\gamma}} \quad \ln \int \mathscr{P}(\mathscr{Y}|\mathscr{X},\boldsymbol{w})\mathscr{P}(\boldsymbol{w};\boldsymbol{B},\boldsymbol{\gamma})\mathrm{d}\boldsymbol{w}, \tag{3}$$

where $\mathscr{Y}$ and $\mathscr{X}$ are the output and input features, respectively. $\mathscr{P}(\boldsymbol{w};\boldsymbol{B},\boldsymbol{\gamma})$ satisfies multivariate Gaussian distribution with hyper-parameters $\boldsymbol{B}$ and $\boldsymbol{\gamma}$ defined in Equation (2).

According to Formulation (3), to obtain the intra-group correlation $\boldsymbol{B}_i$ and the inter-group importance γ_i, we need give a concrete form of $\mathscr{P}(\mathscr{Y}|\mathscr{X},\boldsymbol{w})$. Nevertheless, in practice, because of non-linear operations in CNN models, it is hard to obtain the closed form of the likelihood function $\mathscr{P}(\mathscr{Y}|\mathscr{X},\boldsymbol{w})$ and the integral of the marginal likelihood in Formulation (3) is intractable in neural networks [21]. Like in [19], we consider the linear relationship between the input and the output features of each layer before a nonlinearity is applied.

3.2 Maximum Type II Likelihood Estimation

We consider the linear relationship between the input and the output features of each layer before a nonlinearity is applied. To represent the vanilla grouped convolution in the form of matrix-vector multiplication, we reshape the input features as shown in Figure 4:

- In Step 1, we reshape the input features of one group to be a block-diagonal matrix. As the parameter kernel window slides on the input feature, the corresponding features are flattened to be a vector with length k^2. Therefore, we flatten the feature in one channel to be an $HW \times k^2$ matrix. Then the feature matrices of all Ci' channels are reshaped to be a block-diagonal matrix.

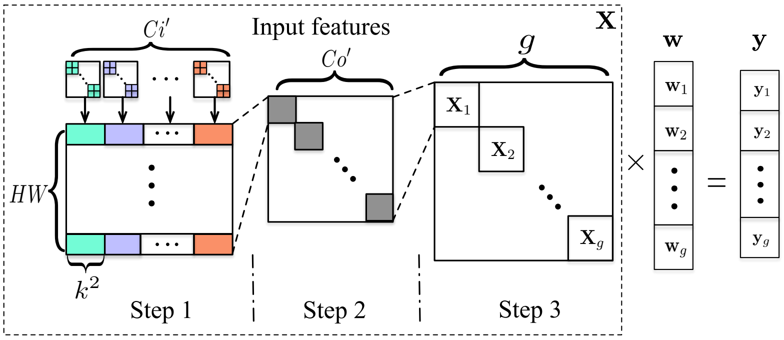


Figure 4: Reshape input features. The vanilla grouped convolution operation is transformed as matrix-vector multiplication.

- In Step 2, we duplicate the input feature block matrix by Co' times to generate a larger block-diagonal matrix.
- In Step 3, we place the block-diagonal matrices of the g groups at the diagonal of the final feature matrix. The parameters are also reshaped in the same manner, as mentioned above in Figure 3(b).

For each group, the matrix-vector multiplication with model error $\boldsymbol{v}_i$ can be represented as $\boldsymbol{y}_i = \boldsymbol{X}_i \boldsymbol{w}_i + \boldsymbol{v}_i$, where $\boldsymbol{y}_i \in \mathbb{R}^{MCo'}$ and $\boldsymbol{X}_i \in \mathbb{R}^{MCo' \times NCo'}$ represent the reshaped outputs and inputs respectively, with $M \triangleq HW$. For a layer with g groups, the vanilla grouped convolution is $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{v}$, where $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_g^\top]^\top$, $\boldsymbol{X} = \mathrm{diag}[\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_g]$, and $\boldsymbol{v} = [\boldsymbol{v}_1^\top, \cdots, \boldsymbol{v}_g^\top]^\top$. The model error $\boldsymbol{v}$ is assumed to follow independent identical Gaussian distribution, *i.e.*, $\mathscr{P}(\boldsymbol{v}) \sim \mathscr{N}(\boldsymbol{0}, \lambda \boldsymbol{I})$, where λ is a hyper-parameter controling the precision of model error. $\boldsymbol{I}$ is an identity matrix. The concrete form of the likelihood function in Formulation (3) can be obtained as follows:

$$\mathscr{P}(\mathscr{Y}|\mathscr{X}, \boldsymbol{w}) = \mathscr{P}(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}; \lambda) \sim \mathscr{N}(\boldsymbol{X}\boldsymbol{w}, \lambda \boldsymbol{I}). \tag{4}$$

According to the network parameters prior $\mathscr{P}(\boldsymbol{w}; \gamma, \boldsymbol{B})$ defined in Equation (2) and the likelihood function $\mathscr{P}(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})$ defined in Equation (4), the posterior of network parameters also follows multivariate Gaussian distribution $\mathscr{P}(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\gamma}, \boldsymbol{B}, \lambda) \sim \mathscr{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ are represented as follows [13]:

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{X}^\top (\lambda \boldsymbol{I} + \boldsymbol{X} \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{X}^\top)^{-1} (\boldsymbol{y} - \boldsymbol{X} \boldsymbol{\mu}_{\boldsymbol{w}}), \boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1} + \frac{1}{\lambda} \boldsymbol{X}^\top \boldsymbol{X})^{-1}, \tag{5}$$

where $\boldsymbol{\mu} \triangleq [\boldsymbol{\mu}_1^\top, \cdots, \boldsymbol{\mu}_g^\top]^\top$, and $\boldsymbol{\Sigma} \triangleq \mathrm{diag}[\boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_g]$. $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the posterior mean and the covariance matrix of network parameters in the group i, respectively.

Now to determine the intra-group correlation $\boldsymbol{B}_i$ and the inter-group importance γ_i, we can transform Formulation (3) as follows [22, 23, 24]:

$$\min_{\boldsymbol{B}, \boldsymbol{\gamma}, \lambda} \quad \mathscr{L}(\boldsymbol{B}, \boldsymbol{\gamma}, \lambda), \tag{6}$$

where

$$\mathscr{L}(\boldsymbol{B}, \boldsymbol{\gamma}, \lambda) \triangleq -2 \ln \mathscr{P}(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{B}, \boldsymbol{\gamma}, \lambda) = \ln |\lambda \boldsymbol{I} + \boldsymbol{X} \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{X}^\top| + (\boldsymbol{y} - \boldsymbol{X} \boldsymbol{\mu}_{\boldsymbol{w}})^\top (\lambda \boldsymbol{I} + \boldsymbol{X} \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{X}^\top)^{-1} (\boldsymbol{y} - \boldsymbol{X} \boldsymbol{\mu}_{\boldsymbol{w}}). \tag{7}$$

Since it has the ability to adaptively learn and exploit intra-group correlation for better performance and only takes few iterations, in next section, we illustrate how to use a group LASSO type method to handle Formulation (6) so that the intra-group correlation $\boldsymbol{B}_i$, the inter-group importance γ_i and the hyper-parameter λ can be well determined.

3.3 Optimization via Group LASSO Type Algorithm

In this subsection, we follow the work [25] and use a group LASSO type algorithm to determine hyper-parameters so that it can achieve fast convergence. The main idea is shown as follows: Firstly, we find the upper-bound of the cost function $\mathscr{L}(\boldsymbol{B},\boldsymbol{\gamma},\lambda)$ defined in Equation (7). Then the upper-bound can be transformed to be a group LASSO problem. As a result, we can solve it with a typical group LASSO solver more efficiently.

In order to find an appropriate upper-bound of $\mathscr{L}(\boldsymbol{B},\boldsymbol{\gamma},\lambda)$, we introduce a temporary function $h(\boldsymbol{\alpha}) \triangleq [\frac{1}{\lambda}||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\mu}_{\boldsymbol{w}}-\boldsymbol{X}\boldsymbol{\alpha}||_2^2+\boldsymbol{\alpha}^\top\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\boldsymbol{\alpha}]$. $\boldsymbol{\alpha}$ is defined as a temporary variable. There is a global minimum $\boldsymbol{\alpha}_0$, *i.e.*, $h(\boldsymbol{\alpha}_0) \leq h(\boldsymbol{\alpha})$, with the first derivative $h(\boldsymbol{\alpha}_0)' = \boldsymbol{0}$. Substituting $\boldsymbol{\alpha}_0$ into the function $h(\boldsymbol{\alpha})$ and using Woodbury matrix identity [26] lead to the upper-bound of Equation (7)

$$\mathscr{U}\mathscr{L}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{B},\lambda) = \ln|\lambda\boldsymbol{I}+\boldsymbol{X}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{X}^\top|+\frac{1}{\lambda}||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\mu}_{\boldsymbol{w}}-\boldsymbol{X}\boldsymbol{\alpha}||_2^2+\boldsymbol{\alpha}^\top\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\boldsymbol{\alpha}.$$

Here, we temporarily fix $\boldsymbol{B}$ and λ. Then instead of directly optimizing Formulation (6), we minimize the upper-bound as follows:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\gamma}} \quad \mathscr{U}\mathscr{L}(\boldsymbol{\alpha},\boldsymbol{\gamma}). \tag{8}$$

Furthermore, considering the term $(1/\lambda)||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\mu}_{\boldsymbol{w}}-\boldsymbol{X}\boldsymbol{\alpha}||_2^2$ is independent of $\boldsymbol{\gamma}$ in $\mathscr{U}\mathscr{L}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{B},\lambda)$, Formulation (8) can be handled in two steps alternatively and iteratively.

In the first step, $\mathscr{U}\mathscr{L}(\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{B},\lambda)$ can be transformed as Equation (9).

$$f(\boldsymbol{\alpha}) = \min_{\boldsymbol{\gamma},\boldsymbol{z}\geq\boldsymbol{0}} \quad \boldsymbol{\alpha}^\top\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1}\boldsymbol{\alpha}+\boldsymbol{z}^\top\boldsymbol{\gamma}-g^*(\boldsymbol{z}) = \min_{\boldsymbol{\gamma},\boldsymbol{z}\geq\boldsymbol{0}} \quad \sum_{i=0}^{g}\left(\frac{\boldsymbol{\alpha}_i^\top\boldsymbol{B}_i^{-1}\boldsymbol{\alpha}_i}{\gamma_i}+z_i\gamma_i\right)-g^*(\boldsymbol{z}), \tag{9}$$

where $\boldsymbol{z} = [z_1,z_2,\cdots,z_g]^\top$. Minimizing Equation (9) *w.r.t.* $\boldsymbol{\gamma}$, we have

$$\gamma_i = z_i^{-\frac{1}{2}}\sqrt{\boldsymbol{\alpha}_i^\top\boldsymbol{B}_i^{-1}\boldsymbol{\alpha}_i}, \quad i = 1,2,\cdots,g. \tag{10}$$

However, γ_i relies on z_i. According the duality property [27], we can obtain

$$z_i = \mathrm{Tr}[\boldsymbol{B}_i\boldsymbol{X}_i^\top(\lambda\boldsymbol{I}+\boldsymbol{X}_i\boldsymbol{\Sigma}_{\boldsymbol{w}_i}\boldsymbol{X}_i^\top)^{-1}\boldsymbol{X}_i]. \tag{11}$$

According to Equation (10) and Equation (11), $\boldsymbol{\gamma}$ relies on $\boldsymbol{z}$ and $\boldsymbol{z}$ relies on $\boldsymbol{\gamma}$ ($\boldsymbol{\Sigma}_{\boldsymbol{w}}$). Therefore, in the first step, we minimize (9) by updating $\boldsymbol{\gamma}$ and $\boldsymbol{z}$, alternatively.

In the second step, after $\boldsymbol{\gamma}$ and $\boldsymbol{z}$ are determined, Formulation (8) is transformed as follows:

$$\min_{\boldsymbol{\alpha}} \quad ||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\mu}_{\boldsymbol{w}}-\boldsymbol{X}\boldsymbol{\alpha}||_2^2+\lambda\sum_{i=1}^{g}2z_i^{\frac{1}{2}}\sqrt{\boldsymbol{\alpha}_i^\top\boldsymbol{B}_i^{-1}\boldsymbol{\alpha}_i}. \tag{12}$$

Formulation (12) is an implicit group LASSO formulation, which can handled by calling classical group LASSO solver (*e.g.*, [28]) to determine $\boldsymbol{\alpha}$.

Note that during the above process, we fix the intra-group correlation $\boldsymbol{B}$ and the hyper-parameter λ. In fact, the hyper-parameter λ can be automatically determined by a group LASSO solver [28]. Besides, according to [25], since $\boldsymbol{\alpha}$ has the approximate covariance with $\boldsymbol{w}$, $\boldsymbol{B}$ can be approximately estimated by $\boldsymbol{\alpha}$ from the previous iteration, that is

$$\boldsymbol{B}_i = \frac{1}{\gamma_i}\boldsymbol{\Sigma}_{\boldsymbol{w}_i} \approx \frac{1}{\gamma_i}\mathbb{E}[(\boldsymbol{\alpha}_i-\mathbb{E}(\boldsymbol{\alpha}_i))(\boldsymbol{\alpha}_i-\mathbb{E}(\boldsymbol{\alpha}_i))^\top]. \tag{13}$$

In particular, according to [29], the first-order auto-regressive process corresponding to the Toeplitz matrix is more sufficient to capture intra-group correlation. Therefore, the intra-group correlation matrix $\boldsymbol{B}_i$ is replaced by $\hat{\boldsymbol{B}}_i$ as follows:

$$\hat{\boldsymbol{B}}_i = \mathrm{Toeplitz}([1,\mathrm{r},\cdots,\mathrm{r}^{\mathrm{NCo}'-1}]), \tag{14}$$

where $r = \bar{m}_1/\bar{m}_0$, $\bar{m}_0$ and $\bar{m}_1$ are the averages of elements along the main diagonal and the main sub-diagonal of $\boldsymbol{B}_i$, respectively. In summary, the developed group LASSO type algorithm flow is shown in Algorithm 1.

Algorithm 1 Group LASSO to handle Formulation (6).

```
Require: $\boldsymbol{X}$, $\boldsymbol{y}$ from one grouped convolutional layer, network parameters $\boldsymbol{w}$.
Initialize $\boldsymbol{B}$, $\boldsymbol{\gamma}$, $\boldsymbol{z}$ and $\lambda$.
repeat
    Update $\boldsymbol{\gamma}$ by Equation (10);
    Update $\boldsymbol{z}$ by Equation (11);
    Solve Formulation (12) to obtain $\boldsymbol{\alpha}$ and $\lambda$;
    Update $\boldsymbol{B}_i = \hat{\boldsymbol{B}}_i$ by Equations (13) and (14);
until Convergence

return hyper-parameters $\boldsymbol{B}$, $\boldsymbol{\gamma}$ and $\lambda$.
```

3.4 Overall flow

In this subsection, we will give an overall flow about how to share parameters among different groups so that the vanilla grouped convolution can be transferred as the sharing structure.

After $\boldsymbol{B}$, $\boldsymbol{\gamma}$ and λ are determined by Algorithm 1, in each group, model parameters $\boldsymbol{w}_i$ can be determined by the posterior mean as shown in Equation (5), that is $\boldsymbol{w}_i = \boldsymbol{\mu}_i$. To share the parameters among different groups in one grouped convolutional layer, the mean of the sharing parameters $\boldsymbol{\mu}_{\boldsymbol{w}_b}$ is defined as a prior mean as follows:

$$\boldsymbol{\mu}_{\boldsymbol{w}_b} = \frac{\sum_i^g \gamma_i \boldsymbol{w}_i}{\sum_i^g \gamma_i}. \tag{15}$$

The mean is the weighted average of all network parameters obtained in the last iteration, with the inter-group importance γ_i. Then in Equations (7) and (8), the prior mean is $\boldsymbol{\mu}_{\boldsymbol{w}} = \mathbf{1}_g \otimes \boldsymbol{\mu}_{\boldsymbol{w}_b} = [\boldsymbol{\mu}_{\boldsymbol{w}_b}^\top, \boldsymbol{\mu}_{\boldsymbol{w}_b}^\top, \cdots, \boldsymbol{\mu}_{\boldsymbol{w}_b}^\top]^\top$, and $\mathbf{1}_g \in \mathbb{R}^g$ is a vector whose all elements are 1. $\otimes$ represents the Kronecker product. The sharing process is shown in Algorithm 2. As shown in Figure 5, initially, all groups have different parameters. After few iterations, parameters will gradually become the same by our proposed Bayesian sharing framework. In particular, the mean sharing method is a special case of our proposed Bayesian sharing method, *i.e.*, $\boldsymbol{\gamma} \equiv \mathbf{1}_g$.

Algorithm 2 Bayesian sharing framework

```
Require: $\boldsymbol{X}$, $\boldsymbol{y}$ from one grouped convolutional layer, network parameters $\boldsymbol{w}$.
Initialize $\boldsymbol{\mu}_{\boldsymbol{w}_b} = \sum_i^g \boldsymbol{w}_i / g$, $\boldsymbol{\mu}_{\boldsymbol{w}} = \mathbf{1}_g \otimes \boldsymbol{\mu}_{\boldsymbol{w}_b}$;
repeat
    Update $\boldsymbol{B}$, $\boldsymbol{\gamma}$ and $\lambda$ by Algorithm 1;
    Update model parameters $\boldsymbol{w}_i$ by the posterior mean in Equation (5);
    Update the sharing model parameters $\boldsymbol{\mu}_{\boldsymbol{w}_b}$ by Equation (15) and $\boldsymbol{\mu}_{\boldsymbol{w}} = \mathbf{1}_g \otimes \boldsymbol{\mu}_{\boldsymbol{w}_b}$;
until Convergence
return The sharing weights $\boldsymbol{w}_b = \boldsymbol{\mu}_{\boldsymbol{w}_b}$.
```

For the whole CNN model, we adopt a separate-merge methodology [17] to share weights in all grouped convolutional layers, that is separately updating parameters by loss function in the back-propagation stage and computing loss function value in the forward propagation stage. Given a pre-trained CNN model, we fix model parameters in non-grouped convolutional layers and update model parameters in all grouped convolutional layers by our proposed Bayesian sharing method as shown in Algorithm 2 from front layers to back layers sequentially in the forward propagation stage.

The loss value is calculated by all updated shared grouped convolution parameters and other fixed model parameters. Then the loss value is used to updated all model parameters. By performing this sharing process for few epochs, the final sharing model can be obtained.

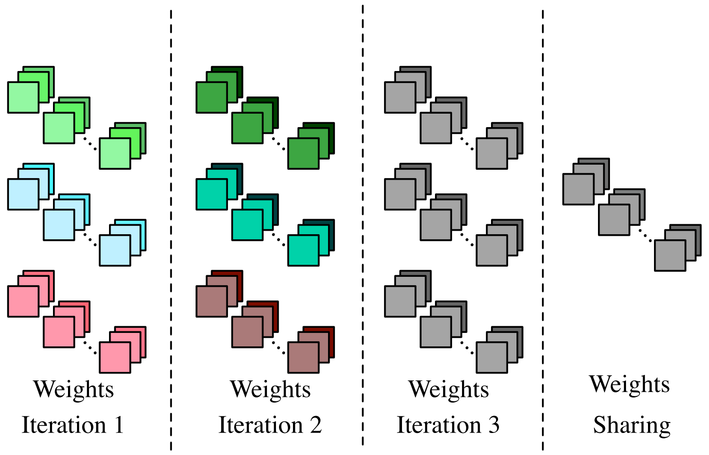


Figure 5: The sharing process of grouped convolution parameters. Green, blue and red boxes represent parameters (kernels) in three groups. After few iterations, all groups have the same kernels (grey boxes), which are shared.

4 Experimental Results

In this section, we apply our Bayesian sharing framework on some popular grouped convolutional neural networks, including ResNeXt [10], ShuffleNet [30] and G-DenseNet [31, 32]. We test them on CIFAR-10 and CIFAR-100 [14]. As an ablation study, to clarify the impact of the proposed Bayesian sharing framework, the directly trained sharing grouped convolutional neural networks and the mean sharing method are also implemented for comparison. The direct training method constructs a network with the proposed sharing structure and then trains it. The mean sharing method trains the model from scratch and each group has its own weights. At some certain training epochs, *e.g.*, 80 and 100 epochs, we average the weights and then continue the training process. In the experimental results, "$-D$" represents the results of directly trained sharing grouped convolutional neural networks, "$-M$" represents the results of mean sharing, and "$-B$" represents the Bayesian sharing.

4.1 Implementation details and experimental settings

4.1.1 Training settings

On CIFAR-10 and CIFAR-100, we test all of these three methods. The initial learning rate is set as 0.1. For ResNeXt and ShuffleNet, the batch size is 128 and the learning rate is gradually divided by 10 at 81 and 122 epochs, with 164 training epochs in total. For G-DenseNet, the batch size is 64, and the learning rate is divided by 10 at 150 and 225 epochs, with a total of 300 training epochs. CIFAR-10 and CIFAR-100 are shorted as C-10 and C-100 in the result tables. Our optimizer uses momentum optimizer, with momentum 0.9 and weight decay 2×10^{-4}.

4.1.2 Evaluation Metrics

Parameter volume, model accuracy, and grouped convolution compression ratio (GCR) are considered as the evaluation metrics. Parameter volume, abbreviated as "#P", counts all the parameters in the model, including grouped convolutional layers and other linear or nonlinear layers. GCR is only for grouped convolutional layers, *i.e.*, volume of the sharing layer divided by the original volume before sharing. The compression ratio of the baseline model is also 100%. For a grouped convolutional layer with g groups, after sharing, the compression ratio is $1/g$. Therefore, our compression ratio relies on the number of groups. The number of floating point operations (FLOPs) and runtime are also attached.

4.2 Experiments on CIFAR Dataset

Our sharing method is applied to some baseline models, *i.e.*, ResNeXt, ShuffleNet and G-DenseNet to test CIFAR-10 and CIFAR-100, with some necessary model modifications in Tables 3 and 4. For ResNeXt-35 and RexNeXt-50, to test the cardinality, some tests are conducted on grouped convolutional layers with 4, 8, 16 groups, while the kernel size is 3×3. The point-wise convolutional layers are not considered here since they are not in the grouped convolutional layers of these two models. For ShuffleNet, grouped convolutional layers with 4 and 8 groups are tested. Different from ResNeXt, the point-wise (1×1) convolutions in ShuffleNet are grouped convolutional layers. Some experiments are conducted on ShuffleNet with 1×1 convolutions to further demonstrate the effectiveness of our sharing method. DenseNet contains both 3×3 and 1×1 convolutional layers, which are both tested to further validate the compatibility of our method.

As ablation studies, to clarify the impacts of our proposed Bayesian sharing framework, we compare the directly trained model with sharing grouped convolution, the mean sharing, and the proposed Bayesian sharing. The results are shown in Tables 3 and 4. For all of these tests, compared with the corresponding baseline models, the performance degradations occur in all directly trained models. The mean sharing method can achieve slight accuracy improvements in the most cases but G-DenseNet-86 since it is able to combine parameters among different groups without discrimination, *i.e.*, $\boldsymbol{\gamma} \equiv \mathbf{1}_g$ in Equation (15). Compared with the mean sharing method, our Bayesian sharing framework can bring significant accuracy improvements, mostly more than 2%, since it considers the intra-group correlation and the inter-group importance to combine parameters among different groups with discriminations. In other words, it is able to discriminately combine parameters to achieve message passing to different features according to the importances learned from maximum likelihood estimation in Equation (6). Some tests achieve higher improvements, *e.g.*, in Table 3, ResNeXt-50-B with 8 groups on CIFAR-100 improves the accuracy by $76.11\% - 73.16\% = 2.95\%$ with the less parameter volume. As a result, the proposed Bayesian sharing framework can improve the parameter efficiency, reduce the parameter redundancy and alleviate the overfitting issue.

Our Bayesian sharing method can result in impressive compression and runtime performance. Since for grouped convolutional layers with g groups, the GCR is $1/g$, more groups mean better compression ratio. According to Tables 3 and 4, as the group number increases, our method achieves higher compression ratios. Convolutional layers with 4 groups have the minimal GCR, *i.e.*, compressed to 0.25 times. Dividing to 16 groups can bring the maximal compression ratio, *i.e.*, 0.0625 times. Except for grouped convolutional layers, a typical neural network contains many other linear or non-linear layers. The models with more grouped convolutional layers have better compression performance for parameter volume by using our Bayesian sharing method. In Table 3, for ResNeXt models with the limited number of 3×3 convolutional layers, we can achieve up to 21% $((2.01 - 1.58)/2.01)$ overall volume reduction. In Table 4, for the sharing G-DenseNet-86, the parameter volume is reduced by 46.77% $((0.62 - 0.33)/0.62)$. The sharing ShuffleNet-1x reduces the parameter volume by 54.8% $((0.62 - 0.28)/0.62)$, and the parameter volume in ShuffleNet-2x reduces more than 64.17% $((1.34 - 0.48)/1.34)$. The proposed sharing method can achieve the more significant parameter reductions for CNN with the more grouped convolutional layers. Generally, the deeper and larger models suffer from higher risks of overfitting. With our Bayesian sharing framework, we can alleviate this problem by reducing parameter volume.

In particular, compared with these baseline methods, our proposed sharing grouped convolution does not reduce FLOPs in the inference stage. However, as shown in Figure 2(c), the parameters are shared among different groups. The sharing parameter strategy can reduce the actual number of memory accesses so that the inference time can be reduced, as shown in Tables 3 and 4. The runtime results are tested on one Kaggle Nvida Tesla P100 (16 GB memory, 720 GB/s bandwidth). It is believed that we can achieve better run performances on FPGA with dataflow optimizations[33].

5 Conclusion

In this paper, we propose a sharing grouped convolution structure with the Bayesian sharing framework to efficiently eliminate parameter redundancy and boost model performance. Intra-group correlation and inter-group importance are introduced into the prior of the parameters. We handle the Maximum Type II likelihood estimation problem of

Table 3: ResNeXt on CIFAR Dataset.

		RexNeXt-35					RexNeXt-50					
Model	g	#P	Acc. (%)		FLOPs	Time	#P	Acc. (%)		FLOPs	Time	GCR
		(M)	C-10	C-100	(M)	(ms)	(M)	C-10	C-100	(M)	(ms)	(%)
ResNeXt (baseline)	4	1.29	92.87	72.91	202	33.4	2.01	93.67	73.08	279	42.8	100.00
ResNeXt-D	4	1.19 (↓)	92.02 (↓)	72.17 (↓)	202	26.2	1.58 (↓)	92.89 (↓)	72.86 (↓)	279	33.3	25.00
ResNeXt-M	4	1.19 (↓)	93.31 (↑)	73.44 (↑)	202	26.2	1.58 (↓)	93.90 (↑)	73.55 (↑)	279	33.3	25.00
ResNeXt-B	4	**1.19** (↓)	**94.15** (↑)	**74.56** (↑)	**202**	**26.2**	**1.58** (↓)	**94.93** (↑)	**75.46** (↑)	**279**	**33.3**	**25.00**
ResNeXt (baseline)	8	1.31	93.00	73.07	214	43.3	2.04	93.22	73.16	291	47.3	100.00
ResNeXt-D	8	1.18 (↓)	92.32 (↓)	72.51 (↓)	214	38.5	1.70 (↓)	93.14 (↓)	72.71 (↓)	291	42.3	25.00
ResNeXt-M	8	1.18 (↓)	93.42 (↑)	73.62 (↑)	214	38.5	1.70 (↓)	93.78 (↑)	73.93 (↑)	291	42.3	12.50
ResNeXt-B	8	**1.18** (↓)	**94.08** (↑)	**74.97** (↑)	**214**	**38.5**	**1.70** (↓)	**95.04** (↑)	**76.11** (↑)	**291**	**42.3**	**12.50**
ResNeXt (baseline)	16	1.35	92.93 (↑)	73.23 (↑)	222	48.7	2.12	93.23 (↑)	73.31 (↑)	309	55.0	100.00
ResNeXt-D	16	1.26 (↓)	92.48 (↓)	72.57 (↓)	222	43.8	1.88 (↓)	93.22 (↓)	72.93 (↓)	309	52.6	6.25
ResNeXt-M	16	1.26 (↓)	93.38 (↑)	73.64 (↑)	222	43.8	1.88 (↓)	93.78 (↑)	73.79 (↑)	309	52.6	6.25
ResNeXt-B	16	**1.26** (↓)	**94.77** (↑)	**75.88** (↑)	**222**	**43.8**	**1.88** (↓)	**95.17** (↑)	**75.87** (↑)	**309**	**52.6**	**6.25**

Table 4: ShuffleNet and G-DenseNet on CIFAR Dataset.

Model	g	#P	Acc. (%)		FLOPs	Time	GCR
		(M)	C-10	C-100	(M)	(ms)	(%)
ShuffleNet-1x (baseline)	4	0.62	91.65	71.48	106	23.0	100.00
ShuffleNet-1x-D	4	0.28 (↓)	90.78 (↓)	70.56 (↓)	106	17.6	25.00
ShuffleNet-1x-M	4	0.28 (↓)	92.47 (↑)	72.29 (↑)	106	17.6	25.00
ShuffleNet-1x-B	4	**0.28** (↓)	**93.56** (↑)	**73.83** (↑)	**106**	**17.6**	**25.00**
ShuffleNet-2x (baseline)	4	1.34	91.48	71.65	123	29.8	100.00
ShuffleNet-2x-D	4	0.48 (↓)	90.32 (↓)	70.49 (↓)	123	22.9	25.00
ShuffleNet-2x-M	4	0.48 (↓)	92.68 (↑)	72.07 (↑)	123	22.9	25.00
ShuffleNet-2x-B	4	**0.48** (↓)	**93.79** (↑)	**73.89** (↑)	**123**	**22.9**	**25.00**
ShuffleNet-1x (baseline)	8	1.35	92.29	72.12	204	33.4	100.00
ShuffleNet-1x-D	8	0.60 (↓)	91.19 (↓)	71.57 (↓)	204	28.1	12.50
ShuffleNet-1x-M	8	0.60 (↓)	93.16 (↑)	72.26 (↑)	204	28.1	12.50
ShuffleNet-1x-B	8	**0.60** (↓)	**94.00** (↑)	**72.98** (↑)	**204**	**28.1**	**12.50**
G-DenseNet-86 (baseline)	4	0.62	93.21	73.89	102	69	100.00
G-DenseNet-86-D	4	0.33 (↓)	92.89 (↓)	73.14 (↓)	102	53	25.00
G-DenseNet-86-M	4	0.33 (↓)	93.78 (↑)	73.76 (↓)	102	53	25.00
G-DenseNet-86-B	4	**0.33** (↓)	**94.91** (↑)	**75.12** (↑)	**102**	**53**	**25.00**

the intra-group correlation and inter-group importance by a group LASSO type algorithm. Experiments demonstrate the proposed sharing grouped convolution structure with the Bayesian sharing framework can reduce parameters and improve prediction accuracy. The proposed sharing framework can reduce parameters up to 64.17%.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[2] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, “Pose transferrable person re-identification,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4099–4108.

[3] R. Chen, Y. Liu, M. Zhang, S. Liu, B. Yu, and Y.-W. Tai, “Dive deeper into box for object detection,” in *European Conference on Computer Vision (ECCV)*, 2020, pp. 412–428.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[5] H. Geng, H. Yang, L. Zhang, J. Miao, F. Yang, X. Zeng, and B. Yu, “Hotspot detection via attention-based deep layout metric learning,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2020, pp. 1–8.

[6] Q. Sun, A. A. Rao, X. Yao, B. Yu, and S. Hu, “Counteracting adversarial attacks in autonomous driving,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020, pp. 1–7.

[7] Q. Sun, T. Chen, J. Miao, and B. Yu, “Energy-driven DNN dataflow optimization on FPGA,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019.

[8] Q. Sun, T. Chen, S. Liu, J. Miao, J. Chen, H. Yu, and B. Yu, “Correlated multi-objective multi-fidelity optimization for hls directives design,” in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2021.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[10] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1492–1500.

[11] X. Wang, M. Kan, S. Shan, and X. Chen, “Fully learnable group convolution for acceleration of deep neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9049–9058.

[12] Y. Ma, R. Chen, W. Li, F. Shang, W. Yu, M. Cho, and B. Yu, “A unified approximation framework for compressing and accelerating deep neural networks,” in *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 376–383.

[13] C. Robert, *Machine learning, a probabilistic perspective*. Taylor & Francis, 2014.

[14] “The CIFAR-10 and CIFAR-100 datasets,” https://www.cs.toronto.edu/~kriz/cifar.html.

[15] Z. Zhang, J. Li, W. Shao, Z. Peng, R. Zhang, X. Wang, and P. Luo, “Differentiable learning-to-group channels via groupable convolutional neural networks,” in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3542–3551.

[16] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang, “Interleaved group convolutions,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4373–4382.

[17] Y. He, X. Zhang, and J. Sun, “Channel pruning for accelerating very deep neural networks,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[18] C. Louizos, K. Ullrich, and M. Welling, “Bayesian compression for deep learning,” in *Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 3288–3298.

[19] D. P. Kingma, T. Salimans, and M. Welling, “Variational dropout and the local reparameterization trick,” in *Conference on Neural Information Processing Systems (NIPS)*, 2015, pp. 2575–2583.

[20] J. Wang, H. Bai, J. Wu, and J. Cheng, "Bayesian automatic model compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 727–736, 2020.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2014.

[22] T. Chen, B. Lin, H. Geng, and B. Yu, "Smart building sensor drift calibration," in *Big Data Analytics for Cyber-Physical Systems*. Springer, 2020, pp. 187–202.

[23] T. Chen, B. Lin, H. Geng, S. Hu, and B. Yu, "Leveraging spatial correlation for sensor drift calibration in smart building," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 40, no. 7, pp. 1273–1286, 2021.

[24] T. Chen, B. Lin, H. Geng, and B. Yu, "Sensor drift calibration via spatial correlation model in smart building," in *ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[25] Z. Zhang and B. D. Rao, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," *IEEE Transactions on Signal Processing (TSP)*, vol. 61, no. 8, pp. 2009–2015, 2013.

[26] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[27] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[28] "Group LASSO Solver," https://github.com/fabianp/group_lasso.

[29] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3371–3412, 2011.

[30] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6848–6856.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[32] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger, "CondenseNet: An efficient densenet using learned group convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2752–2761.

[33] X. Wei, Y. Liang, and J. Cong, "Overcoming data transfer bottlenecks in FPGA-based DNN accelerators via layer conscious memory management," in *ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

Energy-Efficient, Reliable and QoS-Aware Task Mapping on Cyber-Physical Systems

Lei Mo[1], Angeliki Kritikakou[2], and Xinmei Li[1]
[1] School of Automation, Southeast University, Nanjing 210096, China.
E-mails: lmo@seu.edu.cn, lixinmei1999@foxmail.com
[2] University of Rennes, INRIA, IRISA, CNRS, Rennes 35042, France.
E-mail: angeliki.kritikakou@irisa.fr

Abstract

Cyber-Physical Systems (CPS) usually consist of a set of embedded systems (CPS nodes) connected through wireless communication, providing multiple functionalities that support different types of applications. During CPS deployment, application tasks are mapped on the CPS nodes with the objective of enhancing real-time performance, energy efficiency, and execution reliability. To satisfy these requirements, effective task mapping approaches should be designed based on different types of tasks, platforms, application and system requirements. In this paper, we provide a comprehensive survey regarding the task mapping methods in CPS.

1 Introduction

Embedded systems can support diverse functionalities on a tiny platform, e.g., data collection and processing, wireless communication. With the characteristics of simple structure, high degree of customization, low cost and low power consumption, embedded systems have became the critical part of networked systems, such as Cyber-Physical Systems (CPS). They are widely used in the wireless nodes, such as sensors, actuators or controllers. The system applications, including control, sensing, data processing and date transmission, contain multiple dependent tasks. These tasks can be mapped on the wireless nodes so as to achieve the desired system performance. Following the “Fog/Edge-computing” model, instead of collecting and sending all data to a remote Base Station (BS), a part of the data processing is done on the wireless nodes, and thus, only a small part of pre-processed data is sent to the BS. As a result, the use of system resources can be optimized.

During the task mapping process, i.e., task allocation and task scheduling, the constraints related to energy consumption and real-time execution should be taken into account. This is because, most of the wireless nodes have limited energy budget, especially for the energy-harvesting or battery-powered devices. In addition, real-time responsiveness is required by many applications, e.g., mobile target tracking, as missing task deadline can have serious consequences. With a proper task mapping scheme, the tasks can be executed in parallel on various nodes so as to improve the real-time execution. On this basis, by employing energy efficiency methods, such as Imprecise Computation (IC), Dynamic Voltage and Frequency Scaling (DVFS), and Dynamic Power Management (DPM), the time and the energy consumed to execute the tasks can be optimized. However, DVFS will influence the reliability of task execution. Usually, the higher is the frequency to execute the tasks, the higher is the task reliability.

Energy efficiency, task deadline and reliability are important, but these are conflicting objectives, since enhancing real-time execution and task reliability often require to consume more energy. Several methods have been proposed to balance these requirements. In the rest of the paper, we will first provide the main preliminaries for system model used in task mapping, and then, give a comprehensive survey of the state-of-the-art.

2 System Model

In this section, we describe the typical characteristics of the system model used for task mapping. Table 5 links the task and platform model with the corresponding variables used in the task mapping methods.

Table 5: Link of main task and platform characteristics with task mapping variables.

Variables	*Task model*		*Platform model*					
	IC	Dependent	Multi-core	DVFS	DPM	Reliability	Migration	Network
Task frequency				√	√	√		
Task allocation			√					√
Task sequence		√						
Task start time					√			
Task adjustment	√							
Task duplication						√		
Task partition							√	
Node communication								√

2.1 Task Model

Tasks can be modeled as imprecise computation (IC) tasks and precise computation tasks, based on their characteristics [1]. In the IC model, a task can be logically divided into: 1) a *mandatory* subtask, which guarantees the basic QoS, and 2) an *optional* subtask, which further improves QoS. Both the mandatory subtask and the optional subtask must be completed before the task's deadline to generate a correct and in-time result, and the optional subtask is executed after the mandatory subtask. By executing the mandatory subtask, we obtain the basic Quality of Service (QoS). When the system resources are available, the optional subtask can be executed. The longer the optional subtask is executed, the better is the QoS of the result. Compared with the IC tasks, the precise computation tasks can be considered as a special case of IC tasks, where the complete task corresponds to the mandatory subtasks, while the optional subtask is empty.

The characteristics of an IC task τ_i can be described by a tuple $\{o_i, M_i, O_i, t_i^s, D_i, T_i\}$ [2], where M_i is the mandatory subtask, o_i is the optional subtask and it has an upper bound O_i, i.e., $0 \leq o_i \leq O_i$. M_i and O_i are measured in Worst Case Execution Cycles (WCECs). t_i^s, D_i and T_i are the start time, the deadline and the period of task τ_i, respectively. During the task mapping process, o_i and t_i^s are the optimization variables since they influence task scheduling decision. A real-time application is usually consisting of a set of N dependent tasks $\{\tau_1, \ldots, \tau_i, \ldots, \tau_N\}$. They can be described by a Directed Acyclic Graph (DAG) $G(\mathscr{V}, \mathscr{E})$, where vertexes $\mathscr{V}$ represent the set of tasks to be executed, while edges $\mathscr{E}$ represent the data dependencies between the tasks. The dependency between the tasks can be further described by an $N \times N$ binary matrix $\boldsymbol{S}$, where $s_{ij} = 1$ represents task τ_i precedes task τ_j, i.e., τ_j starts after the end time of τ_i, otherwise, $s_{ij} = 0$. Since the tasks are dependent, the adjustment of task start time t_i^s and t_j^s is restricted by the task sequence s_{ij}.

Based on the IC task, the QoS function provides the obtained QoS based on the number of execution cycles of the optional subtask. Usually, it can be formulated as: 1) *Linear* function, e.g., $\sum_{i=1}^{N}(a_i o_i + b_i)$ [3]; or 2) *Concave* function, e.g., $\sum_{i=1}^{N}(\alpha_i o_i + \beta_i \sqrt{o_i} + \sqrt{3}\gamma_i o_i)$ [4]. The linear function models the case where the system QoS increases uniformly during the optional subtask execution, while the concave function addresses the case where the increase of QoS exhibits a continuously nondecreasing rate as the optional subtask execution goes on. Linear and general concave functions are considered as the most realistic and typical QoS representation in the literature [5], since they adequately capture the behavior of many application areas, such as image and speech processing, control engineering, and automatic target recognition.

2.2 Platform Model

With the increasing requirements of high performance computation, low energy consumption and low task execution delay, *single-core* embedded systems are insufficient for data intensive applications, such as multimedia. High performance computation and low task execution delay usually require more energy consumption. To balance these *contradictory* requirements, *multi-core* embedded systems are used. Multi-core platforms allow the computations to be split and assigned to multiple processors, and each processor can run at a lower voltage and frequency. Compared to a single-core system, this result has a higher energy and time efficiency. For example, the fire detection wireless node MiLive-v2 [6] is equipped with three processor types: 1) a 8-bit low-power AVR processor (ATmega128rfa1), which can be used to run simple tasks; 2) a 32-bit powerful ARM processor (ARM1176JZF), which can be used to run more complicated tasks; and 3) a Digital Signal Processor (DSP) unit, which can be specially used for image processing.

Dynamic Voltage and Frequency Scaling (DVFS) and *Dynamic Power Management* (DPM) are two effective methods to improve energy efficiency of task execution [7]. As long as the resource and application constraints (e.g., task deadline) allow, the methods with DVFS/DPM achieve significant energy reductions. DVFS is able to adjust the supply voltage/frequency of the processor during the task execution process, and thus, the time and the energy required to execute the tasks can be optimized. The power consumption of a processor θ_k is expressed as $P_k^c = P_k^s + P_k^d$ [7], where $P_k^s = C_k^s v_k^{\rho_k}$ is the *static* power of the processor ready to execute (being either on the active or idle mode), $P_k^d = C_k^d f_k v_k^2$ is the *dynamic* power of task execution. C_k^s, ρ_k and C_k^d are constants depending on the type of processor. By lowering the supply voltage/frequency (v_k, f_k), quadratic savings in energy consumption can be achieved. Based on the adjustment manner, DVFS can be classified as: 1) continuous DVFS [4]: the voltage can be changed within the range $[V_{\min}, V_{\max}]$; and 2) discrete DVFS [2]: the processor can select L different voltage/frequency levels $\{(v_1, f_1), \ldots, (v_l, f_l), \ldots, (v_L, f_L)\}$. On the other hand, based on the length of duration, DVFS can be classified as: 1) inter-task DVFS [2]: the voltage/frequency of processor stays constant during the execution of a task; 2) intra-task DVFS [8]: the voltage/frequency of processor can be changed during the execution of a task.

When the assigned tasks are finished, the processor will switch from the *active* mode to the *idle* mode. In addition, when the idle interval of the processor is longer than a certain threshold T_{th} (called break-even time), the processor will turn into the *sleep* mode. The transition time and energy overhead is very small compared to the time and energy required to complete a task. Such overheads are typically incorporated into the execution time and energy of the task [9]. According to the start time t_i^s and the end time t_i^e of each task τ_i, i.e., to adjust the idle interval, the processor can directly switch from the active model to the sleep mode through the DPM. Since sleep mode consumes less energy than idle mode, DPM can further reduce the energy consumption of task execution.

Although the energy efficiency of task execution can be enhanced through the DVFS, DVFS has a negative impact on reliability, mainly due to the increased transient fault rates at low supply voltage/frequency levels. Usually, the reliability follows a Poisson distribution model [10]. When a processor θ_k uses voltage/frequency level (v_k, f_k) to execute a task τ_i with C_i cycles, the reliability of task execution is

$$R_{ik} = e^{-\lambda \times 10^{\frac{d(f_{\max} - f_k)}{f_{\max} - f_{\min}}} \times \frac{C_i}{f_k}}, \tag{16}$$

where $f_{\max} = \max\{f_1, \ldots, f_L\}$ and $f_{\min} = \min\{f_1, \ldots, f_L\}$, λ and d are the constants related to fault rate and sensitivity. Eq. (16) shows that the higher frequency used to execute the tasks, the higher reliability can be obtained. To improve task reliability, besides DVFS, task replication can be also used. For instance, by applying selective task duplication, task τ_i is duplicated when its execution reliability is lower than a given threshold R_{th}. Then, task τ_i and its duplicated task are executed on a different processor, since it is unlikely that the execution of both original and duplicated tasks on different processors fails [11]. Therefore, if the reliabilities of original and duplicated tasks are R_{im} and R_{in}, respectively, the total reliability of task τ_i becomes $R_i = 1 - [1 - R_{im}][1 - R_{in}]$.

Based on the processor's characteristics, multi-core embedded systems can be divided into: 1) *homogeneous* platform, and 2) *heterogeneous* platform. For the homogeneous platform, the processors are the same, and thus, they have the same frequency characteristics (e.g., minimal, maximal and operating frequencies). However, for the heterogeneous platform, the processors are divided into several clusters, where each cluster consists of a set of symmetric processors that have the same frequency characteristics. Since the processors of heterogeneous platform have different voltage/frequency levels, a task execution efficiency $\lambda_{ik} \in (0, 1]$ is usually introduced to describe the task execution efficiency (i.e., the heterogeneity) [9]. Correspondingly, the Worst Case Execution Time (WCET) of task τ_i, when it is executed on processor θ_k, is calculated as $\frac{C_i}{f_k \lambda_{ik}}$. In addition, some heterogeneous platforms, e.g., ARM big.LITTLE platform [12, 13], can support task migration, which means a task can migrate from one cluster (e.g., big cluster) to another cluster (e.g., LITTLE cluster) during task execution, and thus, the efficiency and the schedulability of task execution can be further enhanced. The methods mentioned above and the corresponding optimization variables are summarized in Table 5.

For the networked systems, since the nodes are connected with each other wirelessly, when dependent tasks are assigned to different nodes for execution, the nodes will spend time and energy for data communication [14]. Since the communication range of each node is limited, the task allocation decision will influence the communication cost of the nodes. Hence, we also need to optimize task-to-node allocation. As the task is time sensitive and the energy budget of the node is limited, this impact should be formulated. To achieve that, we can introduce an energy matrix $\boldsymbol{E} = [e_{\beta\gamma k}]_{M\times M\times M}$ and a time matrix $\boldsymbol{T} = [t_{\beta\gamma}]_{M\times M}$, where M is the number of the nodes. $e_{\beta\gamma k}$ is the energy consumed by a node θ_k when relaying unit of data from node θ_β to node θ_γ, and $t_{\beta\gamma}$ is the time required to transmit unit of data from node θ_β to node θ_γ. Therefore, based on the matrices $\boldsymbol{E}$ and $\boldsymbol{T}$, we obtain the corresponding communication cost under the given task allocation decision [15].

3 Task Mapping Methods

The basic task mapping contains two steps: 1) task allocation: determines on which processor/node should the task be executed, and 2) task scheduling: determines when a task starts and ends its execution. Table 6 classifies the relevant state-of-the-art methods, presented in this section, based on 1) the task model (Imprecise, Precise), 2) the target platform (Embedded, Networked), 3) the constraints (Energy, Real-Time, Reliability), 4) the objective (Minimize Energy, Balance Energy, Maximize Reliability, Maximize QoS), and 5) the achieved solutions (Heuristic, Optimal) of task mapping problem under study.

Table 6: Task Mapping Methods

Ref.	Task		Platform		Constraints			Objective				Solution	
	Imprecise	Precise	Embedded	Networked	Energy	Real-time	Reliability	MinE.	BalE.	MaxR.	MaxQoS	H.	O.
[16]		√	√			√		√				√	
[17]		√	√			√		√				√	
[9]		√	√			√		√				√	
[7]		√	√			√		√				√	
[18]		√	√			√		√				√	
[19]		√	√			√		√				√	
[20]		√	√		√	√				√		√	
[21]		√	√		√	√				√		√	√
[22]		√	√		√	√				√		√	
[23]		√	√			√	√	√				√	
[10]		√	√			√	√	√				√	
[11]		√	√			√	√	√				√	
[24]		√	√			√	√	√					√
[4]	√		√		√	√					√	√	√
[25]	√		√		√	√					√	√	
[26]	√		√		√	√					√	√	
[5]	√		√		√	√					√		√
[3]	√		√		√	√					√	√	
[27]	√		√		√	√					√	√	
[28]	√		√		√	√					√	√	√
[29]		√		√	√	√			√			√	
[30]		√		√	√	√			√				√
[31]		√		√	√	√		√				√	
[32]		√		√	√	√		√					√
[33]		√		√	√	√		√					√
[14]		√		√	√	√		√	√			√	√

3.1 Task Mapping on Embedded Systems

Existing task mapping methods can be classified as *Energy-aware* task mapping and *QoS-aware* task mapping.

3.1.1 Energy-Aware Task Mapping

Energy-aware task mapping problem usually considers the precise computation task model and the aim to minimize the energy consumption under energy, real-time and reliability constraints.

When the voltage/frequency level is *discrete*, the corresponding task mapping problem is usually formulated as Integer Programming (IP), e.g., [16, 17, 9]. To efficiently solve the task mapping problem, a hybrid Genetic Algorithm (GA) is proposed in [16], a polynomial-time two-step heuristic is designed in [17], and the IP problem is relaxed to a Linear Programming (LP) in [9]. Combining DVFS and DPM, a Mixed-Integer Linear Programming (MILP)-based task mapping problem is considered in [7] and the problem is solved by CPLEX solver.

When the voltage/frequency level is *continuous*, a convex task mapping problem is proposed in [18] and the problem can be solved by using polynomial-time methods. In [19], Mixed-Integer Non-Linear Programming (MINLP) is used to formulate the task mapping problem. The problem is relaxed to an MILP by linear approximation and is solved by Branch and Bound (B&B) method.

If multiple system requirements are taken into account, the complex coupling between the optimization variables makes the problem difficult to solve, especially when the coupling is non-linear and non-convex. The common methods to deal with the nonlinear items include: 1) linear approximation [19], and 2) variables replacement [7].

Taking the task reliability into account, existing methods include *reliability-optimized* task mapping [20, 21, 22] and *energy-optimized* task mapping [23, 10, 11]. The aim of reliability-optimized task mapping is to maximize task reliability under system resource and application constraints. Regarding energy-optimized task mapping, the aim is to minimize energy consumption, under energy supply, task reliability and real-time constraints.

To maximize the reliability of task execution, as well as to meet energy supply, task dependency and task deadline constraints, the dynamic and static methods are proposed in [20] and [21] to allocate and schedule dependent tasks on the multi-core platforms. The multi-objective task mapping problem is consider in [22], where the aim is to simultaneously maximize the reliability and the lifetime of tasks.

DVFS is applied in [23] to meet task reliability constraint. Since task duplication is not taken into account, higher voltage/frequency level may require to execute the tasks. In [10], full replication is used to meet reliability constraint, and thus, each task is replicated once at least. Although more tasks being duplicated, higher reliability is achieved while task redundancy is incurred, more energy and time are required to execute the tasks. DVFS and task duplication are combined in [11] and [24], where the only partial tasks are duplicated.

3.1.2 QoS-Aware Task Mapping

Existing works consider the QoS-aware task mapping problem using the IC task model and having a goal to maximize the QoS under a set of realtime and/or energy supply constraints.

The target platforms studied in [4] and [25] are single-core platforms. Therefore, there is no need to consider task allocation decision. Although some works target at multi-core platforms, e.g., [26, 5, 3, 27, 28], they focus on different contexts. For example, the task-to-processor allocation is fixed and given in advance for all the tasks in [26], each processor has a predefined frequency in [5, 3], the tasks are independent in [28], and the multi-objective task mapping is consider in [27] with the aim is to maximize the QoS as well as to minimize the energy consumption. For tractability reasons, the variable, optional subtask adjustment, is usually considered as continuous variable in the above studies. When task mapping problem is solved, the result is rounded down. As the tasks execute typically hundreds of thousands of cycles, this impact is negligible [4].

The QoS-aware task mapping problem is a well-known NP-hard problem. Hence, finding an optimal solution satisfying all the given constraints (e.g., energy efficiency, deadline, QoS, task dependency, and DVFS) is very difficult and time consuming. The methods that used to solve the aforementioned problems can be classified into two main classes. The first class includes the methods based on heuristics, e.g., [25, 26, 3, 27]. The second class includes the methods that always produce an optimal solution, e.g., [4, 5, 28].

The heuristic methods mentioned above usually adopted a multi-step optimization, i.e., to decouple the variables and to determine their values in sequence. For instance, a two-step heuristic is proposed in [3]. The aim of the first step is to find a proper task-to-processor allocation, such that the energy consumption is minimized. With the given task allocation decision, the energy consumed to execute one task is proportional to the length of its optional subtask.

Based on the given task allocation decision, the aim of the second step is to adjust the optional subtasks so as to maximize OoS under the energy supply constraint. Although the heuristic methods are able to find feasible solutions in a short amount of time, they do not provide the bounds on the solution quality. In addition, they are sensitive to the problem structure, i.e., when new assumptions or constraints are taken into account they must be redeveloped.

On the other hand, to find the optimal solution, the common methods include: 1) convex optimization [4, 5], and 2) Benders Decomposition (BD) [28]. Instead of solving the binary and the continuous variables of MILP problem simultaneously, BD technique decomposes the original problem into two smaller problems with less variables and constraints: an ILP-based *Master* Problem (MP) and a LP-based *Slave* Problem (SP). Then, it solves the subproblems by utilizing the solution of one in the other. By doing so, the computation time can be significantly reduced. In each iteration, the current MP is solved to determine a lower bound for the original problem along with the temporary values of the binary variables. And then the SP is solved to obtain an upper bound by utilizing the solution of MP. The bounds are updated if the stopping criterion, i.e., the gap between the upper and lower bounds is smaller than a predefined threshold, is not met, and a new constraint (i.e., *Benders cut*) is generated by using the solution of SP and is added to MP in next iteration. As the BD method runs in an iterative way, the stopping criteria can serve as the controllable parameters to trade-off the quality of the solution (i.e., system QoS) and the computational complexity (i.e., computing time). In addition, as the computational complexity of BD method is dominated by the cost of solving the ILP-based MP, an accelerated BD method is proposed in [2], without violating optimality of the solution. This method replaces the optimal solution of MP with the feasible solution and uses it for the iteration between the MP and the SP. The structure of BD algorithm is shown in Fig. 1.

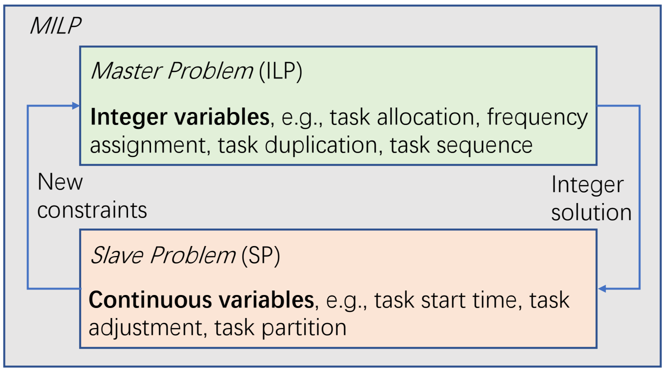

MILP
Master Problem (ILP)
Integer variables, e.g., task allocation, frequency assignment, task duplication, task sequence
New constraints
Integer solution
Slave Problem (SP)
Continuous variables, e.g., task start time, task adjustment, task partition

Figure 1: The structure of BD algorithm.

3.2 Task Mapping on Networked Systems

In the networked systems, there are various approaches to map dependent tasks on the wireless nodes, e.g., [29, 30, 31, 32, 33, 14]. In [29], the objective of task mapping is to minimize the energy consumption of the nodes with low energy level, and thus, the system lifetime can be enhanced. To solve this task mapping problem a multi-step heuristic method is designed. Similar task mapping problem is studied in [30], while a game theory approach is proposed to find the solution. However, the voltage/frequency levels of the processors are fixed in these approaches. The problems of task mapping and DVFS are jointly addressed in [31] and [32], based on the idea of problem decomposition, a heuristic method [31] and an optimal method [32] are presented to solve complex optimization problems. In [33], by using evolutionary algorithms (ant colony and bee colony) to perform task mapping, the energy consumption of the nodes for data communication and task execution can be minimized. The above methods assume that one node transmits data to another node through a fixed path. Since the wireless nodes are connected with each other through a mesh network, the communication between the nodes can be performed through multiple routing paths. The multi-path data routing is considered in [14], where the aim to enhance system lifetime, i.e., to balance the energy consumption of the nodes. The task mapping problem is first formulated as an Integer Non-Linear Programming (INLP) and then is solved by greedy algorithm.

4 Conclusion

This survey provides a current view of task mapping problem in CPS. We follow a three-step approach to summarize the recently published papers in the relevant area. More precisely, a task classification is obtained regarding he characteristics of task models, including the task dependency and the adjustment of execution cycles. A platform classification is proposed based on the platform type and the functions of the processor/node. Finally, the task mapping methods are classified, based on task and system under study, using task and system classifications.

References

[1] J. W. S. Liu, W. K. Shih, K. J. Lin, R. Bettati, and J. Y. Chung, "Imprecise computations," *Proc. IEEE*, vol. 82, no. 1, pp. 83–94, 1994.

[2] L. Mo, A. Kritikakou, and O. Sentieys, "Controllable QoS for imprecise computation tasks on DVFS multicores with time and energy constraints," *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 8, no. 4, pp. 708–721, 2018.

[3] J. Zhou, J. Yan, T. Wei, M. Chen, and X. S. Hu, "Energy-adaptive scheduling of imprecise computation tasks for QoS optimization in real-time MPSoC systems," in *Proc. IEEE DATE*, 2017, pp. 1402–1407.

[4] L. A. Cortes, P. Eles, and Z. Peng, "Quasi-static assignment of voltages and optional cycles in imprecise-computation systems with energy considerations," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 10, pp. 1117–1129, 2006.

[5] H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez, "Optimal reward-based scheduling for periodic real-time tasks," *IEEE Trans. Comput.*, vol. 50, no. 2, pp. 111–130, 2001.

[6] X. Liu, H. Zhou, J. Xiang, S. Xiong, K. M. Hou, C. de Vaulx, H. Wang, T. Shen, and Q. Wang, "Energy and delay optimization of heterogeneous multicore wireless multimedia sensor nodes by adaptive genetic-simulated annealing algorithm," *Wirel. Commun. Mob. Comput.*, vol. 2018, pp. 1–13, 2018.

[7] G. Chen, K. Huang, and A. Knoll, "Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 3, pp. 1–21, 2014.

[8] K. Huang, K. Wang, D. Zheng, X. Jiang, X. Zhang, R. Yan, and X. Yan, "Expected energy optimization for real-time multiprocessor SoCs running periodic tasks with uncertain execution time," *IEEE Transactions on Sustainable Computing*, pp. 1–1, 2018.

[9] D. Li and J. Wu, "Minimizing energy consumption for frame-based tasks on heterogeneous multiprocessor platforms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 810–823, 2015.

[10] M. A. Haque, H. Aydin, and D. Zhu, "On reliability management of energy-aware real-time systems through task replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 813–825, 2017.

[11] C. Gou, A. Benoit, M. Chen, L. Marchal, and T. Wei, "Reliability-aware energy optimization for throughput-constrained applications on MPSoCs," in *International Conference on Parallel and Distributed Systems*, 2018, pp. 1–10.

[12] H. S. Chwa, J. Seo, H. Yoo, J. Lee, and I. Shin, "Energy and feasibility optimal scheduling on big.LITTLE platforms," in *Proc. IEEE RTSOPS*, 2013, pp. 770–777.

[13] L. Mo, A. Kritikakou, and O. Sentieys, "Approximation-aware task deployment on asymmetric multicore processors," in *Proc. ACM/IEEE DATE*, 2019, pp. 1513–1518.

[14] A. Pathak and V. K. Prasanna, "Energy-efficient task mapping for data-driven sensor network macroprogramming," *IEEE Trans. Comput.*, vol. 59, no. 7, pp. 955–968, 2010.

[15] L. Mo, A. Kritikakou, O. Sentieys, and X. Cao, "Real-time imprecise computation tasks mapping for DVFS-enabled networked systems," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8246–8258, 2021.

[16] A. Mahmood, S. A. Khan, F. Albalooshi, and N. Awwad, "Energy-aware real-time task scheduling in multiprocessor systems using a hybrid genetic algorithm," *Electron.*, vol. 6, no. 2, 2017.

[17] H. Xu, F. Kong, and Q. Deng, "Energy minimizing for parallel real-time tasks based on level-packing," in *Proc. IEEE RTCSA*, 2012, pp. 98–103.

[18] F. Kong, W. Yi, and Q. Deng, "Energy-efficient scheduling of real-time tasks on cluster-based multicores," in *Proc. ACM/IEEE DATE*, 2011, pp. 1–6.

[19] L. F. Leung, C. Y. Tsui, and W. H. Ki, "Simultaneous task allocation, scheduling and voltage assignment for multiple-processors-core systems using mixed integer nonlinear programming," in *Proc. IEEE ISCAS*, 2003, pp. 309–312.

[20] Y. Ma, T. Chantem, R. P. Dick, S. Wang, and X. S. Hu, "An on-line framework for improving reliability of real-time systems on "big-LITTLE" type MPSoCs," in *Proc. ACM/IEEE DATE*, 2017, pp. 446–451.

[21] B. Zhao, H. Aydin, and D. Zhu, "On maximizing reliability of real-time embedded applications under hard energy constraint," *IEEE Trans. Ind. Informat.*, vol. 6, no. 3, pp. 316–328, 2010.

[22] J. Zhou, J. Sun, X. Zhou, T. Wei, M. Chen, S. Hu, and X. S. Hu, "Resource management for improving soft-error and lifetime reliability of real-time MPSoCs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 12, pp. 2215–2228, 2019.

[23] G. Xie, Y. Chen, Y. Liu, Y. Wei, R. Li, and K. Li, "Resource consumption cost minimization of reliable parallel applications on heterogeneous embedded systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1629–1640, 2017.

[24] M. Cui, A. Kritikakou, L. Mo, and E. Casseau, "Fault-tolerant mapping of real-time parallel applications under multiple DVFS schemes," in *Proc. IEEE RTAS*, 2021, pp. 387–399.

[25] H. Yu, B. Veeravalli, and Y. Ha, "Dynamic scheduling of imprecise-computation tasks in maximizing QoS under energy constraints for embedded systems," in *Proc. IEEE ASP-DAC*, 2008, pp. 452–455.

[26] H. Yu, B. Veeravalli, Y. Ha, and S. Luo, "Dynamic scheduling of imprecise-computation tasks on real-time embedded multiprocessors," in *Proc. IEEE CSE*, 2013, pp. 770–777.

[27] M. Isabel, O. Javier, S. Rodrigo, and Z. Paula, "Energy-aware scheduling mandatory/optional tasks in multicore real-time systems," *Intl. Trans. in Op. Res.*, vol. 24, no. 12, pp. 173–198, 2017.

[28] L. Mo, A. Kritikakou, and O. Sentieys, "Energy-quality-time optimized task mapping on DVFS-enabled multicores," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2428–2439, 2018.

[29] W. Li, F. C. Delicato, P. F. Pires, Y. C. Lee, A. Y. Zomaya, C. Miceli, and L. Pirmez, "Efficient allocation of resources in multiple heterogeneous wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1775–1788, 2014.

[30] N. Edalat, C. Tham, and W. Xiao, "An auction-based strategy for distributed task allocation in wireless sensor networks," *Computer Communications*, vol. 35, no. 8, pp. 916–928, 2012.

[31] Y. Tian and E. Ekici, "Cross-layer collaborative in-network processing in multihop wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 3, pp. 297–310, 2007.

[32] L. Mo and A. Kritikakou, "Mapping imprecise computation tasks on cyber-physical systems," *Peer-to-Peer Networking and Applications*, vol. 12, no. 6, pp. 1726–1740, 2019.

[33] W. Zhang, B. Song, and E. Bai, "A trusted real-time scheduling model for wireless sensor networks," *Journal of Sensors*, vol. 2016, pp. 1–8, 2016.

Cooperative 3D SLAM in Distributed Way

Yuting Xie and Long Chen

School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China

1 Introduction

The simultaneous localization and mapping (SLAM) is referred to as the ability of robots to extract information from surroundings to build maps and simultaneously utilize the map for self-localization [1]. The SLAM problem has continued to draw considerable attention in the robotic community due to its fundamental importance in most of robotic tasks. Numerous effective solutions for SLAM deployed on a single robot have been proposed [2, 3, 4, 5]. However, single robot systems have limits on resources and efficiency. Moreover, many complex tasks cannot be completed by one single robot. Thus, multi-robot system has become an emerging research hot spot in robotics, so as the SLAM problem.

In cooperative SLAM, each robot in the robot team explores part of the entire environment, which contains overlapped areas. These overlapped areas could be used to establish a consistent coordinate system between robots. These individual maps established by each robot could be merged to produce a complete global map for the explored unknown environment(Fig. 1).

Generally, an cooperative SLAM system can be centralized or distributed [6]. In a centralized system, a predefined central node gathers all collected data and performs tasks, which is hard to deployed in lots of scenarios, such as ruins, subterranean and other large scale wild scenarios, due to the high requirements on network and heavy computational burden on the central node [7, 8, 9, 10]. While in a distributed system, the computational load is divided among robots and the communication load is greatly reduced, which makes it more flexible and applicable. The core problem of cooperative SLAM in distributed way is how to maintain accuracy while reducing the amount of data transmission. Several works have explored distributed SLAM [11][12], but few employed 3D LiDAR.

2 RDC-SLAM: Cooperative 3D SLAM in Distributed Way

In this paper, we developed a complete real-time distributed cooperative SLAM system, called RDC-SLAM. The system performs in a distributed manner where the computation load is shared among robots. Each robot performs procedures described in Fig. 2. When a new laser scan taken by 3D LiDAR arrives, it is fed to the LiDAR odometry and the place recognition module. The PR module extracts a compact description [13] of the laser scan and with occasional communication, produces candidate matches between laser scans from different robots. The LiDAR odometry module computes sequential constraints and loop closure constraints to generate a map for localization. The relative pose module extracts information from the parts of the map at candidate matches to further refine the transformation between corresponding scans or to reject candidate matches. The distributed graph optimization module receives initial guesses from the map, inter-data associations from the relative pose module and intra-data associations from the LiDAR odometry module to update the map by consistently estimating multi-trajectories through communication [14]. This system works continuously as new LiDAR data is acquired.

Robots share information during the encounters (which means that they are in the communication range). To meet system requirements, modules involved in communication should ensure the system's performance with low data transmission.

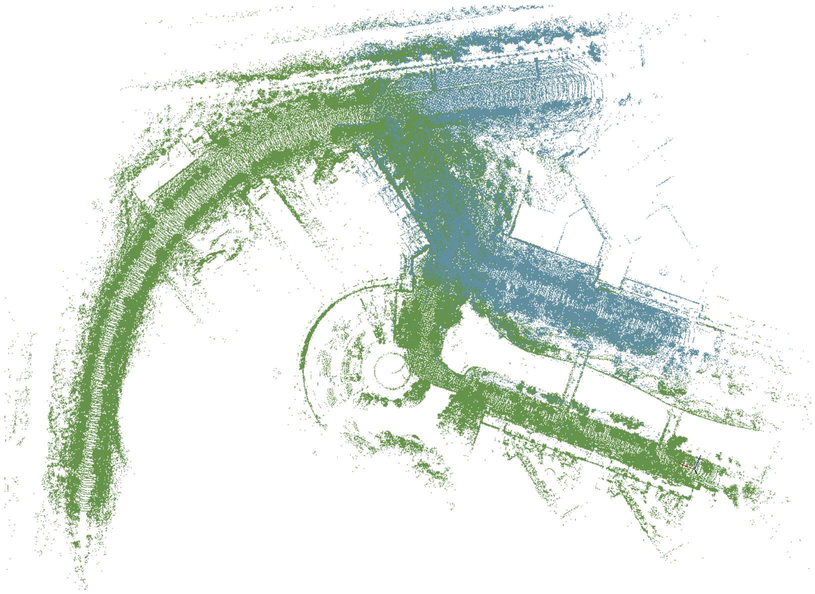


Figure 1: A demonstration for final map of cooperative SLAM. Point cloud in different colors represents data collected from different robots.

Table 7: Impact of network connection distance.

Connection distance(m)	ATE(m)	Total transmitted(KB)
Global coverage	0.0612	372
200	0.0604	323
100	0.0618	276
50	0.0876	53
None connection	0.0676	0

To prove the proposed cooperative SLAM can adopt to short-distance, short-term communication conditions and achieve comparable performance, we have studied how the distance of the network connection affects the accuracy and data transmission amount, which is illustrated in Table 7. The data is analyzed on KITTI [15] sequence 0. It can be seen that the communication distance has little effect on accuracy. To some extent, short distance can even reduce the amount of data transmission.

3 Conclusions

This paper proposes a real-time distributed cooperative SLAM system based on 3D LiDAR called RDC-SLAM. The system is built in distributed manner and proposed with elaborate communication rules to integrate state-of-art components. The front end is responsible for acquisition of intra data associations and inter data associations, which are fed to the back end, while the back end of RDC-SLAM is based on a distributed graph optimization algorithm and each robot maintains only states related to itself. This work not only reduce the pressure of communication bandwidth and time consumption between multiple robots, but also maintain comparable accuracy.

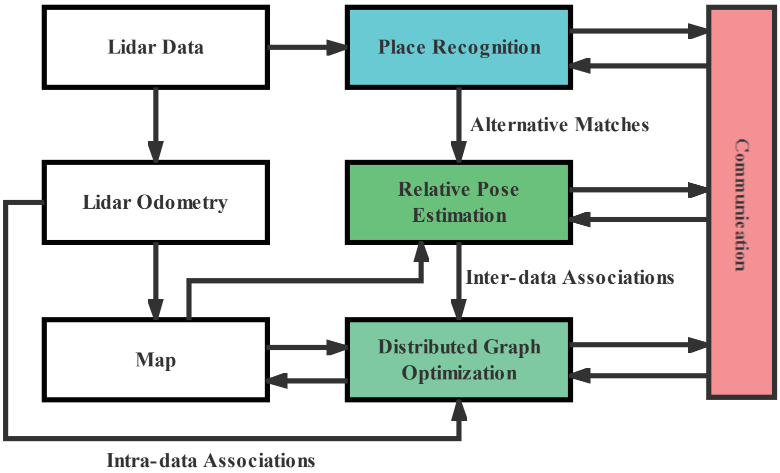


Figure 2: The framework for the proposed RDC-SLAM approach. The system performs in a distributed manner. For each individual, when the new LiDAR data comes in, two threads respectively calculate local odometry (by LiDAR odometry module) and positional relationship with neighbors (by place recognition module and relative pose estimation module). Finally, the distributed graph optimization module integrates the inter-data associations and intra-data associations to maintain the local map. Notablely, the consistency of multiple local maps are coordinated through communication.

4 Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1305002 and in part by the National Natural Science Foundation of China under Grant 61773414.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[3] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[6] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[7] L. Riazuelo, J. Civera, and J. M. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401–413, 2014.

[8] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5266–5272.

[9] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Mal, “A robust pose graph approach for city scale lidar mapping,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1175–1182.

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,” *IEEE Transactions on Robotics*, 2021.

[11] M. T. Lazaro, L. M. Paz, P. Pinies, J. A. Castellanos, and G. Grisetti, “Multi-robot slam using condensed measurements,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1069–1076.

[12] T. Cieslewski, S. Choudhary, and D. Scaramuzza, “Data-efficient decentralized visual slam,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2466–2473.

[13] K. P. Cop, P. V. Borges, and R. Dubé, “Delight: An efficient descriptor for global localisation using lidar intensities,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3653–3660.

[14] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, “Distributed trajectory estimation with privacy and communication constraints: a two-stage distributed gauss-seidel approach,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5261–5268.

[15] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

Toward Efficient Distributed Combinatorial Optimization

Mikhail A. Bragin, University of Connecticut; Bing Yan, Rochester Institute of Technology

Abstract

The emergence of cyber components such as sensors as well as communication and computation devices leads to the interconnectivity of "things" and offers decision support within physical systems such as factories and power systems, thereby providing the infrastructure necessary to support a transition from centralized system planning and operation to decentralized ones. The difficulty along the way of enabling this transition is the combinatorial complexity of the associated discrete optimization problems such as manufacturing scheduling and unit commitment. This newsletter provides an overview of the recent methodological and modeling advancements to enable efficient asynchronous coordination of distributed subsystems, which comprise the corresponding physical systems, supported by cyber components such as sensors, communication devices, and distributed processors.

1 Introduction: Combinatorial Optimization and Cyber-Physical Systems

Combinatorial mathematical optimization is pervasive in many fields and plays a prominent role in problems of importance to the society, such as 1) clean energy smart manufacturing systems for improvements of on-time deliveries and for reduction of energy consumption and 2) smart grids for efficient and reliable provision of electricity at a reduced cost while considering uncertainties due to intermittent renewables and contingencies. The associated decision-making optimization problems involve discrete decision variables to capture the assignment status of orders (such as parts to be processed) within the scheduling problems or the commitment status of power-generating units within unit commitment problems. These problems belong to an important class of so-called "Mixed-Integer Programming" (MIP) problems. Because of the very many possibilities in which 1) orders can be assigned to different machines within manufacturing systems and 2) power generating units can be committed at multiple time slots within power grids, computational requirements to obtain optimal solutions increase exponentially as the problem size increases. However, these optimization problems need to be solved frequently and with strict time limits. For example, manufacturing scheduling or day-ahead unit commitment problems frequently require short solving times such as 5-10 minutes.

With the emergence of the Internet of Things empowered by smart sensors together with advanced computation and communication technologies and with the vision of Industry 4.0, a foreseeable transition is from centralized system planning and operation toward decentralized ones. For example, within self-optimizing manufacturing factories, a system will consist of multiple distributed and interacting components/subsystems that need to be coordinated. Within these futuristic factories, distributed subsystems, such as robots, machines, or parts, will be coordinated through 5G networks to meet certain objectives, such as on-time delivery. The related operation optimization problems include planning, scheduling, and dispatching. Scheduling problems are solved before each shift and require short solving time, such as a few minutes, and online dispatching of a part to a machine may require a few seconds. Because of the many possible interconnections among parts and machines, an efficient communication scheme is required to prevent bandwidth overloading. This motivates the need for efficient and coordinated operation while ensuring high computational and communication efficiency.

2 Discrete Optimization: Difficulties on the Way to Distributed Optimization

The aforementioned problems are naturally created by establishing subsystems first and then by coupling them together to form the overall system. Within manufacturing scheduling, orders can be viewed as subsystems, which are

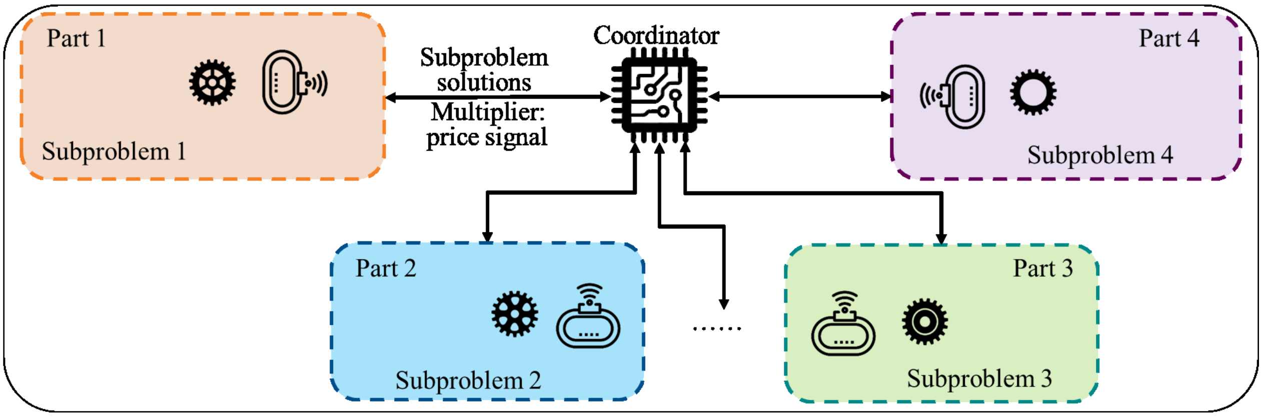


Figure 1: Decomposition and coordination framework for distributed coordination within manufacturing scheduling.

coupled by machine capacity constraints [1, 2]; and within power systems, power generating units can be viewed as subsystems, coupled by system demand and transmission capacity constraints. The problems are often formulated as mixed-integer linear programming (MILP) problems, which are MIP problems with linear structures, and integer linear programming (ILP) problems are a special case. The corresponding objective functions are additive in terms of cost components associated with each subsystem. In addition, constraints that couple subsystems are linear, therefore, are also additive in terms of subsystems. Such MILP problems are thus always separable.

The transition to the sensor-based and communication-enabled cyber-physical systems that include the above problems is also complicated because of the multiple interacting components. Moreover, the inherent combinatorial complexity implies that as a system grows, the number of the possible solutions that a system admits "explodes" exponentially. For example, when a manufacturing factory expands to include more machines to process the growing number of orders, the number of combinations in which parts can be processed on available machines increases exponentially, resulting in computational challenges.

3 Distributed and Asynchronous Surrogate Lagrangian Relaxation: Paving the Way to Efficient Distributed Optimization

To enable the transition toward decentralized system planning and operation, efficient distributed optimization methods are needed. To exploit the beautiful property of exponential reduction of complexity upon decomposition into subproblems (e.g., "machine" or "part" subproblems), Lagrangian Relaxation (LR) was traditionally used. In the offline implementation, after relaxing coupling constraints such as machine capacity constraints, "part" subproblems are coordinated by updating Lagrangian multipliers. In essence, machines can be viewed as a "supply" of resources, and parts can be viewed as a "demand." The multiplies are viewed as "prices," which are iteratively increased when "demand" is higher than "supply," and vice versa. While the standard LR method suffers from slow convergence because of the need to solve all the subproblems, the major difficulties of the method have been overcome by Surrogate Lagrangian Relaxation (SLR) [3]. The main idea to ensure convergence is the "contraction mapping concept" that ensures fast convergence of multipliers. The method has been further improved through an asynchronous update of multipliers, which can be implemented both offline and online, without the need for synchronization as shown in Figure 1 with manufacturing scheduling as an example [4].

Moreover, the communication is limited between the coordinator and subsystems (the "star network"), and the information exchange only involves multipliers transmitted from the coordinator to the subsystems and subproblem solutions transmitted from subsystems to the coordinator, thereby avoiding bandwidth overloading. As a result, the plug-and-play capabilities are also enabled: parts arrive, get processed, and get shipped without disturbing the entire network topology; and the communication requirements are much reduced and the private subsystem information is kept by avoiding the need for subsystems such as parts to communicate with each other. The methodology has also been successfully tested for asynchronous coordination of networked microgrids [5].

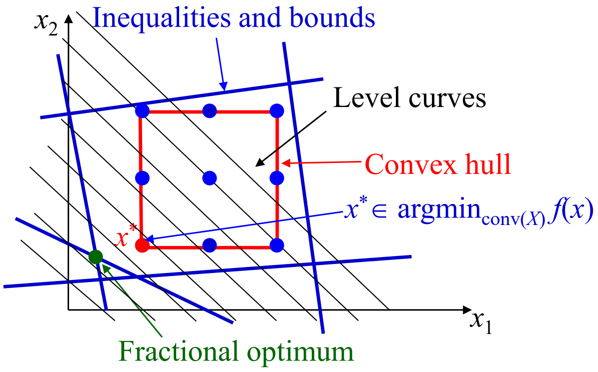


Figure 2: Formulation Tightening: "MILP to LP."

4 Formulation Tightening: Plug-and-play Compatible Acceleration of Solution Methodology

When subsystems of the above MILP systems are complicated with coupling constraints, such as operation precedence requirements in manufacturing scheduling and ramp-rate constraints in unit commitment, the subproblems may be still difficult to solve by state-of-practice MILP methods. To overcome this, the idea is formulation tightening, which is of critical importance but has been much overlooked. For an MILP problem, an optimal solution is guaranteed to be at one of the vertices of its convex hull and if problem constraints can be transformed to directly delineate the convex hull (i.e., the formulation is "tight") in the data pre-processing stage, then a solution can be obtained by using linear programming (LP) methods without combinatorial difficulties as shown in Figure 2.

To tighten subproblem formulations in the pre-processing stage, a systematic approach has been developed for mixed-binary linear programming problems (binary decision variables instead of integer ones) [6]. The idea is to derive vertices of the convex hull without binary requirements. Vertices of the original convex hull are then innovatively obtained by eliminating vertices with fractional values for binary variables. These vertices are converted to tightened constraints. For general use purposes, these tightened constraints are characterized by analyzing constraint structures and relationships between coefficients and subsystem parameters. This tremendously reduces online computational requirements. With tightened constraints, the subproblem solutions are obtained much faster as compared with the original formulation. The tightening procedure is also plug-and-play compatible: given a new type of subsystems such as new parts to be manufactured, the corresponding tightened formulation is developed offline, and the overall problem is solved with an additional type of subproblems [2] by following an asynchronous decomposition and coordination framework of Section 3. The approach has also been extended to MILP problems with special relations between integer and binary variables (integer variables are uniquely determined by the binary variables) [2].

5 Conclusion

This newsletter provides a brief overview of the recent solution methodological and modeling advancements to enable efficient distributed coordination of subsystems within cyber-physical systems involving combinatorial optimization. This synergistic combination of sensor- and communication-based systems with the decomposition and coordination methods paves the way toward enhanced capabilities of the cyber-physical systems to handle complex optimization problems.

References

[1] B. Yan, M. A. Bragin, and P. B. Luh, "Novel Formulation and Resolution of Job-Shop Scheduling Problems," IEEE Robotics and Automation Letters, vol. 3, no. 4, Oct. 2018, pp. 3387-3393. DOI: 10.1109/LRA.2018.2850056

[2] B. Yan, M. A. Bragin, and P. B. Luh, "An Innovative and Systematic Formulation Tightening Method for Job-Shop Scheduling," Early Access, IEEE Transactions on Automation Science and Engineering, 10.1109/TASE.2021.3088047.

[3] M. A. Bragin, P. B. Luh, J. H. Yan, N. Yu, and G. A. Stern, “Convergence of the Surrogate Lagrangian Relaxation Method,” Journal of Optimization Theory and Applications, Vol. 164, Issue 1, 2015, pp. 173-201, DOI: 10.1007/s10957-014-0561-3.

[4] M. A. Bragin, P. B. Luh, and B. Yan, “Distributed and Asynchronous Coordination of a Mixed-Integer Linear System via Surrogate Lagrangian Relaxation,” IEEE Transaction on Automation Science and Engineering, vol. 18, issue 3, June 2020, pp. 1191-1205. DOI: 10.1109/TASE.2020.2998048.

[5] N. Nikmehr, M. A. Bragin, P. B. Luh, and P. Zhang, “Distributed and Asynchronous Operational Optimization of Networked Microgrids.” arXiv preprint available at http://arxiv.org/abs/2102.03496

[6] B. Yan, P. B. Luh, T. Zheng, D. Schiro, M. A. Bragin, F. Zhao, J. Zhao, and I. Lelic, “A Systematic Formulation Tightening Approach for Unit Commitment Problems,” IEEE Transactions on Power Systems, Vol. 35, Issue 1, pp. 782 - 794, 2020.

A Short Survey of Automatic Generation Control Considering Cyber Security

Chunyu Chen[1], Junbo Zhao[2], Xiao Zhang[1]

[1]School of Electrical and Power Engineering, China University of Mining and Technology
[2]Department of Electrical and Computer Engineering, Mississippi State University

Abstract

To address the growing concern of cyber attacks against information system-assisted smart grid applications, cyber security of automatic generation control (AGC) needs careful considerations. Attackers can exploit vulnerabilities of target communication systems (cyber layers) to affect frequency-dependent activities (physical layers). This article presents an overview of cyber attacks against AGC. Specifically, current state-of-the-art detection and mitigation methods are reviewed. By offering a brief introduction of existing studies on AGC cyber security, this short survey article intends to start further investigation into this matter.

1 Introduction

Automatic generation control (AGC), also termed load frequency control (LFC), has been widely used in the secondary control architectures of various electric power systems, e.g., transmission systems and microgrids. As one type of remote control systems, AGC relies on long-distance telecommunication, which bears more cyber risks than its local short-distance counterpart. Even if data transmission can be established over "secure" private networks, system operators cannot fully resist possible cyber attacks. For example, end device vulnerabilities can be exploited to compromise the supervisory control and data acquisition (SCADA) system. Then, attackers can implement multi-stage attacks by meticulously reconfiguring or disabling specific applications and devices. The aforementioned intrusion is usually performed by "far-sighted" intelligent attackers, which will conceal themselves for months to search for the weak communication endpoint. The ultimate objective of this multi-stage attack is to sabotage the whole system by causing wide-scale malfunction and power outages, just as the Ukraine power grid cyber attack in 2015. In fact, not all the attackers can have adequate capabilities to launch multi-stage attacks, especially when it comes to AGC-oriented attacks. Since an AGC attack objective is generally linked to frequency destabilization or the frequency-dependent activities, attackers can simply use remote terminal unit (RTU) vulnerabilities to damage the integrity of frequency data transmission. In summary, AGC-oriented attacks have the following characteristics:

- Attackers want to sabotage the whole control center through meticulous attack planning and coordination. Then, they can design more flexible and complex attack policies.
- Attackers only intrude on peripheral vulnerable edge devices (e.g., RTU). Attackers in this context usually only aim to destabilize the frequency by damaging the frequency data transmission.

Centering around these two characteristics above, various AGC-oriented cyber security studies have been conducted. These latest research can be categorized into two main subjects: (1) cyber attack detection; (2) cyber attack mitigation. Cyber attack detection (anomaly detection), as the name suggests, is the identification of AGC-oriented attack scenarios. As the first stage of cyber attack defense functions, anomaly detection can distinguish anomalous (attack) and normal operating conditions. Then defenders can attenuate the influence in the second stage by executing mitigation policies. Before touching on these two subjects, this article first introduces some typical attack scenarios which can somehow reflect real-life cyber attack activities. Then, some detection and mitigation methods are briefly reviewed. Interested audiences can further investigate the problems and make further contributions to these subjects.

2 AGC-Oriented Cyber Attack Scenario Analysis

Like other remote feedback control systems, AGC is mainly comprised of the controlled process (the turbine governor-supervised power generation) and the controller (AGC control algorithm, e.g., PID), as is shown in Fig. 1. Based on attackers' intellectual levels or capabilities, we have Scenario 1, where "dumb" attackers compromise sensors (RTU) to affect data integrity and frequency control performance. Also, we have Scenario 2, where "intelligent" attackers completely collapse the SCADA system and the control center to reconfigure the AGC application. In Scenario 1, the attacker has far less AGC information than the defender; hence, the attacker cannot design complex polices such as the game-based or optimization-based ones. By contrast, "intelligent" attackers in Scenario 2 can do so since they grasp the whole AGC information (e.g., the model, technical parameters, operating conditions and constraints) after infiltrating the control center. Moreover, attackers can arbitrarily change control commands based on the designed attack policies such that it is more hazardous and flexible than pure communication data-oriented attacks (Scenario 1).

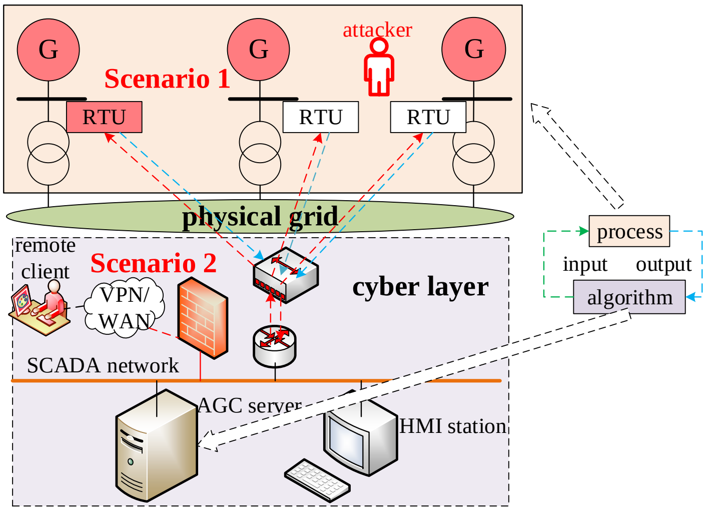


Figure 1: An overview of two AGC-oriented cyber attack scenarios: Scenario 1 represents naive sensor compromise; Scenario 2 represents advanced control center collapse.

Recently, the proliferation of data-driven technologies is intellectualizing the originally "dumb" communication data-oriented attacks by offering online system identification and decision making functions (as shown in Fig. 2). Instead of randomly injecting false data or producing network traffic, attackers will learn to extract patterns and make inferences with the aid of collected data. For example, attackers can obtain the mapping between attack input and interested system variable (e.g., the frequency) using data-driven regression. Then, this mapping can be exploited to achieve specific attack objectives. The aforementioned mapping operation is mainly used to reconstruct a pseudo-model, which can mimic external features of original AGC systems. Hence, attackers can still use model-based techniques to design attack policies based on the pseudo-model [1, 2]. Alternatively, advanced attackers can even use online learning algorithms (e.g., on-policy reinforcement learning) to obtain real-time attack policies. Rather than reconstructing the unknown AGC system, attackers only need to revise actions by exploration and exploitation operations before reaching the optimal attack policy. The AGC system in this context is just an environment on which attackers perform policies and observe system outputs. From the perspective of attackers, data-driven policy making may have problems, including the computational complexity and possible learning non-convergence. However, it can reduce the requirement for AGC information when attackers design more advanced policies.

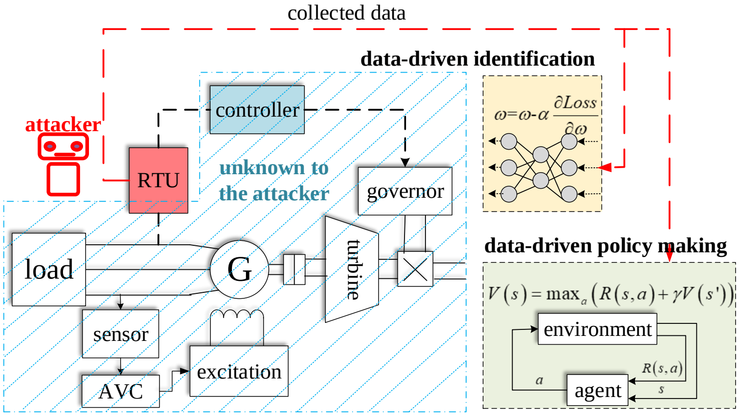

collected data
data-driven identification
attacker
controller
RTU
unknown to the attacker
governor
load
G
turbine
sensor
excitation
AVC
ω=ω-α $\frac{\partial Loss}{\partial \omega}$
data-driven policy making
$V(s)=\max_a(R(s,a)+\gamma V(s'))$
environment
$R(s,a)$
a
agent
s

Figure 2: An overview of data-driven applications in AGC sensor oriented attack scenarios.

3 An Overview of Detection and Mitigation Techniques

Since cyber attack detection in the AGC systems is a special type of anomaly detection, various anomaly detection methods, including classification-based and clustering-based ones, are possible solutions. The earliest works of AGC cyber attack detection can arguably date back to 2014 [3]. Area control errors (ACEs) from normal and attack scenarios are considered to possess different statistical features. Therefore, the outliers (anomalies) can be screened out by checking tails of ACE density function. Most statistical analysis-based detection methods use scalar values, which efface temporal details of the dynamic system responses. These temporal details, in the form of time-series data, may contain more distinguishable patterns that cannot be extracted from scalar data. Consequently, follow-up studies of AGC attack detection use time series data of ACE or the frequency to improve detection accuracy [4, 5]. Inspired by the digital watermarking technique in Steganography, a "small" private signal can be superimposed onto data (e.g., control command signal). The private signal is only known to the participating units and the compromised data will likely exhibit statistical features that are irrelevant to the private signal. Then, AGC attacks can be detected based on this statistical heterogeneity [6]. It should be noted that detection success is highly dependent upon whether the false data is distinguishable from the normal one. Successful detector operation demands the data heterogeneity, which ranges from apparently statistics to inherently heterogeneous patterns. Meanwhile, self-evolving attackers learn to develop resistance mechanisms by making the heterogeneous pattern more homogeneous, thus increasing the difficulty of intrusion detection. For example, a generative adversarial network (GAN) can be used to generate false data that appear authentic to the aforementioned detectors. Nevertheless, if the false data "look" very similar to the normal one, attackers usually cannot obtain attack objectives by incurring substantial damages. Therefore, there is a trade-off between the complexity of detection and hazard level.

Mitigation is often regarded as the second phase of the cyber attack defense mechanism of industrial cyber physical systems (CPS). The mitigation can be classified into active mitigation and passive mitigation. In the active mitigation, defenders will identify both the presence and the quantitative information of cyber attacks. The latter will then contribute to the AGC controller reconfiguration, in which the controller structure or parameter change is usually involved. In this sense, unknown input observer (UIO)-based mitigation policies belong to the active mitigation [7, 8, 9, 10]. The main idea of UIO-based mitigation is the controller reconfiguration with reconstructed cyber attack inputs. As for the reconstruction of cyber attack inputs, both model-based [7, 8, 9] and data-driven approaches [10] can be applied based on system characteristics and specific design conditions. By contrast, the passive mitigation has fixed policy at the design stage. Rather than actively reconfiguring the AGC controller, defenders choose to use the system or the controller redundancy to "tolerate" cyber intrusions. An exploratory study of passive mitigation is for the first time investigated in [11]. The proposed passive mitigation policy is a data-driven one. Instead of using data-driven methods to quantify the attack inputs, a complete and integrated data-driven defending policy is designed using the reinforcement learning technique [11]. Neither the active nor the passive mitigation has overwhelming advantages. Passive mitigation may be applauded for its convenience. But the limited "tolerance level" can render it ineffective when facing some "unseen" attacks. Though the two-stage reconstruction and reconfiguration increase the computational complexity, the active mitigation still has benefits, such as enhanced mitigation degree and attack detection (as a by-product). The specific appealing mitigation policies vary on a case-by-case basis.

4 Conclusion

This article offers a short overview of AGC-oriented cyber security. It focuses on two subjects: 1) the feasibility and principle of AGC-oriented cyber attacks; 2) detection and mitigation of AGC-oriented cyber attacks. The working principles of two realistic attack scenarios, "dumb" sensor-oriented and "intelligent" control center-oriented attack scenarios, are depicted and compared. Further, advanced composite data-driven sensor-oriented scenarios are introduced. This article then briefly compare some SoTA detection and mitigation methods, including the active and passive mitigation policies. Future work will aim to improve the detection and mitigation by considering the reduced dependence upon AGC model and computational burdens. Also, future work will consider promoting the data-driven active and passive mitigation policies to other industrial CPS.

References

[1] Rui Tan, Hoang Hai Nguyen, Eddy YS Foo, David KY Yau, Zbigniew Kalbarczyk, Ravishankar K Ivyer, and Hoay Beng Gooi. Modeling and mitigating impact of false data injection attacks on automatic generation control. *IEEE Transactions on Information Forensics and Security*, 12(7):1609–1624, 2017.

[2] Chunyu Chen, Mingjian Cui, Xinan Wang, Kaifeng Zhang, and Shengfei Yin. Load altering attack-tolerant defense strategy for load frequency control system. *IEEE Access*, 6(1):30414–30423, 2018.

[3] Siddharth Sridhar and Manimaran Govindarasu. Model-based attack detection and mitigation for automatic generation control. *IEEE Transactions on Smart Grid*, 5(2):580–591, 2014.

[4] Chunyu Chen, Kaifeng Zhang, Kun Yuan, Minhui Qian, and Lingzhi Zhu. Novel detection scheme design considering cyber attacks on load frequency control. *IEEE Transactions on Industrial Informatics*, 14(5):1932–1941, 2018.

[5] Wenjun Bi, Kaifeng Zhang, Yaping Li, Kun Yuan and Ying Wang. Detection scheme against cyber-physical attacks on load frequency control based on dynamic characteristics analysis. *IEEE Systems Journal*, 13(3):2859–2868, 2019.

[6] Tong Huang, Bharadwaj Satchidanandan, P.R. Kumar, and Le Xie. An online detection framework for cyber attacks on automatic generation control. *IEEE Transactions on Power Systems*, 33(6):6816–6827, 2018.

[7] Amir Ameli, Ali Hooshyar, Ehab F El-Saadany, and Amr M Youssef. Attack detection and identification for automatic generation control systems. *IEEE Transactions on Power Systems*, 33(5):4760–4774, 2018.

[8] Amir Ameli, Ali Hooshyar, Ameen Hassan Yazdavar, Ehab F El-Saadany, and Amr M Youssef. Attack detection for load frequency control systems using stochastic unknown input estimators. *IEEE Transactions on Information Forensics and Security*, 13(10):2575–2590, 2018.

[9] Mohsen Khalaf, Amr M Youssef, and Ehab F El-Saadany. Joint detection and mitigation of false data injection attacks in AGC systems. *IEEE Transactions on Smart Grid*, 10(5):4985–4995, 2018.

[10] Chunyu Chen, Yang Chen, Junbo Zhao, Kaifeng Zhang, Ming Ni, and Bixing Ren. Data-driven resilient automatic generation control against false data injection attacks. *IEEE Transactions on Industrial Informatics*, 2021, early access.

[11] Chunyu Chen, Mingjian Cui, Xin Fang, Bixing Ren, and Yang Chen. Load altering attack-tolerant defense strategy for load frequency control system. *Applied Energy*, 280(1):116015, 2020.

Technical Activities

1 Conferences and Workshops

- IEEE International Conference on Cyber Physical and Social Computing (CPSCom 2020)
- IEEE Sensors Council Summer School 2020

2 Special Issues in Academic Journals

- IEEE Internet of Things Journal special issue on Security, Privacy, and Trustworthiness in Intelligent Cyber-Physical Systems and Internet-of-Things
- IEEE Transactions on Automation Science and Engineering special issue on Machine Learning for Resilient Industrial Cyber-Physical Systems
- SCIENCE CHINA Information Sciences special issue on Cyber-Physical Systems

Newsletter of Technical Committee on Cyber-Physical Systems (IEEE Systems Council)

The newsletter of Technical Committee on Cyber-Physical Systems (TC-CPS) aims to provide timely updates on technologies, educations and opportunities in the field of cyber-physical systems (CPS). The letter will be published twice a year: one issue in February and the other issue in October. We are soliciting contributions to the newsletter. Topics of interest include (but are not limited to):

- Embedded system design for CPS
- Real-time system design and scheduling for CPS
- Distributed computing and control for CPS
- Resilient and robust system design for CPS
- Security issues for CPS
- Formal methods for modeling and verification of CPS
- Emerging applications, e.g. automotive system, smart energy system, biomedical device, etc.

Please directly contact the editors and/or associate editors by email to submit your contributions.

Submission Deadline:

All contributions must be submitted by **Jan. 15, 2022** in order to be included in the August issue of the newsletter.

Editor:

- Bei Yu, Chinese University of Hong Kong, Hong Kong byu@cse.cuhk.edu.hk

Associate Editors:

- Xianghui Cao, Southeast University, China xhcao@seu.edu.cn
- Long Chen, Sun Yat-Sen University, China chenl46@mail.sysu.edu.cn
- Caiwen Ding, University of Connecticut, USA caiwen.ding@uconn.edu
- Keke Huang, Central South University huangkeke@csu.edu.cn
- Yier Jin, University of Florida, USA yier.jin@ece.ufl.edu
- Subham Sahoo, Aalborg University, Denmark sssa@et.aau.dk

- Muhammad Shafique, Vienna University of Technology, Austria mshafique@ecs.tuwien.ac.at
- Umamaheswara Rao Tida, North Dakota State University, USA umamaheswara.tida@ndsu.edu
- Qi Xu, University of Science and Technology of China, China xuqi@ustc.edu.cn
- Xiaolin Xu, Northeastern University, USA x.xu@northeastern.edu
- Ming-Chang Yang, Chinese University of Hong Kong, Hong Kong mcyang@cse.cuhk.edu.hk
- Junbo Zhao, Mississippi State University, USA junbo@ece.msstate.edu